Massimo Melucci, Stefano Mizzaro and Gabriella Pasi (Eds.)

Proceedings of the
First Italian Information Retrieval Workshop

IIR 2010

Department of Information Engineering, University of Padua, Italy
27 – 28 January 2010
http://ims.dei.unipd.it/websites/iir10/

Contents

Preface

The primary aim of the Italian Information Retrieval (IIR) workshop is to provide an international meeting forum where Italian researchers, and especially early stage researchers, from the domain of Information Retrieval and related disciplines can exchange information and ideas, and present past results and innovative research developments in an informal way. IIR 2010 took place at the University of Padova, Italy, on January 27-28, 2010 in conjunction with the Italian Research Conference on Digital Libraries (IRCDL, January 28-29, 2010).

IIR 2010 contributions are related to different aspects of IR and present ideas ranging from theoretical issues to system descriptions. Most contributors of IIR are PhD students or early stage researchers. All the submissions (both long papers presenting new research results and short papers synthesizing past research) were reviewed by members of the Programme Committee and papers were selected on the basis of originality, technical depth, style of presentation, and impact.

The Workshop Organisers

Massimo Melucci
University of Padua

Stefano Mizzaro
University of Udine

Gabriella Pasi
University of Milano Bicocca

Organization

General Chair

- Massimo Melucci (University of Padua)

Program Chairs

- Stefano Mizzaro (University of Udine)
- Gabriella Pasi (University of Milano Bicocca)

Program Committee

- Gianni Amati (Fondazione Ugo Bordoni)
- Gloria Bordogna (IDPA-CNR Dalmine - Bg)
- Claudio Carpineto (Fondazione Ugo Bordoni)
- Emanuele Di Buccio (University of Padua)
- Giorgio Di Nunzio (University of Padua)
- Paolo Ferragina (University of Pisa)
- Antonio Gulli (Search Technology Center, Bing Search, Microsoft)
- Salvatore Orlando (University of Venice "Ca' Foscari")
- Alessandro Panconesi (University of Rome "La Sapienza")
- Raffaele Perego (ISTI-CNR, Pisa)
- Carol Peters (ISTI-CNR, Pisa)
- Giuseppe Santucci (University of Rome "La Sapienza")
- Fabrizio Silvestri (ISTI-CNR, Pisa)
- Fabrizio Sebastiani (ISTI-CNR, Pisa)
- Giovanni Semeraro (University of Bari)

Organizing Committee

Department of Information Engineering, University of Padua

- Maria Bernini
- Davide Cisco
- Emanuele Di Buccio
- Marco Dussin
- Nicola Montecchio

Library Centre - CAB, University of Padua

- Yuri Carrer
- Ornella Volpato

Proceedings

- Emanuele Di Buccio (University of Padua)

Acknowledgements

The workshop was supported by the

- Department of Information Engineering of the University of Padua
  `http://www.dei.unipd.it`

and was sponsored by the

- Italian Association of Computer Science (Associazione Italiana per il Calcolo Automatico, AICA) - `http://www.aicanet.it`

Semantic Vectors: an Information Retrieval scenario

Pierpaolo Basile
Dept. of Computer Science
University of Bari
Via E. Orabona, 4
70125 Bari (ITALY)
basilepp@di.uniba.it

Annalina Caputo
Dept. of Computer Science
University of Bari
Via E. Orabona, 4
70125 Bari (ITALY)
acaputo@di.uniba.it

Giovanni Semeraro
Dept. of Computer Science
University of Bari
Via E. Orabona, 4
70125 Bari (ITALY)
semeraro@di.uniba.it

ABSTRACT

In this paper we exploit Semantic Vectors to develop an IR system. The idea is to use semantic spaces built on terms and documents to overcome the problem of word ambiguity. Word ambiguity is a key issue for those systems which have access to textual information. Semantic Vectors are able to dividing the usages of a word into different meanings, discriminating among word meanings based on information found in unannotated corpora. We provide an *in vivo* evaluation in an Information Retrieval scenario and we compare the proposed method with another one which exploits Word Sense Disambiguation (WSD). Contrary to sense discrimination, which is the task of discriminating among different meanings (not necessarily known a priori), WSD is the task of selecting a sense for a word from a set of predefined possibilities. The goal of the evaluation is to establish how Semantic Vectors affect the retrieval performance.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods, Linguistic processing; H.3.3 [**Information Search and Retrieval**]: Retrieval models, Search process

Keywords

Semantic Vectors, Information Retrieval, Word Sense Discrimination

1. BACKGROUND AND MOTIVATIONS

Ranked keyword search has been quite successful in the past, in spite of its obvious limits basically due to polysemy, the presence of multiple meanings for one word, and synonymy, multiple words having the same meaning. The result is that, due to synonymy, relevant documents can be missed if they do not contain the exact query keywords, while, due to polysemy, wrong documents could be deemed as relevant. These problems call for alternative methods that work not only at the lexical level of the documents, but also at the meaning level.

In the field of computational linguistics, a number of important research problems still remain unresolved. A specific challenge for computational linguistics is ambiguity. Ambiguity means that a word can be interpreted in more than one way, since it has more than one meaning. Ambiguity usually is not a problem for humans therefore it is not perceived as such. Conversely, for a computer ambiguity is one of the main problems encountered in the analysis and generation of natural languages. Two main strategies have been proposed to cope with ambiguity:

1. **Word Sense Disambiguation:** the task of selecting a sense for a word from a set of predefined possibilities; usually the so called *sense inventory*[1] comes from a dictionary or thesaurus.

2. **Word Sense Discrimination:** the task of dividing the usages of a word into different meanings, ignoring any particular existing *sense inventory*. The goal is to discriminate among word meanings based on information found in unannotated corpora.

The main difference between the two strategies is that disambiguation relies on a sense inventory, while discrimination exploits unannotated corpora.

In the past years, several attempts were proposed to include sense disambiguation and discrimination techniques in IR systems. This is possible because discrimination and disambiguation are not an end in themselves, but rather "intermediate tasks" which contribute to more complex tasks such as information retrieval. This opens the possibility of an *in vivo* evaluation, where, rather then being evaluated in isolation, results are evaluated in terms of their contribution to the overall performance of a system designed for a particular application (e.g. Information Retrieval).

The goal of this paper is to present an IR system which exploits semantic spaces built on words and documents to overcome the problem of word ambiguity. Then we compare this system with another one which uses a Word Sense Disambiguation strategy. We evaluated the proposed system into the context of CLEF 2009 Ad-Hoc Robust WSD task [2].

The paper is organized as follows: Sections 2 presents the IR model involved into the evaluation, which embodies semantic vectors strategies. The evaluation and the results are reported in Section 3, while a brief discussion about the main works related to our research are in Section 4. Conclusions and future work close the paper.

[1]A sense inventory provides for each word a list of all possible meanings.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. AN IR SYSTEM BASED ON SEMANTIC VECTORS

Semantic Vectors are based on WordSpace model [15]. This model is based on a vector space in which points are used to represent semantic concepts, such as words and documents. Using this strategy it is possible to build a vector space on both words and documents. These vector spaces can be exploited to develop an IR model as described in the following.

The main idea behind Semantic Vectors is that words are represented by points in a mathematical space, and words or documents with similar or related meanings are represented close in that space. This provide us an approach to perform sense discrimination. We adopt the Semantic Vectors package [18] which relies on a technique called Random Indexing (RI) introduced by Kanerva in [13]. This allows to build semantic vectors with no need for the factorization of document-term or term-term matrix , because vectors are inferred using an incremental strategy. This method allows to solve efficiently the problem of reducing dimensions, which is one of the key features used to uncover the "latent semantic dimensions" of a word distribution.

RI is based on the concept of Random Projection: the idea is that high dimensional vectors chosen randomly are "nearly orthogonal". This yields a result that is comparable to orthogonalization methods, such as Singular Value Decomposition, but saving computational resources. Specifically, RI creates semantic vectors in three steps:

1. a context vector is assigned to each document. This vector is sparse, high-dimensional and ternary, which means that its elements can take values in {-1, 0, 1}. The index vector contains a small number of randomly distributed non-zero elements, and the structure of this vector follows the hypothesis behind the concept of Random Projection;

2. context vectors are accumulated by analyzing terms and documents in which terms occur. In particular the semantic vector of each term is the sum of the context vectors of the documents which contain the term;

3. in the same way a semantic vector for a document is the sum of the semantic vectors of the terms (created in step 2) which occur in the document.

The two spaces built on terms and documents have the same dimension. We can use vectors built on word-space as query vectors and vectors built on document-space as search vectors. Then, we can compute the similarity between word-space vectors and document-space vectors by means of the classical cosine similarity measure. In this way we implement an information retrieval model based on semantic vectors.

Figure 1 shows a word-space with two only dimensions. If those two dimensions refer respectively to `LEGAL` and `SPORT` contexts, we can note that the vector of the word *soccer* is closer to the `SPORT` context than the `LEGAL` context, vice versa the word *law* is closer to the `LEGAL` context. The angle between *soccer* and *law* represents the similarity degree between the two words. It is important to emphasize that contexts in WordSpace have no tag, thus we know that each dimension is a context, but we cannot know the kind of the context. If we consider document-space rather than word-space, document semantically related will be represented closer in that space.

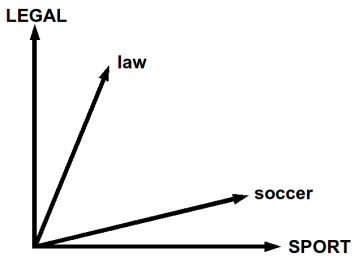


Figure 1: Word vectors in word-space

The Semantic Vectors package supplies tools for indexing a collection of documents and their retrieval adopting the Random Indexing strategy. This package relies on Apache Lucene[2] to create a basic term-document matrix, then it uses the Lucene API to create both a word-space and a document-space from the term-document matrix, using Random Projection to perform dimensionality reduction without matrix factorization. In order to evaluate Semantic Vectors model we must modify the standard Semantic Vectors package by adding some ad-hoc features to support our evaluation. In particular, documents are split in two fields, *headline* and *title*, and are not tokenized using the standard text analyzer in Lucene.

An important factor to take into account in semantic-space model is the number of contexts, that sets the dimensions of the context vector. We evaluated Semantic Vectors using several values of reduced dimensions. Results of the evaluation are reported in Section 3.

3. EVALUATION

The goal of the evaluation was to establish how Semantic Vectors influence the retrieval performance. The system is evaluated into the context of an Information Retrieval (IR) task. We adopted the dataset used for CLEF 2009 Ad-Hoc Robust WSD task [2]. Task organizers make available document collections (from the news domain) and topics which have been automatically tagged with word senses (synsets) from WordNet using several state-of-the-art disambiguation systems. Considering our goal, we exploit only the monolingual part of the task.

In particular, the Ad-Hoc WSD Robust task used existing CLEF news collections, but with WSD added. The dataset comprises corpora from "Los Angeles Times" and "Glasgow Herald", amounting to 169,477 documents, 160 test topics and 150 training topics. The WSD data were automatically added by systems from two leading research laboratories, UBC [1] and NUS [9]. Both systems returned word senses from the English WordNet, version 1.6. We used only the senses provided by NUS. Each term in the document is annotated by its senses with their respective scores, as assigned by the automatic WSD system. This kind of dataset supplies WordNet synsets that are useful for the development of search engines that rely on disambiguation.

In order to compare the IR system based on Semantic Vectors to other systems which cope with word ambiguity

[2]http://lucene.apache.org/

by means of methods based on Word Sense Disambiguation, we provide a baseline based on SENSE. SENSE: SEmantic N-levels Search Engine is an IR system which relies on Word Sense Disambiguation. SENSE is based on the N-Levels model [5]. This model tries to overcome the limitations of the ranked keyword approach by introducing *semantic levels*, which integrate (and not simply replace) the lexical level represented by keywords. Semantic levels provide information about word meanings, as described in a reference dictionary or other semantic resources. SENSE is able to manage documents indexed at separate levels (keywords, word meanings, and so on) as well as to combine keyword search with semantic information provided by the other indexing levels. In particular, for each level:

1. a *local scoring function* is used in order to weigh elements belonging to that level according to their informative power;
2. a *local similarity function* is used in order to compute document relevance by exploiting the above-mentioned scores.

Finally, a *global ranking function* is defined in order to combine document relevance computed at each level. The SENSE search engine is described in [4], while the setup of SENSE into the context of CLEF 2009 is thoroughly described in [7]

In CLEF, queries are represented by topics, which are structured statements representing information needs. Each topic typically consists of three parts: a brief TITLE statement, a one-sentence DESCRIPTION, and a more complex "narrative" specifying the criteria for assessing relevance. All topics are available with and without WSD. Topics in English are disambiguated by both UBC and NUS systems, yielding word senses from WordNet version 1.6.

We adopted as baseline the system which exploits only keywords during the indexing, identified by *KEYWORD*. Regarding disambiguation we used the *SENSE* system adopting two strategies: the former, called *MEANING*, exploits only word meanings, the latter, called *SENSE*, uses two levels of document representation: keywords and word meanings combined.

The query for the *KEYWORD* system is built using word stems in TITLE and DESCRIPTION fields of the topics. All query terms are joined adopting the OR boolean clause. Regarding the *MEANING* system each word in TITLE and DESCRIPTION fields is expanded using the synsets in WordNet provided by the WSD algorithm. More details regarding the evaluation of SENSE in CLEF 2009 are in [7].

The query for the *SENSE* system is built combining the strategies adopted for the *KEYWORD* and the *MEANING* systems. For all the runs we remove the stop words from both the index and the topics. In particular, we build a different stop words list for topics in order to remove non informative words such as *find*, *reports*, *describe*, that occur with high frequency in topics and are poorly discriminating.

In order to make results comparable we use the same index built for the *KEYWORD* system to infer semantic vectors using the Semantic Vectors package, as described in Section 2. We need to tune two parameters in Semantic Vectors: the number of dimensions (the number of contexts) and the frequency[3] threshold (T_f). The last value is used to discard terms that have a frequency below T_f. After a tuning step, we set the dimension to 2000 and T_f to 10. Tuning is performed using training topics provided by the CLEF organizers.

[3]In this instance word frequency refers to word occurrences.

Topic fields	MAP
TITLE	0.0892
TITLE+DESCRIPTION	**0.2141**
TITLE+DESCRIPTION+NARRATIVE	0.2041

Table 1: Semantic Vectors: Results of the performed experiments

System	MAP	Imp.
KEYWORD	0.3962	-
MEANING	0.2930	-26.04%
SENSE	0.4222	+6.56%
SV_{best}	0.2141	-45.96%

Table 2: Results of the performed experiments

Queries for the Semantic Vectors model are built using several combinations of topic fields. Table 1 reports the results of the experiments using Semantic Vectors and different combinations of topic fields.

To compare the systems we use a single measure of performance: the Mean Average Precision (MAP), due to its good stability and discrimination capabilities. Given the Average Precision [8], that is the mean of the precision scores obtained after retrieving each relevant document, the MAP is computed as the sample mean of the Average Precision scores over all topics. Zero precision is assigned to unretrieved relevant documents.

Table 2 reports the results of each system involved into the experiment. The column *Imp.* shows the improvement with respect to the baseline *KEYWORD*. The system SV_{best} refers to the best result obtained by Semantic Vectors reported in boldface in Table 1.

The main result of the evaluation is that *MEANING* works better than SV_{best}; in other words disambiguation wins over discrimination. Another important observation is that the combination of keywords and word meanings, the *SENSE* system, obtains the best result. It is important to note that SV_{best} obtains a performance below the *KEYWORD* system, about the 46% under the baseline. It is important to underline that the keyword level implemented in SENSE uses a modified version of Apache Lucene which implements Okapi BM25 model [14].

In the previous experiments we compared the performance of the Semantic Vectors-based IR system to SENSE. In the following, we describe a new kind of experiment in which we integrate the Semantic Vector as a new level in SENSE. The idea is to combine the results produced by Semantic Vectors with the results which come out from both the keyword level and the word meaning level. Table 3 shows that the combination of the keyword level with Semantic Vectors outperforms the keyword level alone.

Moreover, the combination of Semantic Vectors with word meaning level achieves an interesting result: the combination is able to outperform the word meaning level alone. Finally, the combination of Semantic Vectors with *SENSE* (keyword level+word meaning level) obtains the best MAP with an increase of about the 6% with respect to *KEY-*

System	MAP	Imp.
SV+KEYWORD	0.4150	+4.74%
SV+MEANING	0.3238	-18.27%
SV+SENSE	0.4216	+6.41%

Table 3: Results of the experiments: combination of Semantic Vectors with other levels

WORD. However, *SV* does not contribute to improve the effectiveness of *SENSE*, in fact *SENSE* without *SV* (see Table 2) outperforms *SV+SENSE*.

Analyzing results query by query, we discovered that for some queries the Semantic Vectors-based IR system achieves an high improvement wrt keyword search. This happen mainly when few relevant documents exist for a query. For example, query "10.2452/155-AH" has only three relevant documents. Both keyword and Semantic Vectors are able to retrieve all relevant documents for that query, but keyword achieves 0,1484 MAP, while for Semantic Vectors MAP grows to 0,7051. This means that Semantic Vectors are more accurate than keyword when few relevant documents exist for a query.

4. RELATED WORKS

The main motivation for focusing our attention on the evaluation of disambiguation or discrimination systems is the idea that ambiguity resolution can improve the performance of IR systems.

Many strategies have been used to incorporate semantic information coming from electronic dictionaries into search paradigms.

Query expansion with WordNet has shown to potentially improve recall, as it allows matching relevant documents even if they do not contain the exact keywords in the query [17]. On the other hand, semantic similarity measures have the potential to redefine the similarity between a document and a user query [10]. The semantic similarity between concepts is useful to understand how similar are the meanings of the concepts. However, computing the degree of relevance of a document with respect to a query means computing the similarity among all the synsets of the document and all the synsets of the user query, thus the matching process could have very high computational costs.

In [12] the authors performed a shift of representation from a lexical space, where each dimension is represented by a term, towards a semantic space, where each dimension is represented by a concept expressed using WordNet synsets. Then, they applied the Vector Space Model to WordNet synsets. The realization of the semantic tf-idf model was rather simple, because it was sufficient to index the documents or the user-query by using strings representing synsets. The retrieval phase is similar to the classic tf-idf model, with the only difference that matching is carried out between synsets.

Concerning the discrimination methods, in [11] some experiments in IR context adopting LSI technique are reported. In particular this method performs better than canonical vector space when queries and relevant documents do not share many words. In this case LSI takes advantage of the implicit higher-order structure in the association of terms with documents ("semantic structure") in order to improve the detection of relevant documents on the basis of terms found in queries.

In order to show that WordSpace model is an approach to ambiguity resolution that is beneficial in information retrieval, we summarize the experiment presented in [16]. This experiment evaluates sense-based retrieval, a modification of the standard vector-space model in information retrieval. In word-based retrieval, documents and queries are represented as vectors in a multidimensional space in which each dimension corresponds to a word. In sense-based retrieval, documents and queries are also represented in a multidimensional space, but its dimensions are senses, not words. The evaluation shows that sense-based retrieval improved average precision by 7.4% when compared to word-based retrieval.

Regarding the evaluation of word sense disambiguation systems in the context of IR it is important to cite SemEval-2007 task 1 [3]. This task is an application-driven one, where the application is a given cross-lingual information retrieval system. Participants disambiguate text by assigning WordNet synsets, then the system has to do the expansion to other languages, the indexing of the expanded documents and the retrieval for all the languages in batch. The retrieval results are taken as a measure for the effectiveness of the disambiguation. CLEF 2009 Ad-hoc Robust WSD [2] is inspired to SemEval-2007 task 1.

Finally, this work is strongly related to [6], in which a first attempt to integrate Semantic Vectors in an IR system was performed.

5. CONCLUSIONS AND FUTURE WORK

We have evaluated Semantic Vectors exploiting an information retrieval scenario. The IR system which we propose relies on semantic vectors to induce a WordSpace model exploited during the retrieval process. Moreover we compare the proposed IR system with another one which exploits word sense disambiguation. The main outcome of this comparison is that disambiguation works better than discrimination. This is a counterintuitive result: indeed it should be obvious that discrimination is better than disambiguation. Since, the former is able to infer the usages of a word directly from documents, while disambiguation works on a fixed distinction of word meanings encoded into the sense inventory such as WordNet.

It is important to note that the dataset used for the evaluation depends on the method adopted to compute document relevance, in this case the pooling techniques. This means that the results submitted by the groups participating in the previous ad hoc tasks are used to form a pool of documents for each topic by collecting the highly ranked documents. What we want to underline here is that generally the systems taken into account rely on keywords. This can produce relevance judgements that do not take into account evidence provided by other features, such as word meanings or context vectors. Moreover, distributional semantics methods, such as Semantic Vectors, do not provide a formal description of why two terms or documents are similar. The semantic associations derived by Semantic Vectors are similar to how human estimates similarity between terms or documents. It is not clear if current evaluation methods are able to detect these cognitive aspects typical of human thinking. More investigation on the strategy adopted for the evaluation is needed. As future work we intend to exploit several discrimination methods, such as Latent Semantic Indexing and Hyperspace Analogue to Language.

6. REFERENCES

[1] E. Agirre and O. L. de Lacalle. BC-ALM: Combining k-NN with SVD for WSD. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic*, pages 341–325, 2007.

[2] E. Agirre, G. M. Di Nunzio, T. Mandl, and A. Otegi. CLEF 2009 Ad Hoc Track Overview: Robust - WSD Task. In *Working notes for the CLEF 2009 Workshop*, 2009. http://clef-campaign.org/2009/working_notes/agirre-robustWSDtask-paperCLEF2009.pdf.

[3] E. Agirre, B. Magnini, O. L. de Lacalle, A. Otegi, G. Rigau, and P. Vossen. SemEval-2007 Task 1: Evaluating WSD on Cross-Language Information Retrieval. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic*, pages 7–12. ACL, 2007.

[4] P. Basile, A. Caputo, M. de Gemmis, A. L. Gentile, P. Lops, and G. Semeraro. Improving Ranked Keyword Search with SENSE: SEmantic N-levels Search Engine. *Communications of SIWN (formerly: System and Information Sciences Notes), special issue on DART 2008*, 5:39–45, August 2008. SIWN: The Systemics and Informatics World Network.

[5] P. Basile, A. Caputo, A. L. Gentile, M. Degemmis, P. Lops, and G. Semeraro. Enhancing Semantic Search using N-Levels Document Representation. In S. Bloehdorn, M. Grobelnik, P. Mika, and D. T. Tran, editors, *Proceedings of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, volume 334 of *CEUR Workshop Proceedings*, pages 29–43. CEUR-WS.org, 2008.

[6] P. Basile, A. Caputo, and G. Semeraro. Exploiting Disambiguation and Discrimination in Information Retrieval Systems. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, Milan, Italy, 15-18 September 2009*, pages 539–542. IEEE, 2009.

[7] P. Basile, A. Caputo, and G. Semeraro. UNIBA-SENSE @ CLEF 2009: Robust WSD task. In *Working notes for the CLEF 2009 Workshop*, 2009. http://clef-campaign.org/2009/working_notes/basile-paperCLEF2009.pdf.

[8] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM.

[9] Y. S. Chan, H. T. Ng, and Z. Zhong. NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic*, pages 253–256, 2007.

[10] C. Corley and R. Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[12] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL*, pages 38–44, 1998.

[13] P. Kanerva. *Sparse Distributed Memory*. MIT Press, 1988.

[14] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, New York, NY, USA, 2004. ACM.

[15] M. Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm: Stockholm University, Faculty of Humanities, Department of Linguistics, 2006.

[16] H. Schütze and J. O. Pedersen. Information retrieval based on word senses. In *In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

[17] E. M. Voorhees. *WordNet: An Electronic Lexical Database*, chapter Using WordNet for text retrieval, pages 285–304. Cambridge (Mass.): The MIT Press, 1998.

[18] D. Widdows and K. Ferraro. Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

Social approach to context-aware retrieval

Luca Vassena
University of Udine
via delle Scienze, 206
Udine, Italy
vassena@dimi.uniud.it

ABSTRACT

In this paper we present a general purpose solution to Web content perusal by means of mobile devices, named Social Context-Aware Browser. This is a novel approach for the information access based on the users' context, whose aim is to retrieve what the user needs, even if she did not issue any query. Our solution is built upon a social model that exploits the collaborative efforts of the whole community of users to control and manage contextual knowledge, related both to situations and resources. This paper presents a general survey of our solution, describing the idea and presenting an implementation approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Context-aware retrieval, mobile search, social, folksonomy, Web 2.0

1. INTRODUCTION

Context-aware computing is a computational paradigm that has faced a rapid growth in the last few years, especially in the field of mobile devices. A key-role in this new approach is played by the notion of context, that is roughly described as the situation the user is in. This concept encloses important information that could be used to affect the capabilities of mobile devices, adapting them to the user's needs. In particular, contextual data can be used to predict the user needs and to seek and retrieve information, thereby reducing the complexity of the user-device interaction and providing the right information in the right place at the right time. From this point of view, because of the huge amount of contextual information and its heterogeneity and uncertainty, the mobile and context-aware computing environments represent a new challenge for Information Retrieval (IR). The combination of IR and context-aware computing has been named context-aware retrieval [4].

These considerations guided us towards a new approach to Web contents production and fruition, where contextual data are exploited to capture the dynamic nature of the user needs, of the information available, and of the relevance of this information, typical of a mobile user in the real world. This approach is named *Social Context-Aware Browser* and its novelty is threefold. First of all this is a new radical approach that aims at discovering *"the query behind the context"*: to retrieve what the user needs, even if she did not issue any query [7]. Second this is not a domain dependent application, but a new generic way of interaction and information access, able to adapt to every domain. Third, as current models for context-awareness are too limited for very general applications, this approach brings new models built upon the social dynamics at the basis of Web 2.0.

This paper is structured as follows. We first briefly survey related work (Section 2), presenting the Context-Aware Retrieval field and introducing the main ideas behind Web 2.0. We then describe our solution (Section 3), presenting a general survey, the main ideas, and an implementation approach. In Section 4 we present a brief discussion and finally we draw some conclusions and we present future work (Section 5).

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. RELATED WORK

2.1 Context-Aware Retrieval

Context-Aware Retrieval (CAR) is an extension of classical Information Retrieval (IR) that incorporates the contextual information into the retrieval process, with the aim of delivering information to the users that is relevant within their current context [4]. CAR systems are concerned with the acquisition of context, its understanding, and the application of behaviour based on the recognized context [11].

Typical CAR applications present the following characteristics [4]: a mobile user, i.e., a user whose context is changing; interactive or automatic actions, if there is no need to consult the user; time dependency, since the context may change; appropriateness and safety to disturb the user. Although CAR applications can be both interactive and proactive in their communication with the user, we concentrate on the proactive aspects, since they are more relevant to our proposal. Besides, we concentrate on the association between CAR and mobile application, as they can be considered as the prime field for CAR [4].

An example of CAR system is the Ubiquitous Web [5], a solution based on the spontaneous annotation by a community of users of objects, places, and other people with Web accessible content and services. A more general system is represented by the MoBe framework [7]. In this applica-

tion, a general inferential framework (based on ontologies and Bayesian networks) combines the information coming from sensors to infer new and more abstract contexts (user activities, needs, etc.), that are used to retrieve and execute the most relevant applications.

2.2 Web 2.0, the social web

With Web 2.0 [9] and social software we represent all web-based services with "an architecture of participation", that is, an architecture featuring a high interaction level among users and allowing users to generate, share, and take care of the content. In the plenty of tools provided by Web 2.0, we are mainly focusing on social bookmarking and folksonomies.

Social bookmarking is a method for organizing, searching, and managing documents of interest among users. In a social bookmarking system, users save links to documents of interest in order to remember or share them with the community. Social bookmarking is strictly related with the concept of folksonomy, that is the practice of annotating and categorizing content in a collaborative way, by means of informal tags. Folksonomies, that is a portmanteau of folk and taxonomy, allow users to easyly and informally descrive documents and content. This represents a powerful combination that has gained popularity as it allows a more natural and simpler management of the knowledge. The use of freely choosen categorizations and the collaborative aspect in fact allow also non-expert users to classify and find information. Folksonomies and social bookmarking for example are used in well-known Web 2.0 systems like Flickr[1], Youtube[2], Del.icio.us[3], etc.

Folksonomies however are criticized because the lack of terminological control could lead to unreliable and inconsistent results [3].

3. SOCIAL CONTEXT-AWARE BROWSER

3.1 Description

The Social Context Aware Browser (sCAB for short) [12] is a general purpose solution to Web content navigation by means of context-aware mobile devices. It allows a "physical browsing": browsing the digital world based on the situations in the real world. The main idea behind sCAB is to empower a generic mobile device with a browser able to automatically and dynamically retrieve and load Web pages, services, and applications according to the user's current context.

The sCAB acquires information related to the user and the surrounding environment, by means of sensors installed on the device or through external servers. This information, combined with the user's personal history and the community behaviour, is exploited to infer the user's current context (and its likelihood). In the subsequent retrieval process, a query is automatically built and sent to an external search engine, in order to find the most suitable Web pages for the sensed context and present them to the user.

As current models for context-awareness are too limited for very general applications like the sCAB, this approach brings new social models for CAR that exploit the collaborative efforts of the community of users. The community, in fact, is encouraged to define the contexts of interest, share, use and discuss them, associate context to content (web pages, applications, etc.), to have a dynamic and more user-tailored context representation and to enhance the process of retrieval based on users' actual situation.

In particular users can freely interact with resources and can define that a resource is useful (or not adapt) to their current context, can associate resources to particular contexts, can explicitly define the context their are in, and finally can browse resources relevant for their current context.

3.2 Model

3.2.1 *Context representation*

We represent the context as a folksonomy. Each tag is banally a keyword or string of text and represents a single contextual value [8]. We divide the contextual tags into two categories:

- Concrete tags: represent the information obtained by a set of sensors. These information can be read from the surrounding environment through physical sensors (e.g., temperature sensor), or can be obtained by other software (e.g., calendar) through logical sensors. Concrete tags that directly refers to sensors values are represented using the *triple tags* notation that are tags that uses a particular syntax (`namespace:predicate=value`) to define extra information.
  For example, `geo:longitude=12.456` is tag for the geographical longitude coordinate whose value is 12.456. Other concrete tags, can be automatically obatined by the sensed values (e.g. `afternoon`, `summer`, ...).

- Abstract tags: represent the high level contextual information that are freely associated by the users to the concrete contexts, in order to detail their context description. Some examples are: `home`, `shopping`, etc.

The difference between the two categories is faded since the contexts cannot be unambiguously assigned to one or the other category. However this partition is helpful in order to distinguish the low level information coming from sensors and the high level contextual information intoduced by users.

The user context is a "cloud" composed by an undefined number of concrete and abstract tags (Figure 1).

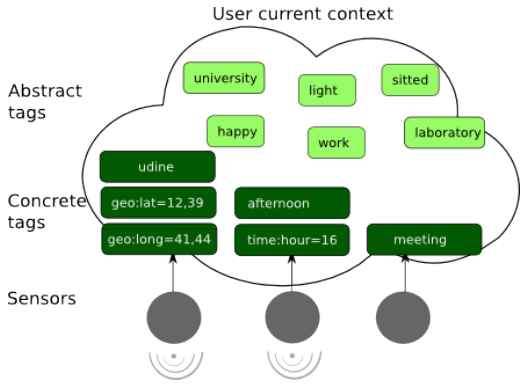


Figure 1: User's current context.

[1] `www.flickr.com`
[2] `www.youtube.com`
[3] `www.del.icio.us.com`

3.2.2 *Operations*

In the sCAB conceptual model [12] there are six main operations. The first two are performed automatically and continuosly by the system. With the *inference* operation (Figure 2), starting from the concrete tags sensed by sensors, the most relevant abstract tags are retrieved and become part of the user's context representation. Then with the *retrieval* operation (Figure 2), starting from the set of all the tags in the user's current context, the most relevant resources are retrieved. For example, starting from the GPS coordinates, the system enhance the user's context with the abstract tags "`walk out park dog`"; then starting from all the tags, the system retrieves resources relevant to the given context, as Web pages that teaches how to train dogs, etc.

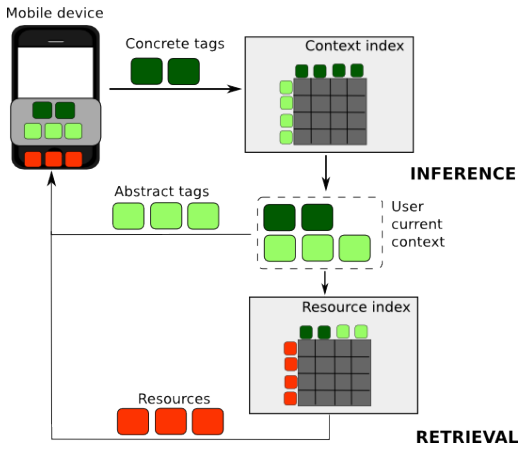


Figure 2: Inference and retrieval operations.

The other four operations are strictly related to the user interaction: the main two are *definition* and *annotation* (Figure 3). The *definition* is used to manage the contextual information and it is performed when a user directly define her context, or when she provides contextual tags during the annotation of a resource. In particular, this operations manages the associations between concrete and abstract tags, and the strength of their relationships. The *annotation* on the contrary is used to manage the association between contextual tags and resources and it is performed when the users link resources to particular contexts. We can imagine a user at a park with her dog: she wants to associate to her context a particular Web page teaching dog training. For this reason she bookmarks that resource with the contextual tags "`out dog park sunny train`". Doing so, first the added abstract tags are related to the sensed concrete tags and for all the users with a similar concrete tag cloud, these abstract tags (or part of them) can become part of the their context representation. Second, that particular Web page is enhanced with all the tags, and it will be automatically proposed to users every time they will be in a similar context.

As the users are the main actors in the process of context definition and resource annotation, problems related to the quality of context and resources are likely to appear. To cope with this problem we propose the adoption of a social evaluation/reputation mechanism. We exploit the ideas presented in [6]: every element in the model (users, contexts, resources) has a score that increases or decreases based on the community behavior. The score of each user is used to

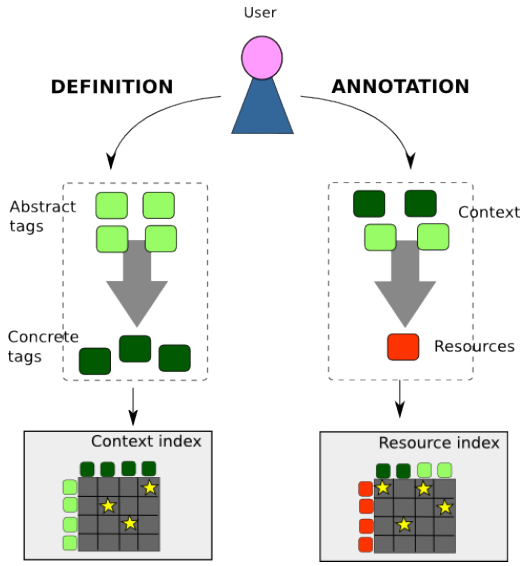


Figure 3: Definition and annotation operations.

weight the operations she performs, while the scores of contextual tags and resources define their quality and relevance. If a resource annotated with contextual information is never used in that context, the related score decreases and more relevant resources will stand out.

3.3 Implementation approach

Concrete and abstract tags, and resources are the main elements in our implementation model. Concrete tags, as output of sensors, are exploited to retrieve the most relevant abstract tags, and in the same way all the tags are exploited to retrieve the most relevant resources.

In the following sections we show an implementation proposal and how the different operations in the model have effect on the system, from a low level point of view.

3.3.1 *Indexes*

We exploit two indexes. In the first one, called *contexts index*, abstract tags are indexed over concrete tags, while in the second one, called *resources index*, resources are indexed over the set of all tags (both concrete and abstract). The proposed approach is community based, thus the indexes and the inferential system are managed by remote servers and not stored on the mobile device. Since the approach is similar for both the indexes, we are going to show just the first one.

The contexts index is a matrix that describes the frequency of abstract tags over the concrete ones. Each column corresponds to a concrete tag, and each row corresponds to an abstract tag. Each entry in the matrix has three values (Figure 4):

- U_{ij}: represents the user that has associated the abstract tag i to the concrete tag j first;
- S_{ij}: a score that defines how relevant the abstract tag i is for the concrete tag j. This value is in the interval $[0, 1]$;
- σ_{ij}: steadiness value that defines how steady is the association between the abstract tag i and the concrete

tag j.

	c_1	c_2	...
a_1			
a_2		$(U_{22}, S_{22}, \sigma_{22})$	
$\vdots$			

Figure 4: *Contexts index* example

Intuitively, since not all the abstract tags can be related to all concrete tags, the proposed index will be a very sparse matrix. At the same time, because of the very high number of both concrete and abstract tags, the index can assume very huge dimensions. However a lot of research is being performed on indexes designing and analysis, also in the CAR field [2]. The related discussion is out of the scope this work.

3.3.2 Users' score

In our approach two values are associated to each user and they define the goodness of the user in working with contextual information:

- S_{U_c}: a score that defines how good the user is in associating concrete tags to abstract tags;
- S_{U_r}: a score that defines how good the user is in associating resources to contexts;

As previously, we are concentrating only on the management of values related to concrete and abstract tags, since the approach is exactly the same working at the higher level of tags and resources.

Every time a new relation between abstract and concrete tags is created with a *definition* ("filling a hole" in the index), the user who performed the operation is associated to that relation. Then on the basis of how the community interacts with those contextual information, the user's score will be update. It is calculated as follows: for each association among tags ij performed by the user U, S_{U_c} corresponds to the mean of the products $\frac{\sigma_{ij}}{\sigma_{max}} \times S_{ij}$, where σ_{max} is the max steadiness value in the index.

New associations have a low steadiness value, thus their score, as their have not steadied yet, will have low influence on the user's score. Good associations will have high score and steadiness values, and they will reflect on high users' score. In the same way, low users' scores are due to bad associations between contextual tags. Since $S_{ij} \in [0, 1]$, also $S_{U_c} \in [0, 1]$.

In this approach, for simplicity, only new associations between tags are considered for the computation of the users' score. An extension could consider all the existing associations. In this way a user is "good" because she defines good new associations and because she exploits existing good association.

3.3.3 Values update

The proposed indexes are not static, but the values related to the association between concrete and abstract tags and resources are continuosly updated, based on the interaction of users with resources in context.

With every *definition* operation the values in the *contexts index* are updated according to the following system (for the values in the *resources index* with the *annotation* operation the approach is similar) :

- $\sigma_{ij}(t_{i+1}) = \sigma_{ij}(t_i) + S_{U_c}(t_i) \times \beta$
- $v = \dfrac{\sigma_{ij}(t_i) \times S_{ij}(t_i) \pm S_{U_c}(t_i) \times \beta}{\sigma_{ij}(t_{i+1})}$
- $S_{ij}(t_{i+1}) = \begin{cases} v & if\ v > 0 \\ 0 & otherwise \end{cases}$

where t_i represents a discrete time instant and t_{i+1} the subsequent time instant.

While the score is a value in the interval $[0, 1]$, the steadiness is an always increasing value. The higher the steadiness of an association is, the more stable the association is, and then the lesser effect each update operation will have. The user's score is exploited for the update of the values in the index. It can both increase an association, or decrease it (e.g. a user removes a tags from his context). The higher the user's score is, the more effective the update operation will be. This means that good users have more influence on the system than bad users. Finally, β is a parameter greater than 0 and it is used to weight the user score: operation performed explicitly by users (inclusion or removal of abstract tags) have more effect than implicit update performed automatically based on the interaction of the community with the resources.

3.3.4 Inference and retrieval

The *inference* and *retrieval* operations works respectively on the first and second index, but they are similar, thus in the following we are explaining just the inference one.

The approach is the following:

1. starting from the concrete tags in input, we consider only the set of abstract tags that have been associated at least with one of the concrete tags;
2. for each abstract tag we compute a *rank* value, to define an order of relevance for the abstract tags;
3. in order to limit the number of retrievd tags, we retrieve the abstract tags whose *rank* value is higher than the mean of all *rank* values.

The *rank* value is computed following an adapted version of the tf.idf weighting scheme. In particular for each considered abstract tag a_i we have:

- $A = \sum_{c_j} \sigma_{ij} \times S_{ij}$, for each sensed concrete tag c_j
- $B = \dfrac{|C|}{|\{c :\ a_i \in c\}|}$, where $|C|$ is the total number of sensed concrete tags, and $|\{c :\ a_i \in c\}|$ is the number of concrete tags to which the abstract tag a_i has been associated;
- $rank\ value = A\alpha \times B\beta$, where α, β are parameters exploited to weight the different values.

Some considerations can be drawn. First, more are the concrete tags in the current context to which an abstract tag is associated, the higher will be its rank value. Second, abstract tags with high score and steadiness will have an higher rank value. Third, abstract tags related to particular sets of concrete tags will have an higher rank value than

very general ones that are associated to an high number of concrete tags (high frequency).

In addition, starting from this basic approach, we can enhance the rank value computation exploiting other information. For example a reasonable idea is to weight the tags based on their age in the user's context representation, giving more importance to the newest tag. In this we enhance the importance of new contexts.

4. DISCUSSION

Although the conceptual ideas are clear, the implementation approach we propose is in an initial stage of definition. We suggested a possible solution, but several are the ways to refine it and several are the algorithms to be exploited. For this reason the evaluation hold an important role in our work: since different alternative solution exist, it is important to evaluate them and compare their effectiveness.

Even if the knowledge related to the whole community is exploited to infer and refine the current context of single users, the proposed model differentiates the personal from the community level, giving more importance to the first one. For example if a user annotates a situation as "`play`", she is considered to be in "`play`" context, even if most people annotate the same situation as "`work`". On the contrary, if a user is for the first time in a situation (e.g. location never visited), her context is refined just with the information from the community. Considering the previous example, as most people annotate the situation with "`work`", the user is considered to be in "`work`" context.

In the last case, the assumption performed by the system in order to provide the user with relevant resources could be wrong. However this is not a problem. Since we are working with people, it will be hardly possible to provide results that totally satisfy each user, due the intrinsic difference of views and needs in a community. Rather our solution aims at and averagely good behavior.

Talking about the indexes, we have seen how the related information are changed dynamically based on community interaction. However this is not the only possible approach. We can imagine complementary approaches that can support the community statistical one. For example, we could use some geographic gazetteer for associating geonames to geographic coordinates provided from the concrete tags, so as to reinforce the rank of associated abstract tags that contain the same geographic names or names of close localities. The geonames could be useful also for retrieving more relevant resources, those containing the geonames ore close geonames.

5. CONCLUSIONS

In this paper we have presented the Social Context-Aware Browser, a general purpose solution to Web content perusal by means of mobile devices. The sCAB is a novel approach for the information access based on context, where the community of users is called to manage the contextual knowledge, both related to situations and resources, through collaboration and participation. In particular we presented a general survey, the main ideas, and an implementation approach.

As future work we aim at implementing a prototype of the proposed system, and, in particular, we suggest a multistage approach, where implementation and evaluation processes will proceed hand in hand. As first step we want to exploit benchmarks to evaluate detailed implementation solutions, like, for example, different algorithms to assess the relevance of tags for situations and resources. After that, we plan to apply an IIR evaluation methodology, involving users in a controlled environments, following the ideas presented [1, 10]. Finally a broader user-centred evaluation will help us to understand if the sCAB is effective in the real world.

Acknowledgements

The authors acknowledge the financial support of the Italian Ministry of Education, University and Research (MIUR) within the FIRB project number RBIN04M8S8, and the region Friuli Venezia Giulia. This research has been partially supported by MoBe Ltd. (www.mobe.it), an academic spin-off company specializing in software for mobile devices.

6. REFERENCES

[1] P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):8–3, 2003.

[2] A. Göker, S. Watt, H. I. Myrhaug, N. Whitehead, M. Yakici, R. Bierig, S. K. Nuti, and H. Cumming. An ambient, personalised, and context-sensitive information system for mobile users. In *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence*, pages 19–24. ACM, 2004.

[3] S. A. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Arxiv preprint cs.DL/0508082*, 2005.

[4] G. J. F. Jones and P. J. Brown. Context-aware retrieval for ubiquitous computing environments. In *Mobile HCI Workshop on Mobile and Ubiquitous Information Access*, volume 2954, pages 227–243. Springer LNCS, 2004.

[5] D. Lopez de Ipiña, J. I. Vazquez, and J. Abaitua. A context-aware mobile mash-up plaftorm for ubiquitous web. In *Proc. of 3rd IET Intl. Conf. on Intelligent Environments*, pages 116–123, 2007.

[6] S. Mizzaro. Quality control in scholarly publishing: A new proposal. *J. of the Am. Soc. for Information Science and Technology*, 54(11):989–1005, 2003.

[7] S. Mizzaro, E. Nazzi, and L. Vassena. Retrieval of context-aware applications on mobile devices: how to evaluate? In *Proc. of Information Interaction in Context (IIiX '08)*, pages 65–71, 2008.

[8] S. Mizzaro, E. Nazzi, and L. Vassena. Collaborative annotation for context-aware retrieval. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 42–45. ACM, 2009.

[9] T. O'Reilly. What is web 2.0, design patterns and business models for the next generation of software, 2005.

[10] D. Petrelli. On the role of user-centred evaluation in the advancement of interactive information retrieval. *Inf. Process. Manage.*, 44(1):22–38, 2008.

[11] A. Schmidt. *Ubiquitous Computing - Computing in Context*. PhD thesis, Lancaster University, 2003.

[12] L. Vassena. Context-aware retrieval going social. In *3rd Symposium on Future Directions in Information Access (FDIA).*, 2009.

Selecting Features for Ordinal Text Classification*

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi, 1 – 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

ABSTRACT

We present four new feature selection methods for ordinal regression and test them against four different baselines on two large datasets of product reviews.

1. INTRODUCTION

In (text) classification, *feature selection* (FS) consists in identifying a subset $S \subset T$ of the original feature set T such that $|S| \ll |T|$ ($\xi = |S|/|T|$ being called the *reduction level*) and such that S reaches the best compromise between (a) the effectiveness of the resulting classifiers and (b) the efficiency of the learning process and of the classifiers. While feature selection has been extensively investigated for standard classification, it has not for a related and important learning task, namely, *ordinal classification* (OC – aka *ordinal regression*). OC consists in estimating a *target function* $\Phi : D \rightarrow R$ mapping each object $d_i \in D$ into exactly one of an ordered sequence $R = \langle r_1 \prec \ldots \prec r_n \rangle$ of *ranks*, by means of a function $\hat{\Phi}$ called the *classifier*.

We here address the problem of feature selection for OC. We use a "filter" approach, in which a scoring function $Score$ is applied to each feature $t_k \in T$ in order to measure the predicted utility of t_k for the classification task (the higher the value of $Score$, the higher the predicted utility), after which only the $|S|$ top-scoring features are retained. We have designed four novel feature scoring functions for OC, and tested them on two datasets of product review data, using as baselines the only three feature scoring functions for OC previously proposed in the literature, i.e., the *probability redistribution procedure* (PRP) function proposed in [4], and the *minimum variance* (Var) and *round robin on minimum variance* ($RR(Var)$) functions proposed in [2].

2. FEATURE SELECTION FOR OC

Our first method, $Var{*}IDF$, is a variant of the Var method described in [2], and is meant to prevent features occurring very infrequently, such as *hapax legomena*, to be top-ranked, which would obviously be undesirable. It is defined as

$$Score(t_k) = -Var(t_k) * (IDF(t_k))^a \quad (1)$$

where IDF is the standard inverse document frequency and a is a parameter (to be optimized on a validation set) that allows to fine-tune the relative contributions of variance and IDF to the product. Given that $Var(t_k) = 0$ implies that $Score(t_k) = 0$, we smooth $Var(t_k)$ by adding to it a small value $\epsilon = 0.1$ prior to multiplying it by $IDF(t_k)$.

Our second method, $RR(Var{*}IDF)$, is a variant of $Var{*}IDF$ meant to prevent it from exclusively catering for a certain rank and disregarding the others. It consists in (i) provisionally "assigning" each feature t_k to the rank closest to the mean of its distribution across the ranks; (ii) sorting, for each rank r_j, the features provisionally assigned to r_j in decreasing order of their value of the $Score$ function of Equation 1; and (iii) enforcing a "round robin" policy in which the n ranks take turns, for $\frac{x}{n}$ rounds, in picking their favourite features from the top-most elements of their rank-specific orderings.

Our third method, $RR(IGOR)$, is based on the idea of viewing ordinal classification on $R = \langle r_1 \prec \ldots \prec r_n \rangle$ as the generation of $(n-1)$ binary classifiers $\ddot{\Phi}_j$, each of which is in charge of separating $R_j = \{r_1, \ldots, r_j\}$ from $\overline{R}_j = \{r_{j+1}, \ldots, r_n\}$, for $j = 1, \ldots, (n-1)$. For each feature t_k we thus compute $(n-1)$ different values of $IG(t_k, c_j)$ (the classic *information gain* function of binary classification), by taking $c_j = r_1 \cup \ldots \cup r_j$ and $\overline{c}_j = r_{j+1} \cup \ldots \cup r_n$, for $j = 1, \ldots, (n-1)$. Similarly to $RR(Var{*}IDF)$, we (i) sort, for each of the $(n-1)$ binary classifiers $\ddot{\Phi}_j$, the features in decreasing order of $IG(t_k, c_j)$ value, and (ii) enforce a round-robin policy in which the $(n-1)$ classifiers $\ddot{\Phi}_j$ take turns, for $\frac{x}{n-1}$ rounds, in picking their favourite features from the top-most elements of their classifier-specific orderings.

Our fourth method, $RR(NC * IDF)$, directly optimizes the chosen error measure E. Let us define the *negative correlation* of t_k with r_j in the training set Tr as

$$NC_{Tr}(t_k, r_j) = \frac{\sum_{\{d_i \in Tr \mid t_k \in d_i\}} E(\tilde{\Phi}_j, d_i)}{|\{d_i \in Tr \mid t_k \in d_i\}|}$$

where $\tilde{\Phi}_j$ is the "trivial" classifier that assigns all $d_i \in D$ to r_j and $E(\hat{\Phi}, d_i)$ represents the error that $\hat{\Phi}$ makes in classifying d_i. Let the *rank* $R(t_k)$ *associated to a feature* t_k be

$$R(t_k) = \arg\min_{r_j \in R} NC_{Tr}(t_k, r_j)$$

*This is a short summary of a paper forthcoming in the Proceedings of the 25th ACM Symposium on Applied Computing (SAC'10), Sierre, CH, 2010.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

Dataset	$\|Tr\|$	$\|Va\|$	$\|Te\|$	1	2	3	4	5
TripAdvisor-15763	10,508	3,941	5,255	3.9%	7.2%	9.4%	34.5%	45.0%
Amazon-83713	20,000	4,000	63,713	16.2%	7.9%	9.1%	23.2%	43.6%

Table 1: Main characteristics of the two datasets used in this paper; the last five columns indicate the fraction of documents that have a given number of "stars".

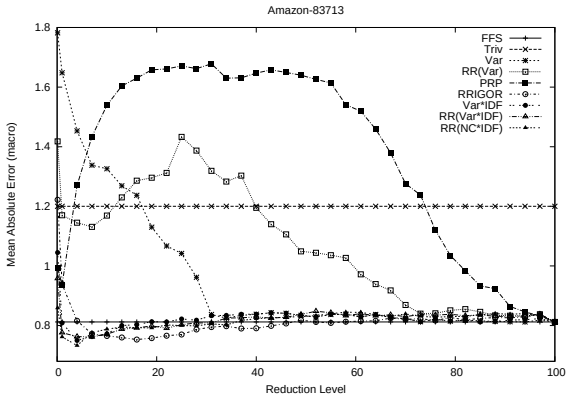


Figure 1: Results obtained on the Amazon-83713 dataset. Results are evaluated with MAE^M; lower values are better. "FFS" refers to the full feature set (i.e., no feature selection), while "Triv" refers to uniform assignment to the 3 Stars class.

We can now define

$$Score(t_k) = -NC_{Tr}(t_k, R(t_k)) * (IDF(t_k))^a \qquad (2)$$

where the a parameter serves the same purpose as in Equation 1. Similarly to the 2nd and 3rd methods, to select the best x features we apply a round-robin policy in which each r_j is allowed to pick, among the features such that $R(t_k) = r_j$, the $\frac{x}{n}$ features with the best $Score$.

More details on these methods can be found in [3].

3. EXPERIMENTS

We have tested the proposed measures on two different datasets, TripAdvisor-15763 and Amazon-83713, whose characteristics are summarized in Table 1. Both datasets consist of product reviews scored on a scale of one to five "stars", and (as shown by Table 1) are highly imbalanced. See [3] for more details.

As the evaluation measure we use the *macroaveraged mean absolute error* (MAE^M), proposed in [1] and defined as

$$MAE^M(\hat{\Phi}, Te) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{|Te_j|}\sum_{d_i \in Te_j}|\hat{\Phi}(d_i) - \Phi(d_i)| \qquad (3)$$

where Te_j denotes the set of test documents whose true rank is r_j and the "M" superscript indicates "macroaveraging".

We compare our methods with the three baselines mentioned at the end of Section 1 and with the "trivial baseline" that consists in scoring all test documents as 3 stars.

As a learning device we use *ϵ-support vector regression* (ϵ-SVR) [5] as implemented in the freely available LibSvm library. As a vectorial representation, after stop word removal (and no stemming) we use standard bag-of words with cosine-normalized $tfidf$ weighting. We have run all our experiments for all the 100 reduction levels $\xi \in \{0.001, 0.01, 0.02, 0.03, \ldots, 0.99\}$. For the $Var * IDF$, $RR(Var * IDF)$ and $RR(NC * IDF)$ methods we have (individually for each method) optimized the a parameter on a validation set Va extracted from the training set Tr, and then retrained the optimized classifier on the full training set Tr. For $RR(NC*IDF)$, $E(\hat{\Phi}, d_i)$ was taken to be $|\hat{\Phi}(d_i) - \Phi(d_i)|$.

The results of our tests are displayed in Figure 1, in which effectiveness is plotted as a function of the tested reduction level. For reasons of space only the Amazon-83713 results are displayed (see [3] for the TripAdvisor-15763 results, which are anyway fairly similar). The experiments show that

- the four novel techniques proposed here are dramatically superior to the four baselines;
- our four techniques are fairly stable across $\xi \in [0.05, 1.0]$, and deteriorate, sometimes rapidly, only for the very aggressive levels, i.e., for $\xi \in [0.001, 0.05]$). This is in stark contrast with the instability of the baselines.
- for $\xi \in [0.01, 0.3]$ the proposed techniques even outperform the full feature set. This indicates that one can reduce the feature set by an order of magnitude (with the ensuing benefits in terms of training-time and testing-time efficiency) and obtain an accuracy equal or even slightly superior (roughly a 10% improvement, in the best cases) to that obtainable with the full feature set.

4. REFERENCES

[1] S. Baccianella, A. Esuli, and F. Sebastiani. Evaluation measures for ordinal text classification. In *Proceedings of the 9th IEEE Int'l Conference on Intelligent Systems Design and Applications (ISDA'09)*, pages 283–287, Pisa, IT, 2009.

[2] S. Baccianella, A. Esuli, and F. Sebastiani. Multi-facet rating of product reviews. In *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, pages 461–472, Toulouse, FR, 2009.

[3] S. Baccianella, A. Esuli, and F. Sebastiani. Feature selection for ordinal regression. In *Proceedings of the 25th ACM Symposium on Applied Computing (SAC'10)*, Sierre, CH, 2010.

[4] R. Mukras, N. Wiratunga, R. Lothian, S. Chakraborti, and D. Harper. Information gain feature selection for ordinal text classification using probability re-distribution. In *Proceedings of the IJCAI'07 Workshop on Text Mining and Link Analysis*, Hyderabad, IN, 2007.

[5] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207—1245, 2000.

Integrating Named Entities in a Semantic Search Engine*

Annalina Caputo
University of Bari
Dept. of Computer Science
via E. Orabona, 4
Bari, Italy
acaputo@di.uniba.it

Pierpaolo Basile
University of Bari
Dept. of Computer Science
via E. Orabona, 4
Bari, Italy
basilepp@di.uniba.it

Giovanni Semeraro
University of Bari
Dept. of Computer Science
via E. Orabona, 4
Bari, Italy
semeraro@di.uniba.it

ABSTRACT

Traditional Information Retrieval (IR) systems are based on bag-of-words representation. This approach retrieves relevant documents by lexical matching between query and document terms. Due to synonymy and polysemy, lexical methods produce imprecise or incomplete results. In this paper we present how named entities are integrated in SENSE (SEmantic N-levels Search Engine). SENSE is an IR system that tries to overcome the limitations of the ranked keyword approach, by introducing *semantic levels* which integrate (and not simply replace) the lexical level represented by keywords. Semantic levels provide information about word meanings, as described in a reference dictionary, and named entities. Our aim is to prove that named entities are useful to improve retrieval performance.

1. BACKGROUND AND MOTIVATION

In recent years a lot of attention has been invested on Named Entities (NE), and their informative and discriminative power within documents. Due to the importance of research on NE, several sub-areas arose, such as entity detection and extraction, entity disambiguation and entity ranking. The typical information extraction task involving NE is Named Entity Recognition (NER). This task has been defined for the first time during the Message Understanding Conference (MUC) [4], and requires the identification and categorization of NE as entity names (for people and organization), place names, temporal expressions and numerical expressions. Named Entities play also a key role in the Information Retrieval context. Indeed, a very common task in that research area is the entity ranking, whose aim is to retrieve entities (rather than documents) that satisfy the user query. Most documents we deal on everyday contain a lot of references to persons, dates, monetary values and places. Moreover, named entity terms are among the most frequently searched terms on the Web. Statistics on Yahoo's top 10 search terms in 2008[1] showed that all the ten search terms consist of named entity terms: six persons, one sport organization, one role-playing game, one fictional character and one TV show.

In this paper we propose a new way of exploiting named entities in Information Retrieval. Named entities mentioned in a document constitute an important part of its semantics. However, when named entities are considered alone they may fail to capture the semantics expressed in a document or in a user query. For that reason we adopt an IR model, called *N-levels* [2], able to capture semantic information in a text by exploiting *word meanings*, described in a reference dictionary (e.g. WordNet), and named entities. Thus, we propose an IR system, called *SENSE* (SEmantic N-levels Search Engine), which manages documents indexed at multiple separate levels: keywords, senses (word meanings) and entities (named entities). The system is able to combine keyword search with semantic information provided by the two other indexing levels. Finally, we present the development of the full-fledged entity level based on a novel model called *Semantic Vectors*.

2. NAMED ENTITY LEVEL

Named entities are phrases that contain the names of persons, organizations, locations and, more generally, entities that can be identified by proper names. In order to identify named entities in a text, several methods can be applied such as Rule-based, Dictionary-based or Statistical ones. We adopted a statistical method exploiting YamCha[2], a generic open source text chunker useful for a lot of NLP tasks. YamCha adopts a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), introduced by Vapnik in 1995. We trained YamCha using the dataset provided by CoNLL-2003 organization during the Shared-Task 2003 [5]. The dataset contains entities extracted from Reuters dataset. In particular three types of entities are extracted: PERSON, LOCATION, ORGANIZATION and MISC, which contains entities that do not belong to the previous three categories. We extract entities from the CLEF 2008 collection [1]. The results of the entity recognition task are exported into a Lucene index. In detail, each document is split in two fields: HEADLINE and TEXT, in compliance with the document structure in CLEF. Each field contains the set of the recognized entities and, for each entity, the number of occurrences.

Building the entity level requires three steps:

1. **pre-processing and entity extraction:** XML files

*The full version appears in [3]

[1]http://buzz.yahoo.com/yearinreview2008/top10/

[2]http://chasen.org/ taku/software/YamCha/

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

provided by CLEF 2008 organizers are processed in order to extract entities. Named entities are stored in IOB2 format. In IOB2, words outside the Named Entity are tagged with O, while the first word in the entity is tagged with B-k (to begin class k), and further words receive the I-k tag, indicating that these words are inside the entity;

2. **entity indexing:** entities extracted in the previous step are stored into an index using Lucene. The entity extraction procedure allows to obtain an entity-based vector space representation, called bag-of-entities (BoE). In this model an entity vector, rather than a word vector, corresponds to a document.

3. **Semantic Vector building:** in this step semantic vectors are built by exploiting the Lucene index. The main idea behind models based on Semantic Vectors [6] is that words and concepts are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space. The SemanticVectors package offers tools for indexing a collection of documents and their retrieval. It relies on Apache Lucene to create a basic term-document matrix. Then the Lucene API is exploited to create a Wordspace model from the term-document matrix, by using Random Projection to perform *on-the-fly* dimensionality reduction. This is a relevant point because it allows us to use the same entity index produced in step 2 to induce semantic vectors. A detailed discussion on Semantic Vectors can be found in [6], whilst a thorough explanation about the entity index can be found in [3].

3. EXPERIMENTAL SESSION

For the evaluation of the system effectiveness, we used the CLEF Ad Hoc WSD-Robust dataset derived from the English CLEF data, which comprises corpora from "Los Angeles Times" and "Glasgow Herald", amounting to 166,726 documents and 160 topics in English and Spanish. The relevance judgments were taken from CLEF. The goal of the evaluation was to prove that the combination of three indexing levels outperforms a single level. In particular, that adding the entity level increases the effectiveness of the search with respect to the keyword and meaning levels. To evaluate system effectiveness, different runs were performed by exploiting a single level at a time, or a combination of two or more levels. Each experiment is identified by the names of the used levels. To measure retrieval performance, we adopted Mean-Average-Precision (MAP) and Geometric-Mean-Average-Precision (GMAP) calculated by *trec_eval 0.8.1*, a simple program supplied by the Text REtrieval Conference organizers[3], on the basis of 1,000 retrieved items per request. Table 1 shows the results for each run, with an overview on the exploited features.

The results confirm our hypothesis: named entity recognition, in conjunction with an IR model capable of expressing semantics, can greatly improve the retrieval performance. If evaluated individually, the entity level does not yield to satisfactory results. This result is due to the presence of topics in which no entity was recognized. Conversely, when search is performed by making use of multiple levels, the entity level is able to improve performance even on those (difficult) topics for which few relevant documents are returned. This result suggests that named entities play a key role in increasing the number of retrieved relevant results previously ignored. Specifically, considering the experiment *K+M+E* where we used all three levels, an improvement of 14.5% in the MAP and 33.3% in the GMAP was observed. Generally speaking, we noted an overall improvement in all the experiments that used the entity level, compared to the equivalent experiments in which that level was not exploited.

[3]http://trec.nist.gov/trec_eval/

Table 1: Results of the performed experiments

Run	*MAP*	*GMAP*
Keyword (K)	0.192	0.041
Meaning (M)	0.188	0.035
K+M	0.220	0.057
Entity (E)	0.134	0.006
K+E	0.220	0.048
M+E	0.228	0.054
K+M+E	**0.252**	**0.076**

4. REFERENCES

[1] E. Agirre, G. M. Di Nunzio, N. Ferro, T. Mandl, and C. Peters. CLEF 2008: Ad Hoc Track Overview. In *Working notes for the CLEF 2008 Workshop*, 2008.

[2] P. Basile, A. Caputo, A. L. Gentile, M. Degemmis, P. Lops, and G. Semeraro. Enhancing Semantic Search using N-Levels Document Representation. In S. Bloehdorn, M. Grobelnik, P. Mika, and D. T. Tran, editors, *Proceedings of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, volume 334 of *CEUR Workshop Proceedings*, pages 29–43. CEUR-WS.org, 2008.

[3] A. Caputo, P. Basile, and G. Semeraro. Boosting a semantic search engine by named entities. In J. Rauch, Z. W. Ras, P. Berka, and T. Elomaa, editors, *ISMIS - Foundations of Intelligent Systems, 18th International Symposium, ISMIS 2009, Prague, Czech Republic, September 14-17, 2009. Proceedings*, volume 5722 of *Lecture Notes in Computer Science*, pages 241–250. Springer, 2009.

[4] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *COLING*, pages 466–471, 1996.

[5] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[6] D. Widdows and K. Ferraro. Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

New Research Directions in Search Results Clustering

Claudio Carpineto, Andrea Bernardini,
Massimiliano D'Amico, Gianni Romano
Fondazione Ugo Bordoni
Rome, Italy
{carpinet, abernardini, romano}@fub.it
mas.damico@gmail.com

ABSTRACT

We discuss which are the main research themes in the field of search results clustering and report some recent results achieved by the Information Mining group at Fondazione Ugo Bordoni.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

1. SEARCH RESULTS CLUSTERING

Search results clustering organizes search results by topic, thus providing a complementary view to the flat list returned by document ranking systems. This approach is especially useful when document ranking fails. Besides allowing direct subtopic access, search results clustering reduces information overlook, helps filtering out irrelevant items, and favors exploration of unknown or dynamic domains.

Search results clustering is related to, but distinct from, conventional document clustering. When clustering takes place as a post-processing step on the set of results retrieved by an information retrieval system on a query, it may be both more efficient, because the input consists of few hundred of snippets, and more effective, because query-specific text features are used. On the other hand, search results clustering must fulfill a number of more stringent requirements raised by the nature of the application in which it is embedded; e.g., meaningful cluster labels, low response times, short input data description, unknown number of clusters, overlapping clusters.

A comprehensive survey of search results clustering, including issues, techniques, and systems is given in [4]. In the remainder of this paper we point out interesting research directions.

1.1 Description-centric clustering algorithms

Given that search results clustering systems are primarily intended for browsing retrieval, a critical part is the quality of cluster labels, as opposed to optimizing only the clustering structure. In fact, the algorithms for performing search results clustering cover a spectrum ranging from data-centric to description-centric techniques, depending on whether the priority is given to cluster formation or cluster labeling.

One of the most recent examples of the latter category is KeySRC (Keyphrase-based Search Results Clustering), described in [1]. This system generates clusters labeled by keyphrases. The keyphrases are extracted from the generalized suffix tree built from the search results and merged through an improved hierarchical agglomerative clustering procedure, representing each phrase as a weighted document vector and making use of a variable dendrogram cut-off value. KeySRC is available at http://keysrc.fub.it.

1.2 Performance evaluation measures

Internal validity measures and comparison with ground truth results are two common ways of evaluating clustering partitions, but they have the disadvantage that the performance of the system in which the document partition is encompassed is not explicitly taken into account. As the intended use of search results clustering is to find documents relevant to the single query's subtopic, it may be more convenient to evaluate the performance on a retrieval oriented task. However, the classical measures related to subtopic retrieval, such as subtopic recall, subtopic precision, and subtopic MRR, assume that the system output consists of a ranked list and thus they are not directly or easily applicable to clustered results, Furthermore, they strictly focus on subtopic coverage; i.e., retrieving at least one relevant document per subtopic.

To address these limitations, we presented a new evaluation measure inspired by Cooper's expected search length: *Subtopic Search Length under k document sufficiency* (kSSL). The idea is to consider the number of elements (cluster labels or search results) that the user must examine to retrieve a specified number (k) of documents relevant to the single subtopics of a query. The shorter the search length, the better the system performance. It is assumed that both cluster labels and search results are read sequentially from top to bottom, and that only cluster with labels relevant to the subtopic at hand are opened. The main advantages of kSSL are that it is suitable for both ranked lists and clustered results and that it allows evaluation of full subtopic retrieval (i.e., retrieval of multiple documents relevant to a query's subtopic). A full description of kSSL is given in [1].

1.3 Test collections

There is almost a complete lack of test collections with subtopic relevance judgments. Two exceptions are the collections developed at the TREC Interactive track, which is

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

small and primarily focuses on the instances of a given concept (e.g., 'what tropical storms – hurricanes and typhoons – have caused property damage and/or loss of life'), and at Image CLEF, which is mainly about geographical diversity of photos associated with a given topic (e.g., 'images of beaches in Brazil').

We created two new test collections for evaluating subtopic retrieval, namely AMBIENT and ODP-239. AMBIENT (AMBIguous ENTries) consists of 44 topics extracted from the *ambiguous* Wikipedia entries, each with a set of subtopics and a list of 100 ranked search results manually annotated with subtopic relevance judgments. AMBIENT is fully described in [3] and is available at http://credo.fub.it/ambient.

ODP-239 consists of 239 topics, each with about 10 subtopics and 100 documents associated with the subtopics. The topics, subtopics, and their associated documents were selected from the Open Directory Project (www.dmoz.org). The distribution of documents across subtopics reflects the relative importance of subtopics. ODP-239 can be downloaded from http://credo.fub.it/odp239.

1.4 Applications in mobile search

The features of search results clustering appear very suitable for mobile information retrieval, where a minimization of user actions (such as scrolling and typing), device resources, and amount of data to be downloaded are primary concerns. Furthermore, such features seem to nicely comply with the most recently observed usage patterns of mobile searchers.

We implemented two mobile clustering engines (for PDAs and cellphones) and evaluated their retrieval performance [3]. We found that mobile clustering engines can be faster and more accurate than the corresponding mobile search engines, especially for subtopic retrieval tasks. We also found that although mobile retrieval becomes, in general, less effective as the search device gets smaller, the adoption of clustering may help expand the usage patterns beyond mere informational search while mobile.

1.5 Meta search results clustering

Just as the results of several search engines can be combined into a meta search engine, the outputs produced by distinct clustering engines can be merged into a meta clustering engine. Currently, there are many different web clustering engines but no attempts has still been made to combine them, to the best of our knowledge.

We studied the problem of meta search results clustering, that has unique features with respect to the relatively well understood field of general meta clustering. After showing that the combination of multiple search results clustering algorithms is empirically justified, we developed a novel meta clustering algorithm that maximizes the agreement between the outputs produced by the input clustering algorithms [5]. The novel meta clustering algorithm applied to web search results is both efficient and effective.

1.6 Clustering versus diversification of search results

Re-ranking search results to promote diversity of top elements is another approach to subtopic retrieval that has received much attention lately. Clustering and diversification of search results are thus different techniques with a similar goal, i.e., addressing the limitations of the probabilistic ranking principle when a topic has multiple aspects of potential interest and the relevance criterion alone is not sufficient.

These two techniques have not been compared so far. We performed a systematic evaluation of several clustering and diversification algorithms using multiple test collections and evaluation measures [2]. It turns out that diversification works well when one wants to get a quick overview of documents relevant to distinct subtopics, whereas clustering is more useful when one is interested in retrieving multiple documents relevant to each subtopic.

1.7 Other research directions

There are further directions that have started to be explored recently by other research groups. They mainly aim to improve the quality and effectiveness of the search results clustering process. A non-exhaustive list is given below.

– Personalized search results clustering
– Integrating external knowledge (e.g., thesauri, metadata, folksonomies, past queries) with search results clustering
– Semi-supervised search results clustering
– Temporal search results clustering
– Visualization of clustered search results
– Search results clustering and faceted hierarchies

2. REFERENCES

[1] A. Bernardini, C. Carpineto, and M. D'Amico. Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. In *Proceedings of 2009 IEEE/WIC/ACM International Conference on Web Intelligence, Milan, Italy*, pages 206–213. IEEE Computer Society, 2009.

[2] C. Carpineto, M. D'Amico, and G. Romano. Evaluating subtopic retrieval system performance: clustering versus diversification. Submitted.

[3] C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero. Mobile Information Retrieval with Search Results Clustering: Prototypes and Evaluations. *Journal of American Society for Information Science and Technology (JASIST)*, 60(5):877–895, 2009.

[4] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of Web clustering engines. *ACM Computing Survey*, 41(3), 2009.

[5] C. Carpineto and G. Romano. Optimal Meta Search Results Clustering. Submitted.

A Visualization Tool of Probabilistic Models for Information Access Components*

Giorgio Maria Di Nunzio
Dept. of Information Engineering
University of Padua
Via Gradenigo 6/a, 35131
Padua, Italy
dinunzio@dei.unipd.it

ABSTRACT

An effective graphic interface is a key tool to improve the fruition of the results retrieved by an Information Retrieval (IR) system. In this work, we describe a two-dimensional interface that represents the documents ranked on a Cartesian space and allows the user to interact with the documents in order to improve the results of the search engine. Results are classified and ranked according to the best separating line of the two classes of documents: relevant and non relevant documents. Mathematical tools such as least squares distances are used to train the supervised algorithm that finds the separating and ranking lines.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Relevance feedback, Retrieval models, Search process*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*

General Terms

Algorithms, Design, Experimentation

Keywords

Information Visualization, Machine Learning, Naïve Bayes Models, Relevance Feedback

1. INTRODUCTION

Visualization is the process of transforming data, information, and knowledge into graphic presentations to support tasks such as data analysis and information exploration. The definition of a spatial structure for information visualization is challenging because data in an information space may be multi-faceted, relationships of data are interwoven and are complicated. Moreover, the definition of such a space means a complex process of extracting displayable attributes from objects, organizing the information, projecting objects onto the structure, and synthesizing search features, objects and object relationships into the visual space [5].

The introduction of visualization environments may add cognitive processes to the user who needs to understand and learn the characteristics of the new environment and interact with them to get the best from the system. In fact, the aim of visualization environments, as external representation of the world of interest, is to reduce the amount of cognitive effort required to solve informationally equivalent problems [4]. In particular, an IR system should provide users an environment in which they can exploit their skills to maximize their cognitive abilities. The visualization of an IR system is nothing but a process that transforms invisible abstract data and their semantic relationships in a visible collection on a display in order to find the user information need more easily.

In this paper, we present the design and implementation a tool for the visualization of Naïve Bayes (NB) probabilistic models for information access components that represents digital objects on the two-dimensional space [2, 3, 1]. The demonstration will applied to the task of automatic text classification and text retrieval.

2. DESIGN

The model which upholds the visualization tool defines a direct relationship between the probability of an object given a category of interest and a point on a two-dimensional space. In this light, it is possible to graph entire collections of objects on a Cartesian plane, and to design algorithms that categorize and retrieve documents directly on this two-dimensional representation. This tool demonstrates to be a valid visualization tool also for understanding the relationships between categories of objects.

The design of the two-dimensional visualization tool follows two main requirements:

- for end-user, the interface should give the opportunity to define the query with simple or advanced options, and to express judgements for the documents retrieved which will be used to re-rank documents;
- for researchers, the interface should display the decisions taken by the search engine in terms of separating line and explain how the relevance feedback given by the user affects the list of ranked documents.

The interface offers the possibility to write free text queries, as any other search engine, or load predefined queries; predefined queries are used for research purposes and recreates

*This is an extended abstract of [1]

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

the environment of evaluation tasks organized by campaign such as TREC[1] or CLEF[2].

The interface associate each document of the collection to a point in the two-dimensional space according to a probabilistic algorithm: the abscissa reflects how much the document is relevant to the query, the ordinate reflects how much the document is not relevant to the query. The pair of numbers gives an indication of the fraction of relevance for that particular document given the query, this pair is plotted on a frame and the relative position of this point with respect to the other documents in the collection determines its position in the list of ranked documents.

In the two-dimensional representation of documents, the equation of the ranking or the classification function has to be written in such a way that each coordinate of a document is the sum of two addends: a variable component $\mathrm{P}(d|c_i)$, the probability of a document d given a category of interest c_i, and a constant component $\mathrm{P}(c_i)$, the prior of the category of interest c_i [3] For example, in the case of NB models the equation becomes:

$$\underbrace{\log\left(\mathrm{P}(d|c_i)\right)+\log\left(\mathrm{P}(c_i)\right)}_{X_i(d)} > \underbrace{\log\left(\mathrm{P}(d|\bar{c}_i)\right)+\log\left(\mathrm{P}(\bar{c}_i)\right)}_{Y_i(d)}$$

When the inequality holds, the document is considered an element of category c_i. If c_i and $\bar{c}_i$ are considered respectively the set of relevant documents and the set of non relevant documents, we can divide the collection of documents in these two sets; if we are only interested in the ranking of documents, we can compute the list of retrieved documents by combining the two components into one *relevance weight*.

Documents can be classified or ranked differently according to the Focused Angular Region algorithm which computes the best separating (or ranking) line by means of regression techniques and least squares orthogonal, and vertical, distances. Information about the categories of documents are collected during the interaction of the user with the interface; in particular, the relevance judgements that the user expresses for the documents are used to re-compute the probabilities and train the algorithm (details of this supervised algorithm are given in [3]). This part can be done automatically by selecting in the interface the option "Blind relevance feedback", which takes the first n documents of the current list of documents and set them as relevant.

3. RESULTS AND OPEN QUESTIONS

This visualization tool was tested on standard benchmark collections and a demonstration was presented at [1] in order to answer the following research questions: how well the ranking or classification functions are learned from the data as separating lines; how particular unbalanced distribution of documents can be corrected by means of parameter estimation; how the multivariate model and the multinomial model perform on different languages; how blind and/or explicit relevance feedback affect ranking list, and how the selection of relevant documents changes the shape of the clouds of relevant and non-relevant documents.

During the interaction with the system, new questions and new research ideas were collected about advances types of interaction: changing the estimated probability of terms directly; smoothing parameters in order to *see* how the clouds

[1]http://trec.nist.gov/

[2]http://www.clef-campaign.org

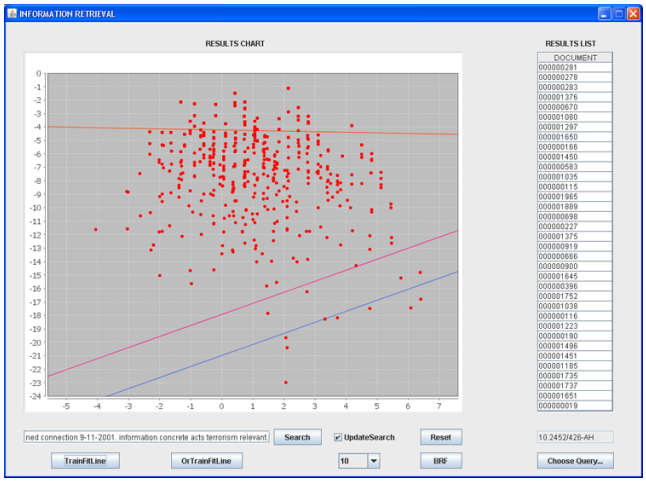


Figure 1: An example of the interface used by researchers.

of points move in the space and how the performance changes accordingly; drawing the clouds of points incrementally, highlighting the contribution of each term to understand which terms better discriminate the two sets of points.

In Figure 1, a screen-shot of the main window of the visualization tool si shown. The example shows the interface used by researchers. The different separating lines are calculated for a blind relevance feedback of 10 documents: the category of relevant documents in blue, the category of non relevant documents in red, the best separating line in purple. The list of retrieved documents is presented on the right. The user can choose to select a document, read it, and judge it as relevant or non relevant. This information is stored and used to train the supervised algorithm when the user selects the "update search" box.

4. REFERENCES

[1] L. De Stefani, G. M. Di Nunzio, and G. Vezzaro. A visualization tool of probabilistic models for information access components. In *Proceedings of Research and Advanced Technology for Digital Libraries (ECDL 2009)*, Corfu, Greece, September/October 2009. LNCS, Springer.

[2] G. M. Di Nunzio. Visualization and Classification of Documents: A New Probabilistic Model to Automated Text Classification. *Bulletin of the IEEE Technical Committee on Digital Libraries (IEEE-TCDL)*, 2(2), 2006.

[3] G. M. D. Nunzio. Using Scatterplots to Understand and Improve Probabilistic Models for Text Categorization and Retrieval. *Journal of Approximate Reasoning*, 50(7):945–956, July 2009. http://dx.doi.org/10.1016/j.ijar.2009.01.002.

[4] M. Scaife, M. Scaife, Y. Rogers, and Y. Rogers. External cognition: how do graphical representations work? *International Journal of Human-Computer Studies*, 45:185–213, 1996.

[5] J. Zhang. *Visualization for Information Retrieval*, volume 23 of *The Information Retrieval Series*. Springer, 2008. ISBN: 978-3-540-75147-2.

Developing the Quantum Probability Ranking Principle

Guido Zuccon
Department of Computing Science
University of Glasgow
Scotland (UK)
guido@dcs.gla.ac.uk

Leif Azzopardi
Department of Computing Science
University of Glasgow
Scotland (UK)
leif@dcs.gla.ac.uk

ABSTRACT

In this work, we summarise the development of a ranking principle based on quantum probability theory, called the Quantum Probability Ranking Principle (QPRP), and we also provide an overview of the initial experiments performed employing the QPRP. The main difference between the QPRP and the classic Probability Ranking Principle, is that the QPRP implicitly captures the dependencies between documents by means of "quantum interference". Subsequently, the optimal ranking of documents is not based solely on documents' probability of relevance but also on the interference with the previously ranked documents. Our research shows that the application of quantum theory to problems within information retrieval can lead to consistently better retrieval effectiveness, while still being simple, elegant and tractable.

1. INTRODUCTION

The idea of using quantum theory in information retrieval (IR) was formally put forward by van Rijsbergen [9] in 2004[1]. In [9], the main thesis of this seminal book is to use quantum theory as a bridge between the three mainstream IR approaches; i.e. vector space models, logic models and probability models. While this direction has been largely unexplored, recently there has been a spate of work which aims to develop quantum inspired or quantum based information retrieval models [1, 2, 3, 4, 5, 6, 8, 7, 13, 11].

In this work, we report on the the development the Quantum Probability Ranking Principle [14, 12]. The ranking principle is derived by developing an *analogy* between the famous double-slit experiment and document ranking. The double slit experiment was conducted to demonstrate that kolmogorovian probability fails to adequately describe the outcome of physical phenomena, and this motivated the development of quantum probability theory which incorporates the quantum interference between events.

In [14], it is hypothesized that this quantum interference can be used to account for the interdependence between documents and their associated judgements. In certain tasks, the relevance of a document may depend on the previous documents already assessed, for example in the novelty and diversity tracks. In sub-topic retrieval, the IR system has to provide a document ranking which covers all the possible facets (subtopics) relevant to the user's information need as soon as possible in the ranking. Consequently, following the traditional Probability Ranking Principle, where document dependence is ignored, leads to sub-optimal performance [10][2]. In [12], we perform a series of experiments that also indicate this is the case, and further show that the QPRP leads to better empirical performance. This is because within the QPRP the interdependence between documents is naturally accounted for through the *quantum interference*, and the QPRP suggests that documents ranked until position $n-1$ interfere with the degree of relevance of the document ranked at position n. Intuitively, documents expressing diverse information have higher degree of interference than documents that are similar. For the same reason, documents containing novel information might strongly interfere with documents ranked in previous positions. Even contrary information might be captured by the interference term: documents containing content contrary to the one presented at the previous rank positions might trigger a revision of user's beliefs about the topic.

The remainder of this paper is as follows: the next section briefly outlines the QPRP. Section 3 presents the main results from the study recently performed on sub-topic retrieval. Finally, we conclude in Section 4 by outlining the directions for further work using the quantum based ranking principle.

2. THE QPRP

In [14], the Quantum Probability Ranking Principle is proposed and it's derivation is based on an analogy with the famous double slit experiment. The resultant of this work was the following formulation: when ranking documents, the IR system has to maximise the total satisfaction of the user given the document ranking, achievable by maximising the total probability of the ranking. Using the quantum law of total probability, the resultant ranking strategy impose to select at each rank position a document d such that

$$d = \arg\max \left(P(d_i) + \sum_{d_x \in RA} I_{d_x,d_i} \right) \quad (1)$$

where RA is the list of documents already ranked and I_{d_x,d_i} is the interference between documents d_x and d_i. Note that

[1]Prior to this, van Rijsbergen gave talks as early as 1996 on the topic.

[2]This has led to arguments for the development of a new ranking principle.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

the traditional PRP is equivalent to the QPRP when the interference is null, i.e. $I_{d_x,d_i} = 0, \forall d_x, d_i \in C$, the documents corpus. In physics, the interference indicates the amount and kind of interaction between waves. If two waves strongly interact with each other, then the absolute value of their interference is high, and vice versa low. The interaction can generate two different outcomes: either increase the effect generated by the sum of the two waves (constructive interference, $I > 0$) or decrease it (destructive interference, $I < 0$). In IR, the interference I_{d_x,d_i} could be negative or positive, and thus demote or promote a document in the ranking depending on the context. For instance in sub-topic retrieval it would be sensible if documents related to the same subtopics negatively interfere, lowering the chances to rank both of them at high positions. This scenario is discussed in the next section.

3. THE QPRP IN SUBTOPIC RETRIEVAL

In [12], the QPRP is empirically tested and validated on the subtopic retrieval task. The ranking under the QPRP was compared with the rankings of models which upholds the PRP, and also against state-of-the-art strategies for subtopic retrieval, i.e. MMR and Portfolio Theory (PT) [10].

The main point of this experimentation was to determine whether the inherent document interdependence could be accounted for by the interference component within the QPRP. Intuitively, the interference component depends upon both the inter-document dependencies and the document's relevance probabilities. Since it is not possible to estimate the interference component directly from the text statistics, for the experiments reported in [12], we have used the Pearson's correlation between interfering documents to compute the interference. We performed the empirical investigation over the TREC subtopic retrieval track, which includes documents from the Financial Times of London contained in TREC 6,7 and 8 collections and 20 ad-hoc retrieval topics, composed of subtopics, from the TREC interactive tracks. We retrieved documents and generated the initial probability distribution using Okapi BM25: this represented the PRP ranking. Afterwards we re-ranked the documents according to three different strategies: our QPRP method and two state-of-the-art techniques for subtopic retrieval, i.e. MMR and PT, which required parameters tuning. The experiments were repeated varying the level of retrieval cut-off and the length of the queries.

From the experimental results[3], we found that (1) the QPRP improves upon PRP baselines for all levels of S-precision and S-recall, (2) the QPRP outperforms MMR and PT across most levels, (3) the QPRP consistently outperforms other strategies across all topics when considering S-MRR@100%, meaning that on each topic the QPRP returns complete coverage of all subtopics at a rank lower than all the other strategies. And, unlike MMR and PT, no tuning or training is required!

4. CONCLUSIONS

In this paper we have reported about the recent introduction of a novel ranking strategy, the QPRP, based on quantum probability and inspired by an analogy with the double slits experiment in physics. The QPRP naturally encodes the interdependence between documents through quantum interference. The new ranking strategy has been empirically investigated, showing that the QPRP consistently outperforms both the PRP and state-of-the-art approaches, i.e. MMR and PT, without requiring parameter tuning. This suggests that the use of Quantum Theory to model processes within information retrieval can lead to substantial improvements in retrieval effectiveness.

Future work examining the utility and applications of the Quantum Probability Ranking Principle will be directed towards:

- impact of the Pearson's correlation coefficient as a mean to approximate interference;
- alternative estimations of the interference;
- how to derive a complex amplitude distribution from the document corpus;
- the relationships between interference in the quantum probability framework and conditional probabilities in Kolmogorovian probability theory; and,
- how to apply the QPRP paradigm to other retrieval tasks, e.g. ad-hoc retrieval.

Acknowledgements: We would like to thank Keith van Rijsbergen for his collaboration, support and mentoring, and Massimo Melucci for his comments and suggestions. This work has been funded by the EPSRC Renaissance project (EP/F014384/1) and the Royal Society International Joint Project JP080734.

[3]Experimental results are available online at `http://www.dcs.gla.ac.uk/~guido/qprpresults.html`

5. REFERENCES

[1] S. Arafat. *Foundations research in information retrieval inspired by quantum theory*. PhD thesis, University of Glasgow, December 2007.

[2] S. Arafat and C. J. van Rijsbergen. Quantum theory and the nature of search. In *QI '07*, pages 114–121, 2007.

[3] C. Flender, K. Kitto, and P. Bruza. Beyond ontology in information systems. In *QI 2009*, pages 276–288, 2009.

[4] Y. Hou and D. Song. Characterizing pure high-order entanglements in lexical semantic spaces via information geometry. In *QI 2009*, pages 237–250, 2009.

[5] A. F. Huertas-Rosero, L. A. Azzopardi, and C. J. van Rijsbergen. Eraser lattices and semantic contents. In *QI 2009*, pages 266–275, 2009.

[6] M. Melucci. A basis for information retrieval in context. *ACM TOIS*, 26(3):1–41, June 2008.

[7] B. Piwowarski and M. Lalmas. A quantum-based model for interactive information retrieval. In *ICTIR '09*, pages 224–231, 2009.

[8] B. Piwowarski and M. Lalmas. Structured information retrieval and quantum theory. In *QI '09*, pages 289–298, March 2009.

[9] C. J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.

[10] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR '09*, pages 115–122, 2009.

[11] G. Zuccon, L. Azzopardi, and C. J. van Rijsbergen. Revisiting logical imaging for information retrieval. In *SIGIR '09*, pages 766–767, 2009.

[12] G. Zuccon and L. A. Azzopardi. Using the Quantum Probability Ranking Principle to Rank Interdependent Documents. In *ECIR 2010*, 2010. to appear.

[13] G. Zuccon, L. A. Azzopardi, and C. J. van Rijsbergen. A formalization of logical imaging for information retrieval using quantum theory. In *DEXA '08*, pages 3–8, 2008.

[14] G. Zuccon, L. A. Azzopardi, and K. van Rijsbergen. The quantum probability ranking principle for information retrieval. In *ICTIR '09*, pages 232–240, 2009.

An Empirical Comparison of Collaborative Filtering Approaches on Netflix Data

Nicola Barbieri, Massimo Guarascio, Ettore Ritacco
ICAR-CNR
Via Pietro Bucci 41/c, Rende, Italy
{barbieri,guarascio,ritacco}@icar.cnr.it

ABSTRACT

Recommender systems are widely used in E-Commerce for making automatic suggestions of new items that could meet the interest of a given user. Collaborative Filtering approaches compute recommendations by assuming that users, who have shown similar behavior in the past, will share a common behavior in the future. According to this assumption, the most effective collaborative filtering techniques try to discover groups of similar users in order to infer the preferences of the group members. The purpose of this work is to show an empirical comparison of the main collaborative filtering approaches, namely *Baseline*, *Nearest Neighbors*, *Latent Factor* and *Probabilistic* models, focusing on their strengths and weaknesses. Data used for the analysis are a sample of the well-known Netflix Prize database.

Categories and Subject Descriptors

H.2.8 [**Database Application**]: Data Mining

Keywords

Recommender Systems, Collaborative Filtering, Netflix

1. INTRODUCTION

The exponential growth of products, services and information makes fundamental the adoption of intelligent systems to guide the navigation of the users on the Web. The goal of *Recommender Systems* is to profile a user to suggest him contents and products of interest. Such systems are adopted by the major E-commerce companies, for example Amazon.com [1], to provide a customized view of the systems to each user. Usually, a recommendation is a list of items, that the system considers the most attractive to customers. User profiling is performed through the analysis of a set of users' evaluations of purchased/viewed items, typically a numerical score called *rating*. Most recommender systems are based on *Collaborative Filtering (CF)* techniques [6], which analyze the past behavior of the users, in terms of previously given ratings, in order to foresee their future choices and discover their preferences. The main advantage in using CF techniques relies on their simplicity: only users' past ratings are used in the learning process, no further informations, like demographic data or item descriptions, are needed (techniques that use this knowledge are called *Content Based* [10, 14]). Four different families of techniques have been studied: *Baseline*, *Neighborhood based*, *Latent Factor* analysis and *Probabilistic* models. This work aims to show an empirical comparison of a set of well-known approaches for CF, in terms of quality prediction, over a real (non synthetic) dataset. Several works have focused on the analysis and performance evaluation of single techniques (i.e. excluding ensemble approaches), but at the best of our knowledge there is no previous work that performed such a deep analysis comparing different approaches.

[1]http://amazon.com/

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. BACKGROUND

The following notation is used: u is a user, m is a movie, $\hat{r}_m^u$ is the rating (stored into the data set) expressed by the user u with respect to the movie m (zero if missing), and given a CF model, r_m^u is the predicted rating of the user u for the movie m. On October 2006, Netflix[2], leader in the movie-rental American market, released a dataset containing more of 100 million of ratings and promoted a competition, the Netflix Prize [3], whose goal was to produce a 10% improvement on the prediction quality achieved by its own recommender system, *Cinematch*. The competition lasted three years and was attended by several research groups from all over the world. The dataset is a set of tuple $(u, m, \hat{r}_m^u)$ and the model comparison is performed over a portion of the entire Netflix data [4]. This portion is a random sample of the data, and is divided into two sets: a training set $\mathcal{D}$ and a test set $\mathcal{T}$. $\mathcal{D}$ contains $5,714,427$ ratings of $435,659$ users on $2,961$ movies, $\mathcal{T}$ consists of $3,773,781$ ratings (independent from the training set) of a subset of training users $(389,305)$ on the same set of movies. The evaluation criterion chosen is the **Root Mean Squared Error (RMSE)**:

$$RMSE = \sqrt{\frac{\sum_{(u,m) \in \mathcal{T}} \left(r_m^u - \hat{r}_m^u\right)^2}{|\mathcal{T}|}} \qquad (1)$$

Cinematch achieves (over the entire Netflix test set) an RMSE value equals to 0.9525, while the team BellKor's Pragmatic Chaos, that won the prize, achieved a RMSE of 0.8567. This score was produced using an ensemble of severeal predictors.

[2]http://www.netflix.com/
[3]http://www.netflixprize.com/
[4]http://repository.icar.cnr.it/sample_netflix/

3. COLLABORATIVE FILTERING MODELS

Studied models belong to four algorithm families: Baseline, Nearest Neighbor, Latent Factor and Probabilistic models. A detailed description of all the analyzed techniques follows.

3.1 Baseline Models

Baseline algorithms are the simplest approaches for rating prediction. This section will focus on the analysis of the following algorithms: *OverallMean*, *MovieAvg*, *UserAvg*, *DoubleCentering*. *OverallMean* computes the mean of all ratings in the training set, this value is returned as prediction for each pair (u, m). *MovieAvg* predicts the rating of a pair (u, m) as the mean of all ratings received by m in the training set. Similarly, *UserAvg* predicts the rating of a pair (u, m) as the mean of all ratings given by u. Given a pair (u, m), *DoubleCentering* compute separately the mean of the ratings of the movie r_m, and the mean of all the ratings given by the user r_u. The value of the prediction is a linear combination of these means:

$$r_m^u = \alpha\, r_m + (1 - \alpha)\, r_u \tag{2}$$

where $0 \leq \alpha \leq 1$. Experiments on $\mathcal{T}$ have shown that the best value for α is 0.6 (see Fig. 1).

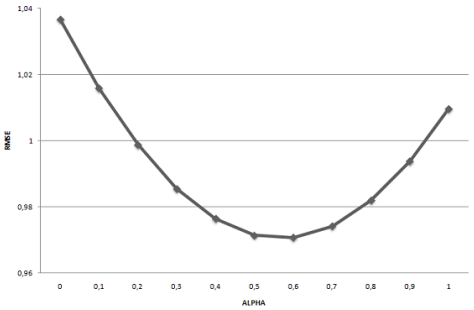


Figure 1: RMSE vs. α

3.2 Nearest Neighbor models

Neighborhood based approaches compute the prediction basing on a chosen portion of the data. The most common formulation of the neighborhood approach is the *K-Nearest-Neighbors (K-NN)*. r_m^u is computed following simple steps. A similarity function associates a numerical coefficient to each pair of user, then K-NN finds the *neighborhood* of u selecting the K most similar users to him, said *neighbors*. The rating prediction is computed as the average of the ratings in the *neighborhood*, weighted by the similarity coefficients. User-based K-NN algorithm is intuitive but doesn't scale because it requires the computation of similarity coefficients for each pair of users. A more scalable formulation can be obtained considering an *item-based* approach [15]: the predicted rating for the pair (u, m) can be computed by aggregating the ratings given by u on the K most similar movies to m: $\{m_1, \ldots, m_K\}$. The underlying assumption is that the user might prefer movies more similar to the ones he liked before, because they share similar features. In this approach the number of similarity coefficients (respectively $\{s_1, \ldots, s_K\}$) depends on the number of movies which is much smaller than the number of users. The prediction is computed as:

$$r_m^u = \frac{\sum_{i=1}^{K} s_i\, r_{m_i}^u}{\sum_{i=1}^{K} s_i} \tag{3}$$

In the rest of the paper, only item-based K-NN algorithms will be considered. The similarity function plays a central role : its coefficients are necessary for the identification of the neighbors and they act as weights in the prediction. Two functions, commonly used for CF, are *Pearson Correlation* and *Adjusted Cosine* [15] coefficients: preliminary studies proved that Pearson Correlation is more effective in detecting similarities than Adjusted Cosine. Moreover as discussed in [9], similarity coefficients based on a larger support are more reliable than the ones computed using few rating values, so it is a common practice to weight the similarity coefficients using the support size, technique often called *shrinkage*. Shrinkage is performed as follows. Let $U(m_i, m_j)$ be the set of users that rated movies m_i and m_j, and let s_{m_i,m_j} be the similarity coefficient between these two movies:

$$s'_{m_i,m_j} = \frac{s_{m_i,m_j}\ |U(m_i, m_j)|}{|U(m_i, m_j)| + \alpha} \tag{4}$$

Where α is an empirical value. Experiments showed that the best value for α is 100, so in the following K-NN algorithms with Pearson Correlation and shrinkage with $\alpha = 100$ will be considered. This first model will be called *SimpleK-NN*. An improved version can be obtained considering the difference of preference of u with respect to the movies in the neighborhood ($\{m_1, \ldots, m_K\}$) of m. Formally:

$$r_m^u = b_m^u + \frac{\sum_{i=1}^{K} s_i\, (\hat{r}_{m_i}^u - b_{m_i}^u)}{\sum_{i=1}^{K} s_i} \tag{5}$$

Where $\{s_1, \ldots, s_K\}$ are the similarity coefficients between m and its neighbors, b_m^u and $b_{m_i}^u$ are baseline values computed using Eq. 2. In this case the model is named *BaselineK-NN*, otherwise, if the baseline values are computed according to the so called *User Effect Model* [2], the model will be called *K-NN (user effect)*. An alternative way to estimate item-to-item interpolation weights is by solving a least squares problem minimizing the error of the prediction rule. This strategy, proposed in [1, 3], defines the *Neighborhood Relationship Model*, one of the most effective approaches applied during the Netflix prize. r_m^u is computed as:

$$r_m^u = \sum_{i=1}^{K} w_{m_i}^m\, \hat{r}_{m_i}^u \tag{6}$$

Where m_i is a generic movie in the neighborhood of m, and $w_{m_i}^m$ are weights representing the similarity between m and m_i computed as the solution of the following optimization problem:

$$min_w \sum_{v \neq u} \left(r_{m_i}^v - \sum_{j=1}^{K} w_{m_i}^m\, \hat{r}_{m_j}^v \right)^2 \tag{7}$$

Fig. 2 shows the behaviors of K-NN models with different values of K. Best performances are achieved by the *Neighborhood Relationship Model*.

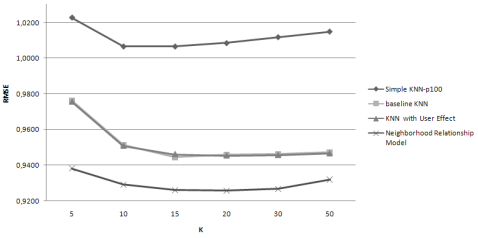


Figure 2: RMSE vs. α

3.3 Latent Factor Models via Singular Value Decomposition (SVD)

The assumption behind Latent Factor models is that the rating value can be expressed considering a set of contributes which represent the interaction between the user and the target item on a set of features. Let A be a matrix $[|users| \times |movies|]$, $A_{u,m}$ is equal to the rank chosen by the user u for the movie m. A can be approximated as the product between two matrices: $A \approx U \times M$, where U is a matrix $[|users| \times K]$ and M is a matrix $[K \times |movies|]$, K is an input parameter of the model and represents the number of features to be considered. Intuitively, A is generated by a combination of users (U) and movies (M) with respect to a certain number of features. Fixed the number of features K, SVD algorithms try to estimate the values within U and M, and give the prediction of r_m^u as:

$$r_m^u = \sum_{i=1}^{K} U_{u,i}\, M_{i,m} \tag{8}$$

where $U_{u,i}$ is the response of the user u to the feature i, and $M_{i,m}$ is the response of the movie m on i. Several approaches have been proposed to overcome the sparsity of the original rating matrix A and to determine a good approximation solving the following optimization problem:

$$(U, M) = \arg\min_{U,M} \left[\sum_{(u,m) in\ \mathcal{D}} \left(\hat{r}_m^u - \sum_{i=1}^{K} U_{u,i}\, M_{i,m} \right) \right] \tag{9}$$

Funk in [5] proposed an incremental procedure, based on gradient descent, to minimize the error of the model on observed ratings. User and movie feature values are randomly initialized and updated as follows:

$$U'_{u,i} = U_{u,i} + \eta\,(2e_{u,m} \cdot M_{i,m}) \tag{10}$$
$$M'_{i,m} = M_{i,m} + \eta\,(2e_{u,m} \cdot U_{u,i}) \tag{11}$$

where $e_{u,m} = \hat{r}_m^u - r_m^u$ is the prediction error on the pair (u, m) and η is the learning rate. The initial model could be further improved considering regularization coefficients λ. Updating rules become:

$$U'_{u,i} = U_{u,i} + \eta\,(2e_{um} \cdot M_{i,m} - \lambda \cdot U_{u,i}) \tag{12}$$
$$M'_{i,m} = M_{i,m} + \eta\,(2e_{um} \cdot U_{u,i} - \lambda \cdot M_{i,m}) \tag{13}$$

An extension of this model could be obtained considering user and movie bias vectors, which define a parameter for each user and movie:

$$r_m^u = c_u + d_m + \sum_{i=1}^{K} U_{u,i}\, M_{i,m} \tag{14}$$

Where c is the user bias vector and d is the movie bias vector. An interesting version of the SVD model was proposed in [13]. According to this formulation, known as *Asymmetric SVD*, each user is modeled through her the rated items:

$$U_{u,i} = \frac{1}{\sqrt{|M(u)| + 1}} \sum_{m \in\ M(u)} w_{i,m} \tag{15}$$

Where $M(u)$ is the set of all the movies rated by the user u. A slight different version, called *SVD++*, proposed in [9], models each user by using both a user-features vector and the corresponding implicit feedback component (movies rated by each user in the training set and the ones for whom is asked the prediction in the test-set).
Latent factor models based on the SVD decomposition change according to the number of considered features and the structure of model, characterized by presence of bias and baseline contributes. The optimization procedure used in the learning phase plays an important role: learning could be incremental (one feature at the time) or in batch (all features are updated during the same iteration of data). Incremental learning usually achieves better performances at the cost of learning time. Several version of SVD models have been tested, considering the batch learning with learning rate 0.001. Feature values have been initialized with the value $\sqrt{\frac{\mu}{K}} + rand(-0.005, 0.005)$ where μ is the overall rating average and K is the number of the considered features. The regularization coefficient, where needed, has been set to 0.02. To avoid overfitting, the training set has been partitioned into two different parts: the first one is used as actual training set, while the second one, called validation set, is used to evaluate the model. The learning procedure is stopped as soon the error on the validation set increases. Performance of the different SVD models are summarized in Tab.1, while Fig.3 shows the accuracy of the main SVD approaches. An interesting property of the analyzed models is that they reach convergence after almost the same number of iteration, no matter how many features are considered. Better performances are achieved if the model includes bias or baseline components; the regularization factors decrease the overall learning rate but are characterized by an high accuracy. In the worst case, the learning time for the regularized versions is about 60 min. The *SVD++* model with 20 features obtains the best performance with a relative improvement on the Cinematch score of about 5%.

Model	Best RMSE	Avg #Iter.
SVD	0.9441	43
SVD with biases	0.9236	45
SVD with baseline	0.9237	45
Reg. SVD	0.9388	32
Reg. SVD with biases	0.9053	186
Reg. SVD with baseline	0.9062	190
SVD++	0.9039	8

Table 1: Performance of SVD Models

3.4 Probabilistic Approaches

Several probabilistic methods have been proposed for the CF, they try to estimate the relations between users or products through probabilistic clustering techniques. The *Aspect Model* [8, 7], also called *pLSA*, is the main probabilistic model used in the CF, and belongs to the class of *Multinomial Mixture Models*. Such models assume that data were independently generated, and introduce a *latent vari-*

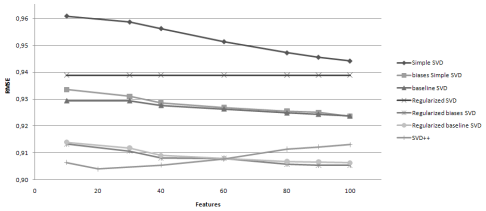


Figure 3: SVD Models Performance

able (also called hidden), namely Z, that can take K values. Fixed a value of Z, u and m are conditionally independent. The hidden variable is able to detect the hidden structure within data in terms of user communities, assuming that Z, associated to observation $(u, m, \hat{r}_m^u)$, models the reason why the user u voted for the movie m with rating $\hat{r}_m^u$. Formally, assuming the user community version, the posterior probability of $\hat{r}_m^u = v$ is:

$$P(\hat{r}_m^u = v|u, m) = \sum_{z=1}^{K} P(\hat{r}_m^u = v|m, z)P(Z = z|u) \quad (16)$$

Where $P(Z = z|u)$ represents the participation in a pattern of interest by u, and $P(\hat{r}_m^u = v|m, z)$ is the probability that a user belonging to pattern z gives rating v on the movie m. A simplified version of the Aspect Model is the *Multinomial Mixture Model* that assumes there is only one type of user [11]:

$$P(\hat{r}_m^u = v|u, m) = \sum_{z=1}^{K} P(\hat{r}_m^u = v|m, z)P(Z = z) \quad (17)$$

The standard learning procedure, for the Multinomial Mixture Model, is the Expectation Maximization algorithm [12]. Fig. 4 shows the RMSE achieved by the Multinomial Mixture Model with different number of latent class. The model

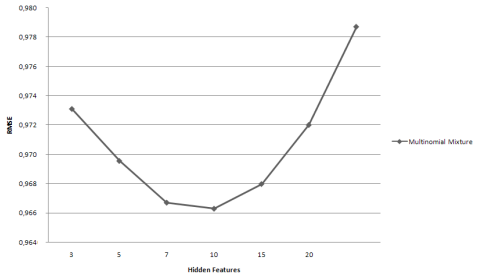


Figure 4: RMSE - Multinomial Mixture

has been initialized randomly and the learning phase required about 40 iterations of the training set but since the first 10 iterations the model reaches the 90% of its potentiality. The best result (0.9662) is obtained considering 10 latent settings for Z. The pLSA model was tested assuming a *Gaussian distribution* for the rating probability given the state of the hidden variable and the considered movie m, in the user-community version. The model was tested for different values of user-communities, as in Fig. 5. To avoid overfitting was implemented the early stopping strategy, described in the previous section. The best pLSA model produces an improvement of around 1% on Cinematch. The

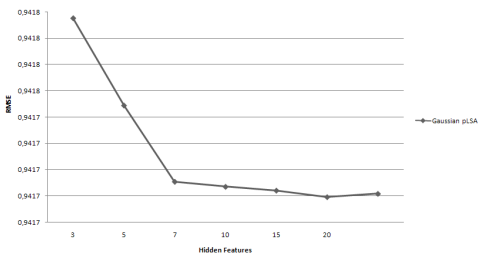


Figure 5: RMSE - pLSA

drawback of the model is the process of learning: a few iterations (3 to 5) of the data are sufficient to overfit the model.

4. MODEL COMPARISON

In this section it is performed a comparative analysis of the above described models. Each model is tuned with it best parameters settings. As said before *Cinematch*, the Netflix's Recommender System, achieves an RMSE equals to 0.9525. Figure 6 shows the RMSE of all Baseline models mentioned. The best model is the *doubleCentering*, but

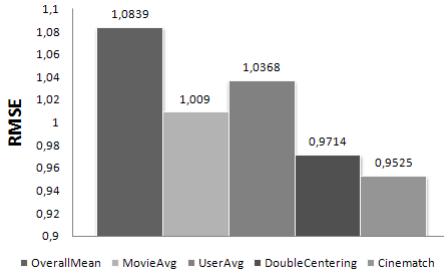


Figure 6: Baseline models

no one of them outcomes the accuracy of Cinematch. Figure 7 shows the mentioned K-NN models performances. Performances are really better than baseline ones. Except

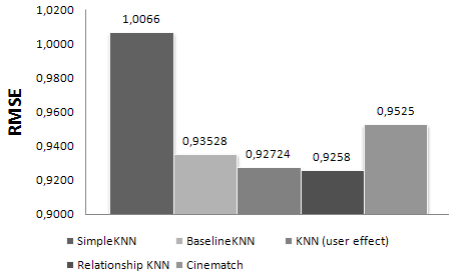


Figure 7: K-NN models

the *SimpleK-NN*, all approaches improve Cinematch's precision, especially the *Neighborhood Relationship Model*. Quality of SVD models is shown in figure 8. SVD models show the best performances, note *SVD++*. Figure 9 shows the behavior of the two proposed probabilistic models. Only *pLSA* outcomes Cinematch. Finally, figure 10 compare the best models for each algorithm family. In this experimen-

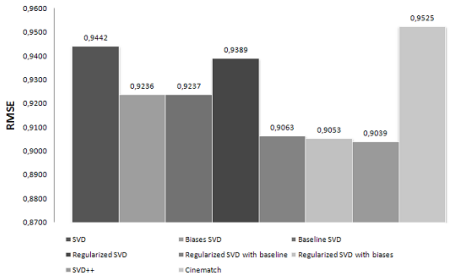


Figure 8: SVD models

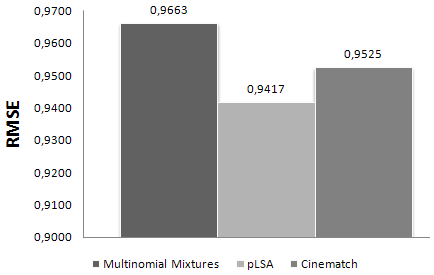


Figure 9: Probabilistic models

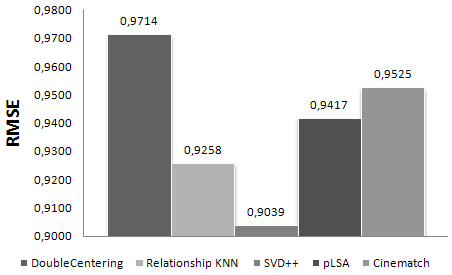


Figure 10: Best models

tation *SVD++* results to be the best model among all proposed ones.

5. CONCLUSIONS AND FUTURE WORK

This work has presented an empirical comparison of some of the most effective individual CF approaches applied to the Netflix dataset, with their best settings. Best performances are achieved by the *Neighborhood Relationship* and the *SVD++* models. Moreover, the symbiosis of standard approaches with simple baseline or biases models improved the performances, obtaining a considerable gain with respect to Cinematch. From a theoretical point of view, probabilistic models should be the most promising, since the underlying generative process should in principle summarize the benefits of latent modeling and neighborhood influence. However, these approaches seem to suffer from overfitting issues: experiments showed that their RMSE value is not comparable to the one achieved by SVD or K-NN models. Future works will focus on the study of the *Latent Dirichlet Allocation (LDA)* [4] that extends the *pLSA* model reducing the risk of over fitting, and on the integration of baseline/bias contributes in probabilistic approaches.

6. REFERENCES

[1] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM.

[2] R. M. Bell and Y. Koren. Improved neighborhood-based collaborative filtering. In *In Proc. of KDD-Cup and Workshop at the 13th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining*, pages 7–14, 2007.

[3] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 43–52, Washington, DC, USA, 2007. IEEE Computer Society.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[5] S. Funk. Netflix update: Try this at home. URL: http://sifter.org/ simon/Journal/20061211.html.

[6] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:61–70, 1992.

[7] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266, New York, NY, USA, 2003. ACM.

[8] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, January 2004.

[9] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[10] H. Lieberman. Letizia: An Agent that Assists Web Browsing. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, pages 924 – 929, 1995.

[11] B. Marlin. Modeling user rating profiles for collaborative filtering. In *In NIPS*17*, 2003.

[12] T. K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60, 1996.

[13] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*, pages 39–42, 2007.

[14] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2007.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

User Evaluation of Multidimensional Relevance Assessment

Célia da Costa Pereira
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013
Crema (CR), Italy
pereira@dti.unimi.it

Mauro Dragoni
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013
Crema (CR), Italy
dragoni@dti.unimi.it

Gabriella Pasi
Università degli Studi di Milano Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione
Viale Sarca, 336, I-20126
Milano (MI), Italy
pasi@disco.unimib.it

ABSTRACT

In this paper a user evaluation is proposed to assess the effectiveness of systems based on multidimensional relevance assessment. First of all, we introduce our approach to multidimensional modeling and aggregation, and the criteria used for the experiments. Then, we describe how the user evaluation has been performed, and finally, we discuss the results obtained.

1. INTRODUCTION

In the first traditional approaches to Information Retrieval (IR), relevance was modeled as "topicality", and its numeric assessment was based on the matching function related to the adopted IR model (*boolean model*, *vector space model*, *probabilistic model* or *fuzzy model*). However, relevance is, in its very nature, the result of several components or dimensions. Cooper [2] can be considered as one of the first researchers who had intuitions on the multidimensional nature of the concept of relevance. He defined relevance as *topical relevance with utility*. Mizzaro, who has written an interesting article on the history of relevance [8], proposed a relevance model in which relevance is represented as a four-dimensional relationship between an information resource (surrogate, document, and information) and a representation of the user's problem (query, request, real information need and perceived information need). A further judgment is made according to the: topic, task, or context, at a particular point in time. The dimensions pointed out by Mizzaro are in line with the five manifestations of relevance suggested by Saracevic [10]: *system or algorithmic relevance*, *topical or subject relevance*, *cognitive relevance or pertinence*, *situational relevance or utility* and *motivational or effective relevance*. However, the concept of *dimension* used in this paper which is similar to that used by Xu and Chen in [12] is somehow different from that used by Mizzaro and Saracevic. They defined several kinds of relevance and call them *dimensions of relevance* while we define relevance as a *concept of concepts*, i.e., as a point in a n-dimensional space composed by n criteria. The document score is then the result of a particular combination of those n space components as explained in [3, 4].

One of the problems raised by considering relevance as a multidimensional property of documents is how to aggregate the related relevance scores. In [3, 4] an approach for prioritized aggregation of multidimensional relevance has been proposed. The proposed aggregation scheme is user dependent: a user can be differently interested in each dimension. The computation of the overall relevance score to be associated with each retrieved document is then based on the aggregation of the scores representing the satisfaction of the considered dimensions. A problem raised by this new approach is how to evaluate its effectiveness. In fact, there is no test collection suited to evaluate such a model. In this paper, we first recall the models for aggregating multiple dimensions evaluations for relevance assessment presented in [3] and [4]. We focus on observing how document rankings are modified after applying the two operators on the different typologies of users (different dimensions orderings).

The paper is organized as follows. Section 2 recalls the aggregation models used in the paper. Section 3 presents the performed user evaluation and, finally, Section 4 concludes the paper.

2. PRIORITIZED MULTICRITERIA AGGREGATION

In this section, after a brief background on the representation of a multicriteria decision making problem, two prioritized approaches for aggregating distinct relevance assessments are shortly presented.

2.1 Problem Representation

The presented multicriteria decision making approaches have the following components:

- the set C of the n considered criteria: $C = \{C_1, \ldots, C_n\}$, with C_i being the function evaluating the ith criterion;
- the collection of documents D;
- an aggregation function F to calculate for each document $d \in D$ a score $F(C(d))$[1] $= RSV(d)$ on the basis of the evaluation scores of the considered criteria.

[1]Actually, it corresponds to $F(C_1(d), \ldots, C_n(d))$.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

$C_j(d)$ represents the satisfaction scores of document d with respect to criterion j. The weight associated with each criterion $C_i \in C$, with $i \neq 1$, is document and user-dependent. It depends on the preference order of C_i for the user, and also on both the weight associated to criterion C_{i-1}, and the satisfaction degree of the document with respect to C_{i-1}[2]. Formally, if we consider document d, each criterion C_i has an importance $\lambda_i \in [0, 1]$.

Notice that different users can have a different preference order over the criteria and, therefore, it is possible to obtain different importance weights for the same document for different users.

We suppose that $C_i \succ C_j$ if $i < j$. This is just a representational convention which means that the most preferred criteria have lower indexes.

We suppose that:

- for each document d, the weight of the most important criterion C_1 is set to 1, i.e., by definition we have: $\forall\, d\ \lambda_1 = 1$;
- the weights of the other criteria C_i, $i \in [2, n]$, are calculated as follows:

$$\lambda_i = \lambda_{i-1} \cdot C_{i-1}(d), \tag{1}$$

where $C_{i-1}(d)$ is the degree of satisfaction of criterion C_{i-1} by document d, and λ_{i-1} is the importance weight of criterion C_{i-1}.

2.2 The Prioritized Scoring model

This operator allows us to calculate the overall score value from several criteria, where the weight of each criterion depends both on the weights and on the satisfaction degrees of the most important criteria — the higher the satisfaction degree of a more important criterion, the more the satisfaction degree of a less important criterion influences the overall score.

Operator F_s is defined as follow: $F_s : [0, 1]^n \rightarrow [0, n]$ and it is such that, for any document d,

$$F_s(C_1(d), \ldots, C_n(d)) = \sum_{i=1}^{n} \lambda_i \cdot C_i(d). \tag{2}$$

The RSV_s of the alternative d is then given by:

$$RSV_s(d) = F_s(C_1(d), \ldots, C_n(d)). \tag{3}$$

Formalizations and properties of this operator are presented in [3].

2.3 The Prioritized "min" Operator

In this section a prioritized "min" (or "and") operator is recalled [4]. This operator allows to compute the overall satisfaction degree for a user whose overall satisfaction degree is strongly dependent on the degree of the least satisfied criterion. The peculiarity of such an operator, which also distinguishes it from the traditional "min" operator, is that the extent to which the least satisfied criterion is considered depends on its importance for the user. If it is not important at all, its satisfaction degree should not be considered, while if it is the most important criterion for the user, only its satisfaction degree is considered. This way, if we consider a document d, for which the least satisfied criterion C_k is also the least important one, the overall satisfaction degree will be greater than $C_k(d)$; it will not be C_k as it would be the case with the traditional "min" operator — the less important is the criterion, the lower its chances to represent the overall satisfaction degree.

The aggregation operator F_m is defined as follows. $F_m : [0, 1]^n \rightarrow [0, 1]$ is such that, for all document d,

$$F_m(C_1(d), \ldots, C_n(d)) = \min_{i=1,n} (\{C_i(d)\}^{\lambda_i}). \tag{4}$$

Formalizations and properties of this operator are presented in [4].

[2]If there are more than one criterion with the same priority order, the average weight and the average satisfaction degree are considered.

3. USER EVALUATION OF THE PRIORITIZED AGGREGATION OPERATORS

In [3, 4] the proposed approach for prioritized aggregation of the considered relevance dimensions has been applied to personalized IR without loss of generality. The considered personalized approach relies on four relevance dimensions: aboutness, coverage, appropriateness, and reliability. The aboutness is computed as the similarity between the document vector and the query vector. The scores of the coverage and the appropriateness criteria are computed based on a similarity of the document vector and a vector of terms representing the user profile. While the reliability represents the trust degree for a user of the source from which document comes.

3.1 Preliminary Assumptions

The prioritized aggregations approach is based on the user's indication (either explicit or implicit) of the importance order of relevance dimensions. In [3, 4] different user's behaviors have been described. In the case in which a user formulates a query with the idea of locating documents which are about the query and which also cover all his interests, and at the same time he does not care about the fact that the document also focuses on additional topics the user can be called "coverage seeker". If on the contrary the user's intent is to privilege documents which perfectly fit his interests the user is called "appropriateness seeker"

On the contrary, a user who formulates a query which has no intersection with his interests or users who do not have a defined list of interests – *interest neutral* – will not give any importance to the coverage and appropriateness criteria. Users of this kind are just looking for a satisfactory answer to their current concern, as expressed by their query. Finally, users who are cautious about the trustworthiness of the origin of the retrieved documents – *cautious* – will give more importance to the reliability criterion than to the others.

For example, *coverage seeker* users can be defined as follows:

$CARA_p$: $coverage \succ aboutness \succ reliability \succ appropriateness$;

3.2 Experiments

In this section, the impact of the proposed prioritized aggregation operators in the personalized IR setting is evaluated. In Section 3.2.1 we present the settings used to perform the experiments, while in Section 3.2.2 we discuss the obtained results.

3.2.1 Experimental Settings

The traditional way to evaluate an information retrieval system is based on a test collection composed by a document collection, a set of queries, and a set of relevance judgments which classify a document as being relevant or not for each query. Precision and recall are then computed to evaluate the effectiveness of the system. Unfortunately, there is not a test collection suited to evaluate a system based on approaches like the one proposed in this paper. It is important to notice that in the case of a user-independent aggregation of the multiple relevance numeric assessments, a traditional system's evaluation could be applied. In fact if for example the single assessment scores are aggregated by a mean operator, the system could produce the same result for a same query and a same document, independently of the user judgments. When applying the prioritized aggregation that we have proposed, a same document evaluated with respect to a same query, could produce distinct assessment scores depending on the adopted prioritized scheme, which is user-dependent.

The evaluation approach proposed in this paper is based on an analysis of how document rankings are modified accordingly to the prioritized aggregations associated with the user's typologies that we have identified in Section 3.1.

The relevance criteria and their aggregation discussed in the previous sections have been implemented on top of the well-known Apache Lucene open-source API [3]. The Reuters RCV1 Collection (over 800,000 documents) has been used. The method that we have used to generate both queries and user's profiles has been inspired by the approach presented by Sanderson in [9]. In this work the author presents a method to perform simple IR evaluations by using the Reuters collection that does not have queries nor relevance judgments, but has one or more subject codes associated with each document.

He splits the collection in two parts, a query set **"Q"** and a test set **"T"**, and documents are randomly assigned to one of the two subsets. Then, all subject codes are grouped in a set **"S"**. For each subject code s_x, all documents tagged with the subject code s_x are extracted from the set **"Q"**. From these documents, the pairs (word, weight) are generated to create a query. Then, the query is performed on the set **"T"**. The precision/recall curves are calculated by considering as relevant, the documents that contain the subject code s_x.

We have been inspired by Sanderson's approach to build both the queries and the user's profiles. The queries have been created as expressed above. The creation of the user's profile has been done in the following way. The set **"Q"** has been split in different subsets based on the subject code of each document (ex. "sport", "science", "economy", etc.). Each subset of **"Q"** represents the set of documents known by the users interested in that particular topic. For example, the subset that contains all documents tagged with the subject code "sport" represents the set of documents known by the users interested in sports.

We have indexed each subset of **"Q"** and, for each created index, we have calculated the TF-IDF of each term. Then, we have computed a normalized ranking of these terms and we have extracted the most significant ones. The TF-IDF of each term represents the interest degree of that term in the profile, that is, how much the term plays the role of a good representation of the user's interests.

An example of user's profile is illustrated in Table 1. For example, the users associated with the "BIOTECH" profile have, with respect to the term "disease", an interest degree of 0.419. Each profile is viewed as a long term information need, therefore, it is treated in the same way as documents or queries.

To study the behavior of the system, we have carried out a user evaluation as proposed in [1] [5] [6].

The user evaluation described in this paper has been inspired by the one suggested in [7] that simply consists in a procedure in which a set of at least 6 users performs a set of at least 6 queries.

In these experiments we have considered eight users with eight different profiles, each one associated with a subset of **"Q"** (Table 2).

BIOTECH					
scientist	1.000	gene	0.402	patient	0.260
researcher	0.563	study	0.386	brain	0.259
disease	0.419	clone	0.281	people	0.254
cancer	0.410	animal	0.279	experiment	0.249
human	0.406	planet	0.267	drug	0.247

Table 1: The top 15 interest terms of the BIOTECH profile.

The aims of these experiments are to verify that: (i) when a user performs queries in-line with his interests, by applying a prioritized aggregation operator, the system produces an improved ranking with respect to the one produced by simply averaging the scores, and (ii) when a user performs queries that are not-in-line with his interests, by applying a prioritized aggregation operator, the quality of the produced rank does not decrease with respect to the situation in which the prioritized aggregation operators are not applied.

Two kinds of queries have been considered. Those which are in-line with the interests contained in the user's profile, Q_i, and those which are not-in-line with the interests contained in the user's profile, Q_n. Table 2 illustrates the set Q_i and shows the associations between the user's profiles and the performed queries. In these preliminary experiments only one query has been generated for each user. For instance, for User 1, the set Q_i is composed only by the query Q1, while the set Q_n is composed by all the other queries from Q2 to Q8.

For User 2, the set Q_i is composed only by the query Q2, while the set Q_n is composed by the query Q1 and the queries from Q3 to Q8, and so on for the other users.

User	Profile Name	Query
User1	SPACE	Q1: "space shuttle missions"
User2	BIOTECH	Q2: "drug disease"
User3	HITECH	Q3: "information technology"
User4	CRIMINOLOGY	Q4: "police arrest sentence fraud"
User5	DEFENSE	Q5: "russia military navy troops"
User6	DISASTER	Q6: "flood earthquake hurricane"
User7	FASHION	Q7: "collection italian versace"
User8	SPORT	Q8: "premiership league season score"

Table 2: The queries executed for each user profile.

When a user submits a query, the matching between the query vector and each document vector is made first (aboutness), then, on each document the coverage and the appropriateness criteria are evaluated by comparing the document vector with the user's profile vector. Finally, the value of the reliability criterion, which corresponds to the degree to

[3]See URL `http://lucene.apache.org/`.

which the user trusts the source from which the document comes, is taken into account. These are the values to be aggregated — aboutness, coverage, appropriateness and reliability.

The evaluation of the produced rank is made by the eight real users that used the system. Each user analyzed the top 10 documents returned by the system and assessed, for each document, if it is relevant or not.

3.2.2 Discussion of the Results

In this section we present the obtained results. For space reasons some ranks have not been inserted, however the complete archive of the ranks produced in these experiments are available online [4]. For convenience, only the top 10 ranked documents are reported in each table. The rationale behind this decision is the fact that the majority of search result click activity (89.8%) happens on the first page of search results [11], that is, generally, users only consider the first 10 (20) documents. The baseline rank for the "Scoring" operator is obtained by applying the average operator to calculate document assessment. Such rank corresponds to the average assessment of the documents considering the four criteria and without considering priorities among the criteria. Instead, the baseline rank for the "Min" operator is obtained by applying the *standard min* operator. Table 3 illustrates an example of rank produced by the average operator after performing a query in Q_i, while Table 4 illustrates an example of rank produced by the standard min operator after performing a query in Q_i. The entries marked with the asterisk before the title, have been considered relevant with respect to both the performed query and the user profile. We can notice that there are more non-relevant documents in the top 10 list resulting from the application of average operator than in the list resulting from the application of the standard min operator. This is due to the compensatory nature of the average operator.

We illustrate the behavior of the system by taking into account different kinds of aggregations applied to the User 1, the user associated to the "SPACE" profile. In particular, we present in Tables from 5 to 10 the results obtained by applying both the Prioritized "Scoring" Operator and the Prioritized "Min" Operator, with the aggregations ACA_pR, CA_pAR, and A_pCAR

We can notice that the proposed document rankings are improved, with respect to the baselines ranking for both operators and for the considered aggregations, in the sense that the number of relevant documents in the top 10 is greater than the number of relevant documents in the baseline ranking — non relevant documents are put down in the ranking.

We can also notice that, while the document in the 9th position of the top 10 documents in Table 3 is deemed sufficiently topical for the user with profile "SPACE", the same document is not even considered in the top 10 list of any table corresponding to the prioritized "Scoring" operator. This is due to the fact that, even though the document satisfies the query because it contains information about space mission, its content is instead related to space exploration. Instead, for example, the document in the first position in the scoring baseline rank, is also proposed in almost all the top ten documents (scoring and min) including the min baseline rank. An exception is Table 6 where that document does not appear. The reason is that this document comes from a source with a very low degree of reliability.

Different considerations have to be done when the user's query is not in-line with his profile (i.e. the user's query is in the set Q_n). We will discuss about two different scenarios. In the first one the user associated with the "BIOTECH" profile executes the query associated to the "FASHION" profile, while in the second scenario, the user associated to the "CRIMINOLOGY" profile executes the query associated to the "SPACE" profile. We have noticed that, for the scoring operator, the results for all aggregations are in general similar to the baseline. The previous considerations are not valid for the prioritized min operator. It is due to its definition. Indeed, if just one criterion is weak satisfied, the overall assessment is very low. Now, if users make queries not in line with their profile, the criteria like coverage and appropriateness are weakly satisfied and then the overall value is low. Instead, when considering the prioritized min operator, the result depends also on the importance degree of the least satisfied criterion. We can conclude that the (prioritized) min operator should not be used for the users who make queries that are not in line with their profile.

[4]http://www.dti.unimi.it/dragoni/files/ MultirelevanceUserEvaluation.rar

4. CONCLUSION AND FUTURE WORK

In this paper, a user evaluation for aggregating multiple criteria has been presented and discussed.

The experimental results have been obtained thanks to a case study on personalized Information Retrieval with multi-criteria relevance. These results show that: (i) the proposed operators allow to improve the ranking of the documents which are related to the user interest, when the user formulates an interest-related query; (ii) for the "scoring" operator, when a user has no interests or formulates a query which is not related to his interests, the ranking of the documents is similar to the ranking obtaining by using the average operator; and (iii) for the "min" operator, when the user formulates a non interest-related query this operator is not suitable.

R.	Document Title	Score
1	***Shuttle Atlantis blasts off on schedule.**	0.626
2	Countdown starts for Sunday shuttle launch.	0.575
3	***Shuttle finally takes Lucid off space station Mir.**	0.573
4	U.S. spacewoman breaks another record.	0.573
5	***Shuttle Discovery heads for Florida.**	0.572
6	***Shuttle Atlantis heads for Mir despite problem.**	0.568
7	Scientists delighted with U.S. shuttle flight.	0.567
8	***U.S. shuttle launched on mission to Mir.**	0.563
9	Boeing-Lockheed group signs $7 billion shuttle pact.	0.562
10	***U.S. shuttle leaves space station Mir.**	0.561

Table 3: Results for "SPACE" profile by applying the average operator.

R.	Document Title	Score
1	***Part of planned space station arrives in Florida.**	0.250
2	***French astronaut to join Russian space mission.**	0.242
3	***Russia, hurt by Mars failure, sends probe to space.**	0.231
4	***Astronauts board shuttle for U.S. launch.**	0.228
5	***Shuttle Columbia blasts off to mission.**	0.228
6	***Shuttle Atlantis blasts off on schedule.**	0.225
7	***Shuttle Discovery lands in Florida.**	0.216
8	***U.S. space shuttle crew set for Thursday landing.**	0.215
9	***U.S. shuttle leaves space station Mir.**	0.210
10	RUSSIA: Frenchman's August Mir flight scrapped.	0.202

Table 4: Results for "SPACE" profile by applying the standard min operator.

5. REFERENCES

R.	Document Title	Score	Gap
1	***Shuttle Discovery takes off on schedule.**	1.521	25
2	***Shuttle Atlantis blasts off on schedule.**	1.427	-1
3	***U.S. space shuttle heads home.**	1.381	85
4	***Shuttle Discovery heads for Florida.**	1.333	1
5	***U.S. shuttle crew set up space laboratory.**	1.323	35
6	***Columbia shuttle mission extended one day.**	1.317	35
7	***Shuttle Atlantis heads for Mir despite problem.**	1.313	-1
8	***Shuttle Discovery lands in Florida.**	1.275	3
9	***U.S. space shuttle crew set for Thursday landing.**	1.264	62
10	***U.S. shuttle will not flush Mir's water.**	1.253	32

Table 5: Results for "SPACE" profile by applying the Prioritized Scoring Operator and ACA_pR aggregation.

R.	Document Title	Score	Gap
1	***Shuttle Atlantis to return home on Wednesday.**	0.661	53
2	***With spacewalk off, shuttle astronauts relax.**	0.652	30
3	***U.S. space shuttle heads for rendezvous with Mir.**	0.643	39
4	***U.S. shuttle crew prepares to retrieve satellite.**	0.632	257
5	***Shuttle-deployed telescope ready for action.**	0.631	260
6	***Space shuttle deploys U.S.-German satellite.**	0.628	217
7	***Shuttle crew prepares for nighttime landing.**	0.628	264
8	***Hubble service crew prepares to return home.**	0.625	150
9	***Satellites line up behind shuttle Columbia.**	0.621	129
10	RUSSIA: Sticken Mir crew stands down, says worst over.	0.620	256

Table 6: Results for "SPACE" profile by applying the Prioritized Min Operator and ACA_pR aggregation.

[1] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):152, 2003.

[2] W. S. Cooper. On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, 24(2):87–100, 1973.

[3] C. da Costa Pereira, M. Dragoni, and G. Pasi. Multidimensional relevance: A new aggregation criterion. In *ECIR'09*, pages 264–275, 2009.

[4] C. da Costa Pereira, M. Dragoni, and G. Pasi. A prioritized "and" aggregation operator for multidimensional relevance assessment. In *AI*IA 2009, to appear*, 2009.

[5] P. Ingwersen. *Information Retrieval Interaction.* Taylor Graham, 1992.

[6] P. Ingwersen. Cognitive perspectives of information retrieval interaction: elements of a cognitive ir theory. *Journal of Documentation*, 52(1):3–50, 1996.

[7] P. Ingwersen and K. Järvelin. *The Turn Integration of Information Seeking and Retrieval in Context Series.* Springer, 2005.

[8] S. Mizzaro. Relevance: the whole history. *J. Am. Soc. Inf. Sci.*, 48(9):810–832, 1997.

[9] M. Sanderson. The reuters collection. In *Proceedings of the 16th BCS IRSG Colloquium*, 1994.

[10] T. Saracevic. The stratified model of information retrieval interaction: Extension and applications. *Journal of American Society for Information Science*, 34:313–327, 1997.

[11] A. Spink, B. Jansen, C. Blakely, and S. Koshman. A study of results overlap and uniqueness among major web search engines. *Inf. Process. Manage.*, 42(5):1379–1391, 2006.

[12] Y. C. Xu and Z. Chen. Relevance judgment: What do information users consider beyond topicality? *J. Am. Soc. Inf. Sci. Technol.*, 57(7):961–973, 2006.

R.	Document Title	Score	Gap
1	***Russians aim to fix Mir before US Shuttle arrives.**	0.777	52
2	***Russians hope to fix Mir before Shuttle arrives.**	0.742	68
3	***With spacewalk off, shuttle astronauts relax.**	0.707	53
4	Countdown continues for U.S. spacewoman's return.	0.700	70
5	***Shuttle Columbia blasts off to mission.**	0.700	137
6	***Shuttle Atlantis blasts off on schedule.**	0.682	-5
7	***Navigational problem crops up on shuttle mission.**	0.681	40
8	***U.S. shuttle launched on mission to Mir.**	0.679	0
9	Sticken Mir crew stands down, says worst over.	0.676	78
10	***Astronaut Lucid tones up for ride home.**	0.673	96

Table 7: Results for "SPACE" profile by applying the Prioritized Scoring Operator and CA_pAR aggregation.

R.	Document Title	Score	Gap
1	***Shuttle Atlantis blasts off on schedule.**	0.466	5
2	***U.S. shuttle leaves space station Mir.**	0.460	7
3	***Astronauts board shuttle for U.S. launch.**	0.459	1
4	***Shuttle Atlantis moved to pad for Mir mission.**	0.453	27
5	Russians, Ukrainian set for 1997 shuttle flights.	0.452	12
6	***Shuttle finally takes Lucid off space station Mir.**	0.450	41
7	***Shuttle Discovery takes off on schedule.**	0.447	15
8	Astronauts arrive for U.S. shuttle launch.	0.446	12
9	***U.S. shuttle launch further delayed.**	0.446	66
10	***Shuttle Columbia blasts off to mission.**	0.446	-5

Table 8: Results for "SPACE" profile by applying the Prioritized Min Operator and CA_pAR aggregation.

R.	Document Title	Score	Gap
1	***Shuttle Columbia blasts off to mission.**	0.364	141
2	***Shuttle Atlantis blasts off on schedule.**	0.364	-1
3	***Part of planned space station arrives in Florida.**	0.362	69
4	***Astronauts board shuttle for U.S. launch.**	0.351	48
5	***French astronaut to join Russian space mission.**	0.336	89
6	***Russia, hurt by Mars failure, sends probe to space.**	0.332	208
7	***U.S. shuttle leaves space station Mir.**	0.332	3
8	***U.S. space shuttle crew set for Thursday landing.**	0.314	63
9	Russians, Ukrainian set for 1997 shuttle flights.	0.303	117
10	***U.S. shuttle launched on mission to Mir.**	0.299	-2

Table 9: Results for "SPACE" profile by applying the Prioritized Scoring Operator and A_pCAR aggregation.

R.	Document Title	Score	Gap
1	***Part of planned space station arrives in Florida.**	0.250	0
2	***French astronaut to join Russian space mission.**	0.242	0
3	***Russia, hurt by Mars failure, sends probe to space.**	0.231	0
4	***Astronauts board shuttle for U.S. launch.**	0.228	0
5	***Shuttle Columbia blasts off to mission.**	0.228	0
6	***Shuttle Atlantis blasts off on schedule.**	0.225	0
7	***Shuttle Discovery lands in Florida.**	0.216	0
8	***U.S. space shuttle crew set for Thursday landing.**	0.215	0
9	***U.S. shuttle leaves space station Mir.**	0.210	0
10	Lack of funds threaten Russia's space programme.	0.204	258

Table 10: Results for "SPACE" profile by applying the Prioritized Min Operator and A_pCAR aggregation.

From Entities to Geometry: Towards exploiting Multiple Sources to Predict Relevance

Emanuele Di Buccio
Department of Information Engineering
University of Padua, Italy
dibuccio@dei.unipd.it

Mounia Lalmas
Department of Computing Science
University of Glasgow, UK
mounia@acm.org

Massimo Melucci
Department of Information Engineering
University of Padua, Italy
melo@dei.unipd.it

ABSTRACT

The goal of an Information Retrieval (IR) system is to predict which information objects can help users in satisfying their information needs, i.e. predict relevance. Different sources of evidence can be exploited for this purpose. These sources are the properties of the different entities involved when retrieving and accessing information, where examples of entities include the information objects, the task, the user, or the location. The main hypothesis of this paper is that, to exploit the variety of entities and sources, it is necessary to model the relationships existing between the entities and those existing between the properties of the entities. Such relationships are themselves possible sources that can be used to predict relevance. This paper proposes a methodology that supports the design of an IR system able to model in a uniform way the properties of the entities involved, the properties of their relationships and the relationships between the different properties. The methodology is structured in four steps, aiming, respectively, at supporting the selection of the sources, collecting the evidence, modeling the sources and their relationships, and using the latter two to predict relevance. Sources and relationships are modeled and then exploited through a previously proposed geometric framework, which provides a uniform and concrete representation in terms of vector subspaces.

1. INTRODUCTION

The goal of an IR system is to predict which information objects can help users in satisfying their information needs. For instance, if the information need is expressed by the user as a textual query, the IR system has to predict which documents are relevant to the formulated query. According to this interpretation, IR can be framed as a problem of evidence and prediction [1]. The prediction can be performed through the different sources of *evidence* involved in the retrieval process. Content, meta-data and annotations of the information objects are examples of such sources, and have been used by many retrieval systems.

These sources have been shown to be effective to predict relevance, but other sources exist. An example is the behavior of the user during the search process, for instance described in terms of interaction features – display time, click-through data, amount of scrolling, or other features e.g. [2]. These features have been adopted as sources of evidence to estimate relevance, e.g. display-time in [3], click-through data in [4], or a combination of several features in [5, 6]. Nowadays commercially available devices, e.g. mobile phones, are equipped with tools that can capture information about the user location and from the surrounding environment, besides having access to all the information provided by the web or the user personal data.

The various sources may not have the same impact in predicting relevance, and as such their relative contributions should be investigated. For instance ranking algorithms that are based on different object representations will usually return sets of relevant information objects with little overlap [8]. It is therefore important, as stated in [8], to "explicitly describe and combine multiple sources of evidence about relevance" when developing ranking algorithms. More precisely, it is important to explicitly consider the relationships existing between sources. However, the design and the implementation of distinct ranking algorithms, one for each type of sources, may not allow for considering relationships between sources. It is thus important to investigate approaches that combine evidences rather than approaches that combine ranking algorithms. This would allow for the relationships between sources to be explicitly integrated in the ranking algorithm.

This paper proposes a methodology that supports the design of an IR system able to model in a uniform way the properties of the entities involved, the properties of their relationships and the relationships between the different properties. The methodology is structured in four steps, aiming, respectively, at supporting the selection of the sources, collecting the evidence, modeling the sources and their relationships, and using the latter two to predict relevance. The last two steps are based on the geometric framework proposed in [9], which provides a uniform and concrete representation of the sources and their relationships in terms of vector subspaces.

The methodology aims at being general, in the sense that it is not related to a specific source or set of sources. However, for illustration purpose, two sources will be considered in this paper, namely, the content of the information objects to be ranked and the behavior of the users when accessing or retrieving information. The former has been selected because past research in IR provides a number of representations of the content that have been shown to lead to effective retrieval [8]. The latter has been extensively investigated in

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

Information Science (IS) and has in the last decade become a subject of investigation in IR. Indeed, experimental evaluation has shown how usage data stored in transaction logs [3, 4, 6, 10] or so-called interactive IR systems [11, 12] can effectively predict relevance. The use of the Entity-Relationship database model for describing IR objects was introduced in [13] for automatic hypertext construction purpose – this paper enlarges that view and connect the entities and relationship at the conceptual level to a mathematical model which provides a language at the logical level.

2. MOTIVATIONS AND METHODOLOGY RATIONALE

IR systems can exploit the evidence provided by different sources to improve retrieval effectiveness. In [8] the author considers several document representations and discusses approaches to combine the contribution provided by each representation. In [14] the inference network framework is adopted to combine link-based evidence with content-based evidence for web retrieval. Evidence on the structure of the documents can be incorporated, for instance, using the Dempster-Shafer theory of evidence [15]. However, the different document representations are only a subset of the available sources.

Let us consider, for instance, the scenario where a user is looking for information about restaurants in London. If Venice is the location where the search is performed, this probably suggests that the user is planning a trip in London, and restaurants in an arbitrary London area may be of interest. If the search is performed on a mobile phone and the GPS position indicates that the user is in London, probably the user is more interested in restaurants near his current position. We can see that in this scenario, other units besides the information objects are involved. In this paper, we refer to units as *entities*. For instance, in our scenario, the entities involved are the user, the location, the task the user is performing when looking for information – i.e. "travel in London" – and the specific topic within the task[1] – i.e. "finding restaurants in London".

Each entity is characterized by a number of properties. When the entity is an "information object", examples of properties include content, meta-data and annotation. For the entity "location", instances of properties are the GPS position or the IP address.

Each entity exists independently of the properties we can observe about it, but the observed properties are the evidence that can be used to build a model of the entity, that is to obtain a description of the entity – in this work a mathematical description – that can be used to predict relevance. In other words, *the properties of the entities are the sources of evidence that can be exploited to help predicting the relevance of information objects.*

Not only the properties of the entities are sources of evidence, but also the relationships between entities (if any) can provide additional evidence to predict relevance. Let us consider a list of results returned by an IR system in response to a query and the user who formulated the query. The behavior of the user when examining a result is one of the properties to describe the relationship between the entity user and the entity result; such property constitutes a source that can be exploited to predict relevance. Indeed, research in Interactive IR has shown that a retrieval system can benefit from evidence gathered from the information seeking activities of a user. For example, Implicit Relevance Feedback (IRF) algorithms [10] exploit the information gathered from the interactions between the user and the documents to recommend query expansion terms or to re-rank documents. Even the concept of *relevance* can be defined as "a relation between a document and a person, relative to a given information need" [1], the document and the person being two entities.

The set of entities and relationships, and their properties, are neither fixed nor unique, as they depend on the specific retrieval application – e.g. the entity location is crucial for search carried out on a mobile phone or to customize search results according to the country where the search originates. Therefore, the *selection of the sources* is an important issue that needs to be addressed.

Once the appropriate sources have been identified, each of them has to be modeled, so that to be exploited for retrieval. In this work, we refer to the model of a source as a *dimension*. A first step to obtain a dimension is to identify a set of features that describe it. *Feature* here refers to the information obtained by the observation of a property of an entity or a relationship. For an entity "location" described by the dimension "GPS position", the features are the GPS position components. For a "web result" entity, the keywords in the title, the snippet or the URL of the result are example of features. Since the features constitute the evidence that model a source, a procedure to *select and collect features* has to be designed and implemented.

The description (model) of the sources is what get used to predict relevance. In this work the framework adopted to build the description is the vector subspace formalism proposed in [9]. The basic rationale for this is that we want to map the collected data, prepared in a matrix, in a new vector space basis – the vector subspace spanned by the basis is the model of the source.

Once a representation in terms of subspaces has been built both for the sources and the information objects, a trace-based function, the one exploited in [9], can be adopted to rank information objects by exploiting the information about the different sources of evidence that have been modeled. In other words the trace-based function, which we briefly describe in Section 4, is a tool to handle the *prediction* problem.

In summary, four steps have been identified, and each of them needs to be addressed to be able to predict relevance using multiple sources of evidence, namely, sources selection, features collection, source modeling and relevance prediction. Figure 1 illustrates these four steps for the relationship between the entities "user" and "information objects"; here, the relationship is characterized by the source "user behavior" described in terms of "interaction features".

In this paper we will focus on two of the above steps, specifically evidence collection and source modelling, which will be discussed respectively in Section 3 and Section 4; some remarks on the implementation of these methodology steps and their evaluation are reported in Section 5.

[1]We take the definition of *task* and *topic* from [2]: "Task was defined for this study as the goal of information-seeking behavior, and topic was defined as the specific subject within a task."

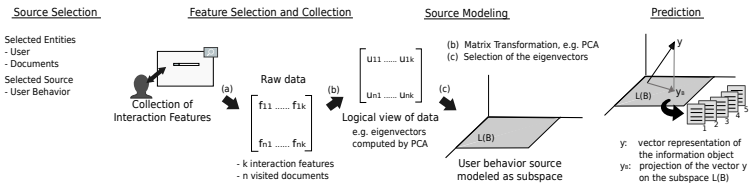


Figure 1: Methodology steps and specific application to the user interaction behavior.

3. EVIDENCE COLLECTION

Let us return to the scenario of a user looking for information about restaurants in London. Let us suppose the user, to satisfy his information need, interacts with a search engine and submits the query "restaurants in London". The search engine returns a ranked list of results. For simplicity, we focus on two entities only, namely, the user and the result. When examining the returned results, the user interacts with them and with the information objects the results refer to. In this scenario the behavior of the user when examining and (eventually) accessing the results can be considered as a property to describe the involved entities and, particularly, as a source to assist relevance prediction. In the above scenario another source available is the content of the abstracts (title, snippet and URL) of the results and the content of the corresponding information objects.

Once the sources have been selected, the next step is to *collect the evidence* to build the model of these sources. This step consists of selecting the features to be gathered to build a model of these sources, and then the actual collection of the selected features.

In the event of the source "user behavior" a possible choice, as depicted in step two of Figure 1, is the adoption of so-called interaction features. This is for instance the approach adopted in [5, 6] where several interaction features are exploited simultaneously. In particular, in [6] a subset of the features gathered in the user study described in [2] was exploited to obtain a vector subspace representation of the user behavior. When using a representation personalized for each user and tailored on the specific search task to re-rank the documents, the keywords extracted from the top re-ranked documents were shown to be effective as source for query expansion. The methodology proposed in that work assumed that the interaction features were available for all the documents to be re-ranked. But this assumption does not hold in our considered scenario, unless the documents have been already visited with regard to past queries when performing the same task. Therefore, the *availability* of the interaction features is an issue to address. A possible solution is not to consider the features with regard to a single user, but with regard to a group of users, e.g. performing the same task.

Another reason to exploit group interaction data is the *reliability* of the interaction features. The features need to be reliable indicators of the user needs, interests or intents. To clarify what we mean by "reliable feature", let us consider the display-time: this feature, when considered in isolation and referring to a single user, is subject to variations. Exploiting this feature when predicting relevance may be difficult [3], thus making it not reliable. But in [3] the authors found that display-time, when used as implicit measure, is more consistent when referring to multiple subjects performing the same task, than when personalized to each user.

Individual users and user groups, does not necessarily need to be considered as mutually exclusive sources for interaction features. For instance, in [5] user behavior models to predict user preferences for web ranking are learned by exploiting simultaneously feature values derived from the individual's behavior and those aggregated across all the users and search session for each query-URL pair.

The selection of the features of a source to then be gathered affects the modeling step, since they constitute the evidence used to build a model of the source. However, the procedure to collect features is part of the design of the IR system, in particular, the components aimed at gathering the selected features and managing them. For instance, when interaction features have been selected as implicit indicators, a browser extension can be used to monitor the gathering of such features. This is the approach adopted in the Lemur Query Log Project[2], a study to gather the query logs from users of the Lemur Query Log Toolbar[3][4]. It should be noted that the development of an extension that stores the usage data on the client side may encourage the user to adopt this monitoring tool since no personal data need to be provided to the server.

4. SOURCE MODELING AND PREDICTION

Once the evidence has been gathered, the next step consists of modeling the evidence so that it can be used to predict relevance. In this work the mathematical construct of the vector subspace is used for this purpose.

In this paper, the evidence gathered by the different sources is exploited to rank information objects with respect to a given query. This is done by using the different representations of the objects generated from the sources. For instance, if the user "interaction behavior" is a considered source, an information object can be described in terms of the interaction features monitored when a user is visiting the object — e.g. an object being displayed for 30 seconds, clicked 3 times and on which 5 scrolling actions have been performed, can be represented as the vector $y = (30, 3, 5)$. The same object, if the source "content" is considered, can be described as the vector of the TF·IDF weights of the terms appearing in it. The construct of the vector space basis is particularly suitable to model these multiple representations. Indeed, intuitively, the same information object can be represented with regard to different sources in the same way the same vector can be generated by different vector space basis.

A second reason to adopt the construct of the vector space basis is that some of the vector subspace representations

[2] `http://lemurstudy.cs.umass.edu/`
[3] `http://www.lemurproject.org/querylogtoolbar/`
[4] The goal of the study is to create a database of web search activities that will be provided to the information retrieval research community.

may reveal the logical structure underlying the collected evidence. The collected data, prepared in a matrix, is a vector representation of the source. This data often may be noisy. A matrix transformation, namely a change of basis, can be applied to map the original view of the data to one that is less noisy. Let us consider the re-evaluation of the Vector Space Model (VSM) proposed in [16]. The authors point out how some assumptions underlying the traditional VSM [17] – e.g. that the terms are orthogonal – may suggest that the vector was interpreted as a data structure and not as a logical construct. Subsequent developments show how the vector can be used as a logical construct able to capture dependencies between terms and between documents [16, 18]. The "latent semantics" [18] of the terms in the documents, that is the dependencies between terms, was used as a source for implementing a Pseudo Relevance Feedback algorithm [9] and an Explicit Relevance Feedback algorithm [19] based on the geometric framework adopted in this work.

To explain the role of the matrix transformation techniques in the modeling step, we use the example of information behavior as a source, where the latter is described in terms of interaction features. A matrix A can be prepared where the element (i, j) is feature j observed during the visit of object i, e.g. a display-time of 30 seconds. The matrix A, as mentioned above, can be a noisy vector-based representation of the observed data. A matrix transformation technique such as Principal Component Analysis (PCA) of $A^T A$ can be used to compute a new vector space basis – this is actually the approach proposed in [6]. PCA provides a set of eigenvectors and a subset of them can be used to obtain the user interaction behavior dimension – the model of the source is the subspace spanned by the eigenvectors. As suggested by this example, this geometric framework allows us to achieve one of our goals, which is to generate a representation of the properties of the relationships between entities – in the example mentioned above the user behavior was the property to be modeled.

The two mentioned approaches, that is the one adopted in [9, 19] and that adopted in [6], provide a solution for two distinct sources. In the former case the modeled source is a property of an entity, namely the latent semantics of the terms in the documents. In the latter case, the modeled source is a property of a relationships between entities, namely the user interaction behavior. However, we are also interested in modeling relationships (if any) existing between the properties of the entities, namely between sources, e.g. between the latent semantics of the terms and the user interaction behavior – this is different from modeling properties of relationships, e.g. the user interaction behavior.

Let us return to the scenario of a user looking for information about restaurants in London and suppose the term "jazz" appears in the abstract of one of the displayed results. The user when examining the result may realize that he is more interested in jazz restaurants than in general ones. This example also emphasizes how different sources are not necessarily independent from each other. Indeed, the features observed for a source (e.g. the user behavior) can be "entangled" with the features observed for another source (e.g. the particular meaning of a query feature in the selected results).

The design of one approach per source may not be able to model relationships that may occur between sources and consequently to exploit them, as reported in [20]. In this work, we consider that the relationships are themselves sources. Therefore, it is better to not consider distinct mappings, one for each source, but to compute a single vector space basis to represent the relationships between sources.

The model of the sources can be used in the retrieval process once the information objects have been represented by the features selected to describe the sources. Indeed, the measure of the degree to which the modeled source occurs in an information object can be computed as the distance between the vector representation of the information object, which corresponds to a one-dimensional subspace, and the subspace modeling the source(s) spanned by the vector space basis computed in the source modeling step. This motivates the function proposed in [9], where the author showed how such function can be interpreted as a trace-based function and that the measure is a probability measure. The idea of using trace in IR, and in particular the density operators, was originally introduced in [21], and one of its important consequence – subsequently exploited in [9] – was to "establish a link between geometry and probability in vector spaces" [21].

5. IMPLEMENTATION AND EVALUATION

The specific implementation we are investigating concern the two mentioned sources, that is, the user behavior and the latent semantic of the terms in the information objects.

With respect to user behavior, we are focusing on two issues. The first is the selection of the source for interaction features since, as discussed in Section 3, both individual and user groups interaction data can be exploited to prepare the matrix A and to build the source model. In particular, we are investigating the difference between the two contributions in terms of retrieval effectiveness when PCA is adopted as the matrix transformation technique. PCA allows handing dimensionality reduction and capturing the relationships among the features in an unsupervised manner. However, as stated in [6], the problem is that the eigenvector whose components best combine the interaction features, is not necessarily the first principal eigenvector, and the best performance are achieved when the eigenvector is manually selected. For this reason we are investigating other unsupervised methods to obtain a vector subspace representation of the interaction data.

With respect to the latent semantics of terms, one issue under investigation is the selection of the terms in the feedback documents. Indeed, if the terms appearing in these documents are adopted as evidence to build a source model, one issue, particularly when real-time feedback is required, is to handle matrices whose dimensions are the number of distinct terms in the feedback documents. In this case a possible solution is the selection of a subset of the terms, e.g. the top weighted ones. However, this strategy has been shown to not be effective [19]; therefore, we are investigating selection criteria for "good terms".

Since the main objective of the methodology is to model relationships, we will look into the *relationships* between sources, and investigate their implementation using the proposed geometric framework, and their impact on retrieval effectiveness. Two approaches are possible. The first approach is to rank information objects separately according to different dimensions and then combine the rankings into one. The second approach is to model all the sources as a unique vector subspace and then rank the information ob-

jects against such subspace. The latter approach has the advantage of exploiting all the dimensions simultaneously, thus avoiding any loss of information that may arise from not considering relationships between sources (which is the case with the first approach). In particular, as for the user behavior source, we are investigating unsupervised approaches to model relationships among sources.

Evaluation is crucial to validate the implementation of the methodology. The main problem is the availability of datasets where information about user interaction behavior, the content of results and information objects are available. Transaction logs [7] can provide this data, but no explicit relevance judgments are available to validate the effectiveness of the approaches under investigation; existing datasets with this information are not publicly available.

6. CONCLUDING REMARKS

The purpose of this work was the introduction of a methodology that aims at exploiting evidence coming from multiple sources to predict the relevance of information objects for given queries. Four methodological steps are required to achieve this goal, namely, sources selection, features collection, dimension modeling and relevance prediction. The geometric framework proposed in [9] was chosen to implement the last two steps because it provides a uniform model for the sources, which can be used by to rank objects according to their estimated relevance.

Moreover, we discussed some issues to be addressed when implementing the methodology for two specific sources, that is the user interaction behavior and the latent semantic of the terms in the information objects. The issues specifically concern the evidence collection and source modeling steps.

In future work we want to further investigate the concepts adopted in this paper, namely, *entity*, *relationship*, *dimension* and *feature*. We chose these concepts as they relate to the view of the world to be modeled – in our case in order to predict relevance – which consists of entities and relationships, where the entities exists independently of their properties. The properties, namely the sources, are the information that can be obtained by the observation of entities and relationships between them. This is the same view of the world adopted in the Entity-Relationship (ER) model [22], the most widely used data model for the conceptual design of databases. In the ER model the result of the observation is a *value* and the mapping from the entities set (or the relationship set) to the value set is named *attribute*. The notion of feature adopted in this work can be compared to the ER notion of value set. Moreover the notion of dimension can be compared to the notion of attribute, since both refers to properties of entities and relationships.

The above discussion suggests investigate the relationships among the ER model, the geometric framework proposed in [9] and the methodology proposed in this paper.

Acknowledgements This research is partly funded by a Royal Society International Joint Project (2008/R4). Mounia Lalmas is currently funded by Microsoft Research/Royal Academy of Engineering.

7. REFERENCES

[1] S. E. Robertson, M. E. Maron, and W. S. Cooper. Probability of relevance: A unification of two competing models for document retrieval. *Information Technology: Research and Development*, 1(1):1–21, 1982.

[2] D. Kelly. *Understanding implicit feedback and document preference: a naturalistic user study*. PhD thesis, New Brunswick, NJ, USA, 2004.

[3] R. W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of CIKM'06*, pages 297–306, New York, NY, USA, 2006. ACM.

[4] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of KDD '02*, pages 133–142, New York, NY, USA, 2002. ACM.

[5] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of SIGIR '06*, pages 3–10, New York, NY, USA, 2006. ACM.

[6] M. Melucci and R.W. White. Utilizing a geometry of context for enhanced implicit feedback. In *Proceedings of CIKM'07*, pages 273–282, Lisbon, Portugal, 2007.

[7] B. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library & Information Science Research*, 28(3):407–432, 2006.

[8] W.B. Croft. Combining approaches to information retrieval. *Advances in information retrieval*, 7:1–36, 2000.

[9] M. Melucci. A basis for information retrieval in context. *ACM TOIS*, 26(3):1–41, 2008.

[10] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[11] R.W. White, J.M. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *IP&M*, 42(1):166–190, 2006.

[12] N. Fuhr. A probability ranking principle for interactive information retrieval. *Information Retrieval*, 11(3):251–265, 2008.

[13] M. Agosti, and F. Crestani A methodology for the automatic construction of a hypertext for information retrieval *Proc. of ACM SAC*, 745–753, Indianapolis, Indiana, United States, 1993.

[14] T. Tsikrika and M. Lalmas. Combining evidence for web retrieval using the inference network model: an experimental study. *IP&M*, 40(5):751–772, 2004.

[15] M. Lalmas and I. Ruthven. Representing and retrieving structured documents using the Dempster-Shafer theory of evidence: modelling and evaluation. *Journal of Documentation*, 54:529–565, 1998.

[16] S. K. M. Wong and V. V. Raghavan. Vector space model of information retrieval: a reevaluation. In *Proc. of SIGIR '84*, pages 167–185, Swinton, UK, 1984. British Computer Society.

[17] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JASIS*, 41:391–407, 1990.

[19] E. Di Buccio and M. Melucci. University of Padua at TREC 2009: Relevance Feedback Track. In *Proc. of TREC 2009*, Washington, DC, USA, 2009. To Appear.

[20] E. Di Buccio and M. Melucci. Towards a Methodology for Contextual Information Retrieval. In *Proc. of CIRSE 2009*, Tolouse, France, 2009.

[21] C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, New York, NY, USA, 2004.

[22] P.P. Chen. The entity-relationship model—toward a unified view of data. *ACM TODS*, 1(1):9–36, 1976.

Sentence-Based Active Learning Strategies for Information Extraction

Andrea Esuli, Diego Marcheggiani* and Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi, 1 – 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

ABSTRACT

Given a classifier trained on relatively few training examples, *active learning* (AL) consists in ranking a set of unlabeled examples in terms of how informative they would be, if manually labeled, for retraining a (hopefully) better classifier. An important text learning task in which AL is potentially useful is *information extraction* (IE), namely, the task of identifying within a text the expressions that instantiate a given concept. We contend that, unlike in other text learning tasks, IE is unique in that it does not make sense to rank individual items (i.e., word occurrences) for annotation, and that the minimal unit of text that is presented to the annotator should be an entire sentence. In this paper we propose a range of active learning strategies for IE that are based on ranking individual sentences, and experimentally compare them on a standard dataset for named entity extraction.

Keywords

Information extraction, named entity recognition, active learning, selective sampling

1. INTRODUCTION

In many applicative contexts involving supervised learning, labeled data may be scarce or expensive to obtain, while unlabeled data, even sampled from the same distribution, may abound. In such situations it may be useful to employ an algorithm that ranks the unlabeled examples and asks a human annotator to label a few of them, starting from the top-ranked ones, so as to provide additional *highly informative* training data. The task of this algorithm is thus to rank the unlabeled examples in terms of how informative they would be, once labeled, for the supervised learning task. The discipline that studies these algorithms is called *(pool-based) active learning* (aka *selective sampling*). This paper focuses on the application of active learning to *information extraction* (IE), the task of annotating sequences of one or more words (aka *tokens*) in a text by means of *tags* representing concepts of interest. The hypothetically perfect IE system is thus the one for which, for each tag in the *tagset* of interest, the *predicted sequences* of tokens coincide with the *true sequences*.

In text classification and other text learning tasks different from IE, the units of ranking and the units of annotation are the same; e.g., in text classification, it is the texts themselves that are ranked, and it is the texts themselves that are then annotated in their entirety by the human annotator. IE is peculiar from this standpoint since, while the units of annotation are the tokens, it does not make sense to *rank* individual tokens: if this were to happen, an annotator would be presented with "tokens in context" (i.e., a token in the fixed-size window of text in which the token occurs) and asked to annotate the token, with the consequence that she might be asked to read the same context several times, for annotating neighbouring tokens.

In this paper we take the view that the optimal unit of ranking is the *sentence*. This means that all the sentences of the automatically annotated texts are going to be ranked and presented to the annotator, who will then annotate *all* the tokens of a few sentences, starting from the top-ranked ones. This is different from several other works in the field [6, 8, 12], in which the unit of ranking is a portion of text smaller than a sentence, i.e., a predicted sequence embedded in a fixed-sized text window a few words long. The problem with the latter approach is that, by focusing on *predicted* sequences, the classification mistakes that the annotator corrects are the false positives, while the false negatives are never brought to the light. This results in an imbalanced training set being fed to the learner.

We deem the sentence to be the optimal unit of ranking for additional reasons:

- An entire sentence offers more context for actually interpreting the tokens and the sequences within it than the fixed-size window often used in the literature. This is especially important in complex IE tasks such as opinion extraction (see e.g., [2, 5]), in which, given the variety of devices that language has for conveying opinions, and given the uncertain boundary between fact and opinion, the annotator needs to take very subtle decisions.

- Different sentences never overlap, while different fixed-length windows may do. The sentence-based approach results in smaller annotation effort, since the same token is never examined twice by the annotator.

- From a semantic point of view, sentences are fairly self-contained units. This means that using portions of

*Corresponding author

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

text *larger* than sentences (e.g., paragraphs) as ranking units is unnecessary, also given that it is hardly the case that an annotation crosses the boundary between two consecutive sentences. Conversely, with a fixed-size window centered around a predicted sequence, another true sequence may cross the boundary between the window and its neighbouring text.

In the past, typical strategies adopted in AL for generic learning tasks have relied on ranking objects based either on the classification score attributed by the classifier to the object (*relevance sampling*), or on the confidence score with which the classifier has classified it (*uncertainty sampling*) [9]. In IE, if we want to rank entire sentences we have to come to terms with the fact that *each* token in the sentence has obtained a classification and a confidence score for *each* tag in the previous classification round, and we thus have to generate a sentence-specific score out of the token- and tag-specific scores, for all the tokens contained in the sentence and all the tags in the tagset.

The main contribution of this paper consists in proposing several alternative strategies for combining the token- and tag-specific scores into a sentence-specific score, and comparing these strategies experimentally.

We remark that this paper does *not* deal with active learning algorithms for specific supervised learning devices (such as e.g., [13] for text classification), but presents active learning strategies that are independent of the learning device and that are thus in principle suitable for use with any such device.

The rest of the paper is organized as follows. Our strategies for performing AL in IE are described in Section 2. In Section 3 we move to describing our experiments and the experimental protocol we have followed. We conclude in Section 4 by pointing out avenues for future work.

2. ACTIVE LEARNING STRATEGIES FOR INFORMATION EXTRACTION

2.1 Preliminaries: Information Extraction

This paper focuses on the application of active learning to (*single-tag*) *information extraction* (STIE, or simply IE). Let a text T consist of a sequence $T = \{t_1 \prec s_1 \prec \ldots \prec s_{n-1} \prec t_n\}$ of *tokens* (i.e., word occurrences) and *separators* (i.e., sequences of blanks and punctuation symbols), where "$\prec$" means "precedes in the text". Let $C = \{c_1, \ldots, c_m\}$ be a predefined set of *tags* (aka *labels*, or *classes*), and let $c_\emptyset \notin C$ be a special tag (to be read as "no tag"). We define (single-tag) information extraction as the task of estimating an unknown *target function* $\Phi : T \rightarrow C \cup \{c_\emptyset\}$ that specifies the true tag in $C \cup \{c_\emptyset\}$ attached to each token $t_i \in T$ and to each separator $s_i \in T$. The result $\hat{\Phi} : T \rightarrow C \cup \{c_\emptyset\}$ of this estimation is called the *tagger* (or *wrapper*, or *classifier*)[1]. A further property of both Φ and $\hat{\Phi}$ is that they can attribute a tag c_j to a separator s_i only if they also attribute the same tag to both t_{i-1} and t_i.

In most IE tasks it is actually the case that, rather than isolated tokens and separators, *sequences* of consecutive tokens and separators are annotated with a given tag; e.g., the sequence "George W. Bush", containing three tokens and two separators, might be annotated with the PER ("person name") tag. Such sequences of tokens will here be referred to as *annotated sequences* (ASs); the expressions *true AS* and *predicted AS* will refer to ASs according to Φ and $\hat{\Phi}$, respectively. Note that the reason for considering separators to be the object of tagging too is that the IE system should correctly identify sequence boundaries. For instance, given the expression "Barack Obama, Hillary Clinton and Joe Biden" the perfect IE system will attribute the PER tag, among others, to the tokens "Barack", "Obama", "Hillary", "Clinton", and to the separators (in this case: blank spaces) between "Barack" and "Obama" and between "Hillary" and "Clinton", but *not* to the separator ", " between "Obama" and "Hillary". If the IE system does so, this means that it has correctly identified the boundaries of the sequences "Barack Obama" and "Hillary Clinton".

Note that "single-tag" IE means that each token (resp., separator) has exactly one tag. This is different from multi-tag IE, in which it is assumed that a given token (resp., separator) may have more than one tag (opinion extraction – see e.g., [5] – is an instance of multi-tag IE).

2.2 Sentence-Based AL strategies for IE

Our experimental work is focused on comparing a range of active learning strategies for IE that are based on ranking individual sentences. This section describes the strategies and the intuitions supporting them.

In this work we test two alternative learning devices, *support vector machines* (SVMs) (see e.g., [1]), and *conditional random fields* (CRFs) [7]. For SVMs we have adopted a widely used method to realize a multiclass classifier as a combination of binary classifiers, i.e., a *one versus all* method. The one versus all method consists in learning m binary classifiers $\hat{\Phi}_c : T \rightarrow \mathbb{R}$, each one trained using as the positive examples all the tokens in the training set Tr that are labeled with c, and as negative examples all the other tokens, regardless of the original label. The multiclass classifier is then defined as $\hat{\Phi}(t) = \arg\max_{c \in C \cup \{c_\emptyset\}} \hat{\Phi}_c(t)$, i.e., the assigned label is the one whose binary classifier scored the maximum confidence.

CRFs are a discriminative probabilistic learning method based on an undirected graph model, and is frequently used for labeling sequential data, e.g., a sequence of words composing a text. Given a token t, a CRFs classifier estimates the likelihood $\hat{\Phi}_c(t) = P(c|t)$ for each $c \in C \cup \{c_\emptyset\}$ and, similarly to SVMs, the assigned label is the one scoring the highest $\hat{\Phi}_c(t)$ value. CRFs are nowadays considered the state-of-the-art learning device for information extraction [11].

The strategies we propose are based on two concepts, *label score* and *tag score*. The label score of a token is equal to $ls(t) = \max_{c \in C \cup \{c_\emptyset\}} \hat{\Phi}_c(t)$, i.e., the maximum confidence score that determines the decision taken by the classifier $\hat{\Phi}(t)$. The tag score is instead defined as $ts(t) = \max_{\{c \in C\}} \hat{\Phi}_c(t)$, i.e., the maximum confidence that the classifier as on considering a token as belonging to a tag, regardless of the confidence with respect to $c_\emptyset$.

2.2.1 Tag score-based strategies

The following strategies are based on combining the label scores assigned to the tokens in the sentence, following the intuition that the elements on which the classifier has low confidence could be more useful to the learner, so as to gather knowledge on "difficult" cases.

[1]Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

The *Min Min Confidence* (MMC) strategy assigns to the sentence a value equal to the minimum tag score value among the tokens composing it, i.e.,

$$MMC(s) = min_{t \in s}(ts(t)) \quad (1)$$

Sentence ranking is performed in increasing order of $MMC(s)$ value.

Min Average Confidence (MAC) is a version of MMC that tries instead to be robust to single "extreme" evaluations, averaging the tag scores of all the tokens composing the sentence, i.e.,

$$MAC(s) = avg_{t \in s}(ts(t)) \quad (2)$$

2.2.2 Label score-based strategies

Symmetrically to the tag-score-based strategies, the label-score-based strategies follow the somehow different intuition that the elements on which the classifier has high confidence could be useful, so that the strong beliefs of the learner are confirmed when correct or corrected when a blatant error is found.

The *Max Max Score* (MMS) strategy assigns to each sentence a value equal to the highest label score among the tokens composing it, i.e.,

$$MMS(s) = max_{t \in s}(ls(t)) \quad (3)$$

Sentence ranking is performed in decreasing order of $MAS(s)$.

Similarly to MAC, *Max Average Score* (MAS) instead averages the label scores of all the tokens composing the sentence, i.e.,

$$MAS(s) = avg_{t \in s}(ls(t)) \quad (4)$$

2.2.3 Tag count-based strategies

The following strategies are instead based on counting the number of tokens that are given a tag different from $c_\emptyset$ by the classifier.

The *Max Tag Count* (MTC) strategy counts the number of tokens in the sentence that are given a tag different from $c_\emptyset$, i.e.,

$$MTC(s) = |\{t \in s | \hat{\Phi}(t) \in C\}| \quad (5)$$

Sentence ranking is performed in decreasing order of $MTC(s)$ value.

Since MTC naturally favours long sentences, we have also tested a strategy (*Max Tag Ratio* – MTR) that normalizes the values by sentence length, i.e.,

$$MTR(s) = \frac{|\{t \in s | \hat{\Phi}(t) \in C\}|}{|s|} \quad (6)$$

The *Medium Tag Ratio* (MEDTR) strategy instead top-ranks the sentences with a tag ratio closer to the average tag ratio measured on the training set, i.e.,

$$MedTR(s) = \frac{MTR(s)}{avg_{s' \in Tr} MTR(s')} \quad (7)$$

2.2.4 Round Robin-based strategies

While the previous strategies always combine the confidence values returned on the various tag types, the following strategies are based on computing values separately for each tag, then selecting the most informative sentences using a "round robin" selection process across all the tags.

The *Round Robin Max Score* (RRMS) strategy assigns, for each $c \in C$, a relevance score to the sentence equal to the maximum score obtained by the tokens contained in it, i.e.,

$$RRMS_c(s) = max_{t \in s}(\hat{\Phi}_c(t)) \quad (8)$$

Then a round robin selection process is performed on the $|C|$ rankings produced.

Similarly to MAS, *Round Robin Average Score* (RRAS) uses averaging instead of maximization, i.e.,

$$RRAS_c(s) = avg_{t \in s}(\hat{\Phi}_c(t)) \quad (9)$$

The *Round Robin Max Tag Ratio* (RRMTR) strategy applies instead the MTR strategy considering the various tags separately from each other, so as to avoid favouring the most frequent tags over the most infrequent.

3. EXPERIMENTS

3.1 Experimental setting

The dataset we have used for evaluating our strategies is the CoNLL2003 named entity extraction dataset. The dataset consists of 1,393 Reuters newswire articles, for a total of 301,418 tokens. The tagset consists of 4 tags (LOC, PER, ORG, MISC, standing for "location", "person", "organization", and "miscellaneous", respectively) plus the special tag O, which tags any token / separator not tagged by any tag in {LOC, PER, ORG, MISC}. The tokens inside the corpus are tagged as follows: 10,645 tokens are tagged as LOC, 9,323 as ORG, 10,059 as PER, 5,062 as MISC, while the remaining 266,329 are tagged as O. We used a version of the CoNLL corpus already preprocessed with Pianta and Zanoli's Tagpro system [10], a PoS-tagging system based on YamCha that computes features such as prefixes, suffixes, orthographic information (e.g., capitalization, hyphenation) and morphological features, as well as PoS tags and chunk tags. These features altogether form the vectorial representations of tokens and separators that are fed to the learning device.

For this latter, we have tested two alternative, off-the-shelf packages, i.e., YamCha[2] and CRF++[3], respectively based on support vector machines and conditional random fields.

We evaluate the results of our experiments using the F_1 measure on a *token & separator* evaluation model [3]. The token & separator model considers each token and each separator as being the objects of tagging; for instance, given tag c, the TP ("true positives") entry of the contingency table for c consists in the number of tokens that are correctly assigned token c plus the number of separators that are correctly assigned token c. Once the contingency tables for all the tags in C have been filled, the evaluation is done by using standard micro-averaged and macro-averaged F_1.

3.2 Experimental protocol

In this work we adopt the following iterative experimental protocol. The protocol has three integer parameters α, β, and γ. Let Ω be a set of natural language sentences partitioned into a training set Tr and a test set Te, and let σ be an active learning strategy:

1. Set an iteration counter $t = 0$;

[2] http://www.chasen.org/~taku/software/YamCha/
[3] http://crfpp.sourceforge.net/

2. Set the current training set Tr_t to the set of the first α sentences of Tr; set the current "unlabeled set" $U_t \leftarrow Tr/Tr_t$;
3. For $t = 1, \ldots, \beta$ repeat the following steps:
   (a) Generate a classifier $\hat{\Phi}^t$ from the current training set Tr_t;
   (b) Evaluate the effectiveness of $\hat{\Phi}^t$ on Te;
   (c) Classify U_t by means of $\hat{\Phi}^t$;
   (d) Rank U_t according to strategy σ, thus generating the ranking $\sigma(U_t)$;
   (e) Let $r(U_t, \gamma)$ be the smallest prefix of $\sigma(U_t)$ (i.e., the smallest number of top-ranked elements of $\sigma(U_t)$) that contains at least γ tokens; set $Tr_{t+1} \leftarrow Tr_t \cup r(U_t, \gamma)$; set $U_{t+1} \leftarrow U_t/r(U_t, \gamma)$.

It is important to remark that Step 3b has only the purpose of collecting the results for experimental purposes (i.e., for producing the tables of Section 3.3); since it uses the test set Te, its results should obviously not be (and are not) accessible to the algorithm.

The above protocol simulates the activity of a human annotator who, at the beginning of the process, has available a training set Tr_0 consisting of α manually tagged sentences, and an "unlabeled set" U_0 consisting of $|Tr| - \alpha$ untagged sentences. The annotator generates a classifier $\hat{\Phi}^0$ from Tr_0, uses it to tag the sentences in U_0, asks the active learning agent to rank them, manually labels the top-ranked ones for a total of roughly γ tokens, generates a new classifier $\hat{\Phi}^1$ from an augmented training set that comprises Tr_0 and the newly tagged sentences, and repeats this process β times.

In our experiments we have set $\alpha = 110$ (in the CoNLL 2003 dataset this means approximately 2000 tokens), $\beta = 20$, and $\gamma = 200$; this means that each strategy will be evaluated by testing the accuracy of the classifiers generated from training sets consisting of approximately 2000, 2200, ..., 5800, 6000 training tokens, for a total 20 experiments per strategy. We think these parameters are realistic, since they simulate a situation in which

- there are only about 100 manually tagged sentences at the beginning; (this is reasonable, since in many applications in which significantly more training data are available, human annotators might not find it worthwhile to annotate any further);
- every time the human annotator manually labels 200 unlabeled tokens, he/she wants to retrain the system; (this is reasonable, since he/she wants the operate on a ranking of the unlabeled documents that incorporates as much as possible the feedback he/she has already given to the system;)
- the human annotator does not want to do any further manual labeling once about 6,000 training tokens are available; (this seems reasonable, since at this point the cost-effectiveness of the manual effort has probably decreased significantly.)

As the baseline strategy for the evaluation of our results we adopt the one that consists in adding further labeled sentences to the training set by picking them at random. This simulates the behaviour of a human annotator that picks unlabeled sentences and labels them in no particular order.

3.3 Results

The main results of our experiments are summarized in Table 1. This table reports, for each individual strategy, the values of F_1^μ and F_1^M obtained after 20 training sessions resulting from the protocol of Section 3.2, with $\alpha = 110$, $\beta = 20$, and $\gamma = 200$, using the two different learners, SVMs and CRFs.

Quite surprisingly, the only genuine strategy that outperforms the random baseline is the MAC strategy. The relative improvement of MAC over RAND ranges from 3.9% up to 6.3%. This improvement matches our expectations, given the close relation between the MAC strategy with the *uncertainty sampling* [9] method which already proved to be effective for AL.

Surprisingly, all the other strategies perform worse or no better than the random baseline. In order to understand the possible motivations behind these results we have inspected the sentences selected by the various strategies at the various iterations. This inspection allowed us to draw some specific conclusions on some of the strategies, and a general observation for the entire pool of strategies.

The MTR and RRMTR strategies tend to select very short sentences (two/three words) composed just by named entities. This allows gathering a lot of different instances of named entities, but without a context of use, which is important in order to learn how to perform extraction from longer, more articulated sentences.

The MTC strategy selects sentences of variable length, but tends to exceed in selecting sentences full of named entities, thus with a very limited amount of O-tagged tokens.

A common aspect of all the strategies is that, the more similar two sentences are, the more similar are the scores that the various strategies assign them. If the dataset contains a lot of similar sentences, and such sentences obtain high scores, the contribution of relevant information to the training set is limited, because of the redundancy contained in the set of sentences selected.

A comparison between the strategies based on round robin (RRAS, RRMS, RRMTR) against the respective "single-rank" versions (MAS, MMS, MTR) shows that the RR-strategies produce an improvement in the F_1^M measure, as should be expected when using a class-balancing method as RR.

The comparison of the averaging-based strategies (MAC, MAS, RRAS) against the respective versions based on maximization / minimization (MMC, MMS, RRMS) shows that averaging always perform better than maximization / minimization. This indicates that the smoothing introduced by the averaging helps the strategies to filter out the single "false-relevant" tokens that may appear in otherwise non-relevant sentences.

4. CONCLUSIONS

We have argued that, in active learning for information extraction, the sentence should be the unit of ranking. We have thus studied several strategies for scoring a given sentence for ranking, based on the classification score and the confidence score obtained by each token in the sentence. On the positive side, the experimental results that we have obtained by testing these strategies on a named entity extraction task show one such strategy (Min Average Confidence) to outperfom the others, irrespectively of learning device

		base	MAC	MAS	MMC	MMS	RRAS	RRMS	MTR	RRMTR	MTC	MedTR
F_1^μ	YamCha	.650	**.683**	.583	.645	.530	.639	.596	.628	.607	.632	.555
	CRF++	.656	**.697**	.610	.654	.463	.643	.573	.622	.509	.626	.544
F_1^M	YamCha	.634	**.661**	.525	.633	.526	.644	.577	.593	.597	.613	.538
	CRF++	.639	**.664**	.546	.634	.473	.633	.564	.563	.519	.606	.517

Table 1: Values of F_1 obtained after the last training session, i.e., with classifiers trained on approximately 2,000 training tokens plus approximately 4,000 tokens manually annotated as a result of the active learning strategy. Boldface indicates the best performance.

used (support vector machines or conditional random fields) and evaluation measure (microaveraged or macroaveraged F_1) used. On the negative side, the same results show that all the other strategies, that seem based on solid intuitions, tend to be roughly equivalent to a random strategy. In the future we plan to test these strategies further, possibly on IE tasks more difficult than named entity extraction such as opinion extraction.

Acknowledgments

We thank Emanuele Pianta and Roberto Zanoli for kindly providing us a version of the CoNLL corpus already preprocessed with their Tagpro system [10].

5. REFERENCES

[1] Christopher J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, Sydney, AU, 2006.

[3] Andrea Esuli, Michał Pryczek, and Fabrizio Sebastiani. Evaluating information extraction systems. Technical report, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 2010. Forthcoming.

[4] Andrea Esuli and Fabrizio Sebastiani. Active learning strategies for multi-label text classification. In *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, pages 102–113, Toulouse, FR, 2009.

[5] Andrea Esuli and Fabrizio Sebastiani. Enhancing opinion extraction by automatically annotated lexical resources. In *Proceedings of the 4th Language Technology Conference (LTC'09)*, pages 224–228, Poznań, PL, 2009.

[6] Rosie Jones, Rayid Ghani, Tom Mitchell, and Ellen Riloff. Active learning for information extraction with multiple view feature sets. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM'03)*, number 18–25, Cavtat–Dubrovnik, KR, 2003.

[7] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 282–289, Williamstown, US, 2001.

[8] Florian Laws and Hinrich Schütze. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 465–472, Manchester, UK, 2008.

[9] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 3–12, Dublin, IE, 1994.

[10] Emanuele Pianta and Roberto Zanoli. Tagpro: A system for Italian POS tagging based on SVMs. *Intelligenza Artificiale*, 4(2):8–9, 2007.

[11] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261—377, 2008.

[12] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 589–596, Barcelona, ES, 2004.

[13] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

A study on evaluation on opinion retrieval systems

Giambattista Amati
Fondazione Ugo Bordoni
Rome, Italy
gba@fub.it

Giuseppe Amodeo
Dept. of Computer Science,
University of L'Aquila
L'Aquila, Italy
gamodeo@fub.it

Valerio Capozio
Dept. of Mathematics,
University of Rome "Tor Vergata"
Rome, Italy
valeriocapozio@gmail.com

Carlo Gaibisso
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" - CNR
Rome, Italy
carlo.gaibisso@iasi.cnr.it

Giorgio Gambosi
Dept. of Mathematics,
University of Rome "Tor Vergata"
Rome, Italy
gambosi@mat.uniroma2.it

ABSTRACT

We study the evaluation of opinion retrieval systems. Opinion retrieval is a relatively new research area, nevertheless classical evaluation measures, those adopted for ad hoc retrieval, such as MAP, precision at 10 etc., were used to assess the quality of rankings. In this paper we investigate the effectiveness of these standard evaluation measures for topical opinion retrieval. In doing this we split the opinion dimension from the relevance one and use opinion classifiers, with varying accuracy, to analyse how opinion retrieval performance changes by perturbing the outcomes of the opinion classifiers. Classifiers could be studied in two modalities, that is either to re-rank or to filter out directly documents obtained through a first relevance retrieval. In this paper we formally outline both approaches, while for now focussing on the filtering process.

The proposed approach aims to establish the correlation between the accuracy of the classifiers and the performance of the topical opinion retrieval. In this way it will be possible to assess the effectiveness of the opinion component by comparing the effectiveness of the relevance baseline with that of the topical opinion.

Categories and Subject Descriptors

H.3.0 [**Information Storage and Retrieval**]: General; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory, Experimentation

Keywords

Sentiment Analysis, Opinion Retrieval, Opinion Finding, Classification

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

1. INTRODUCTION

Sentiment analysis aims to documents classification, according to opinions, sentiments, or, more generally, subjective features contained in text. The study and evaluation of efficient solutions to detect sentiments in text is a popular research area, and different techniques have been applied coming from natural language processing, computational linguistics, machine learning, information retrieval and text mining.

The application of sentimental analysis to Information Retrieval goes back to the novelty track of TREC 2003 [13]. Topical opinion retrieval is also known as *opinion retrieval* or *opinion finding* [4, 9, 11]. In [5, 3, 2, **?**] dictionary-based methodologies for topical opinion retrieval are proposed. An application of opinion finding to blogs was introduced in the Blog Track of TREC 2006 [8]. However, there is not yet a comprehensive study of evaluation of topical opinion systems, and in particular of the interaction and correlation between relevance and sentiment assessments.

At first glance, evaluation of opinion retrieval systems seems to not deserve any further investigation or extra effort with respect to the evaluation of conventional retrieval systems. Traditional evaluation measures, such as the Mean Average Precision (MAP) or the precision at 10 [8, 6, 10, 11], can be still used to evaluate rankings of opinionated documents that are also assessed to be relevant to a given topic. However, if we give a deeper look at the performance of topical opinion systems we are struck by the diversity in the observed values of performance. For example the best run for topic relevance in the blog track of TREC 2008 [10] achieves a MAP value equal to 0.4954, that drops to 0.4052, as concerns the MAP of opinion, in the opinion finding task. Performance degradation is as expected because any variable which is additional to relevance, i.e. the opinion one, must deteriorate the system performance. However, we do not have yet a way to set apart the effectiveness of the opinion detection component and evaluate how effective it is, or to determine whether and to which extent, the relevance and opinion detection components are influenced by each other. It seems evident that an evaluation methodology or at least some benchmarks are needed to make it possible to assess how effective the opinion component is. To exemplify: how

effective is the performance value of opinion MAP 0.4052 when we start from an initial relevance MAP of 0.4954? It is indeed a matter of fact that opinion MAP in TREC [8, 6, 10], seems to be highly dependent on the relevance MAP of the first-pass retrieval [9].

The general issue is thus the following: can we assume that absolute values of MAP can be used as they are to compare different tasks, in our case the topical opinion and the ad hoc relevance task; and thus: evaluation measures can be used without any MAP normalization to compare or to assess the state of the art of different techniques on opinion finding?

At this aim, we introduce a completely novel methodological framework which:

- provides a bound for the best achievable opinion MAP, for a given relevance document ranking;
- predicts the performance of topical opinion retrieval given the performance of the topic retrieval and opinion detection;
- viceversa, provides whether a given opinion detection technique gives a significant or marginal contribution to the state of the art;
- investigates the robustness of evaluation measures for opinion retrieval effectiveness.
- indicates what re-ranking or filtering strategy is best suited to improve topical retrieval by opinion classifiers.

This paper is organized as follows. The proposed evaluation method is presented in sections 2 and 4; section 3 introduces the collection used for tests. Results are presented in section 5, and conclusions follow in section 6.

2. EVALUATION APPROACH

An opinion retrieval system is based on a topic retrieval and an opinion detection subsystem [9]: different kinds of "information" are retrieved and weighted in order to generate a final ranking of documents that reflects their relevance with both topic and opinion content. To analyse the effectiveness of the whole system, we should be able to quantify not only the performance of the final result, but also the contribution of each subsystem. As usual, the evaluation metric used in literature for the final ranking is the MAP. But MAP (of relevance and opinion) for the final ranking is not sufficient to fully assess the performance of the whole system: the contribution of each component, taken separately, needs to be identified.

The input to the proposed topical opinion evaluation process is the relevance *baseline*, i.e. the ranking of documents generated by the topic retrieval system, here considered as a black box. The effectiveness of the topic retrieval component is measured by the MAP of opinion and relevance of this baseline.

The evaluation of the effectiveness of the opinion detection component, relies on artificially defined classifiers of opinion. The artificial classifier $\mathcal{C}_{\mathcal{O}}^{k}$ classifies documents as opinionated, $\mathcal{O}$, or not opinionated, $\overline{\mathcal{O}}$, with accuracy k, $0 \leq k \leq 1$. The classification process is independent from the topic relevance of documents. To achieve accuracy k $\mathcal{C}_{\mathcal{O}}^{k}$ properly classifies each document with probability k.

Therefore the number of misclassified documents is $(1-k)\cdot n$, where n is the number of classified documents. Assuming the independence between opinion and relevance, the misclassified documents will be distributed randomly between relevant and not relevant.

The outcomes of these artificial classifiers are then used to modify the baseline. This can be done following two different approaches:

- a *filtering process*: when documents of the baseline are deemed as not opinionated by the classifier, they are removed from the ranking;
- a *re-ranking process*: when documents of the baseline are considered as opinionated by the classifier, they receive a "reward" in their rank.

The filtering process uses the classifier in its classical meaning. This process is particularly suitable to analyse the effectiveness of the technique itself to opinion detection, as a classification task [12], and its effects on topical opinion performance. Opinion filtering also gives some interesting clues on what is the optimal performance achievable by an opinion retrieval technique based on filtering, and also whether filtering strategy is in general superior or not to even very simple re-ranking strategies.

In the re-ranking process a "reward" function for the documents has to be defined. In such a case we introduce bias in assigning correct rewards, and we thus may observe the effectiveness of a re-ranking algorithm as long as the opinion detection performance changes.

By "comparing" the results of an opinion retrieval system with the filtering process, or the re-ranking process at several levels of accuracy, we can obtain relevant clues about:

- the overall contribution introduced by the opinion system only and its robustness;
- the effectiveness of the opinion detection component;

In the following we formally describe both the approaches and focus on the experimentation concerning the filtering process only.

3. EXPERIMENTATION ENVIRONMENT

We used the BLOG06 [7] collection and the data sets of the Blog Track of TREC 2006, 2007 and 2008 [8, 6, 10] for our experimentation. Since 2006, Blog Track has an evaluation track on blogs where the main task is opinion retrieval, that is the task of selecting the opinionated blog posts relevant to a given topic [9]. BLOG06 collection size is 148 GB and contains spam as well as possibly non-blogs and non-English pages.

The data set consists of 150 topics and a list, the *Qrels*, in which the relevance and content of opinion of documents are assessed with respect to each topic. An item in the list identifies a topic t, a document d and a judgement of relevance/opinion assigned as follows:

- 0 if d is not relevant with respect to t;
- 1 if d is relevant to t, but does not contain comments on t;
- 2 if d is relevant to t and contains positive comments on t;

- 3 if d is relevant to t and contains neutral comments on t;
- 4 if d is relevant to t and contains negative comments on t.

Note that not relevant documents are not classified according to their opinion content.

In the following, $[x]$ denotes the set of documents labelled by an $x = 0, 1, 2, 3, 4$, and not labelled documents belong to [0] by default.

TREC organizers also provide the best five *baselines*, produced by some participants, denoted by $BL_1, BL_2, \ldots, BL_5$.

4. EVALUATION FRAMEWORK

The behaviour of artificial classifier $\mathcal{C}^k_{\mathcal{O}}$ is defined through the *Qrels*. $\mathcal{C}^k_{\mathcal{O}}$ predicts the right opinion orientation of each document in the collection by searching it in the *Qrels*. The accuracy k is simulated by the introduction of a bias in the classification. Documents not appearing or assessed as not relevant in the *Qrels*, will be classified according to the distribution of probability of opinionated and not opinionated documents among the relevant ones. Taking into account both relevance and opinion in the test collection we obtain the contingency Table 1. As shown in table 1, the *Qrels* does not provide the opinion classes for not relevant documents. The missing data complicate a little bit, but not much, the construction of our classifiers. To overcome the problem, we assume that

$$Pr(\mathcal{O}|\mathcal{R}) = Pr(\mathcal{O}|\overline{\mathcal{R}}) \tag{1}$$

Equation 1 asserts that there is not a sufficient reason to have a different distribution of opinion among relevant and not relevant documents. An a priori probability, $\Pr(\mathcal{O})$, for opinionated documents is still unknown. However equation 1 implies that $\mathcal{O}$ and $\mathcal{R}$ are independent, thus

$$Pr(\mathcal{O}|\mathcal{R}) = Pr(\mathcal{O}) \tag{2}$$

From equations 1 and 2 follows that

$$Pr(\overline{\mathcal{O}}|\mathcal{R}) = Pr(\overline{\mathcal{O}}|\overline{\mathcal{R}}) = Pr(\overline{\mathcal{O}}) = 1 - Pr(\mathcal{O}) \tag{3}$$

Equations 2 and 3 are equivalent to assume that the set $\{[2] \cup [3] \cup [4]\}$, as defined in Table 1, is a sample of the set of opinionated documents. Thus, without loss of generality, we can define $\Pr(O)$ using only the documents classified as relevant by the *Qrels* as follows:

$$P(\mathcal{O}) = \frac{|\{[2] \cup [3] \cup [4]\}|}{|\{[1] \cup [2] \cup [3] \cup [4]\}|} \tag{4}$$

and consequently

$$P(\overline{\mathcal{O}}) = 1 - P(\mathcal{O}) = \frac{|[1]|}{|\{[1] \cup [2] \cup [3] \cup [4]\}|} \tag{5}$$

In the following we study whether and how the set of relevant and not relevant documents classified as opinionated affects the topical opinion ranking.

We have to say that for both approaches, filtering or re-ranking, a misclassification may have controversial effects on the effectiveness of the final ranking. If we filter documents by opinions with a classifier, for example, the misclassified and removed not relevant documents may bring a positive contribution to the precision measures, because all opinionated and relevant documents that were below them, will have a higher rank after their removal. Even with the re-ranking approach we have a similar situation, but this precision boosting phenomenon is attenuated by the fact that re-ranking is not based on as drastic decision as that of a removal, and the repositioning of a document does not propagate to all documents that are below it in the original ranking.

	$\mathcal{O}$	$\overline{\mathcal{O}}$	
$\mathcal{R}$	$\|\{[2]\cup[3]\cup[4]\}\|$	$\|[1]\|$	
$\overline{\mathcal{R}}$	NA	NA	

Table 1: the contingency table for an opinion-only classifier for documents in the BLOG06 collection. $\mathcal{R}$ denotes relevance, $\overline{\mathcal{R}}$ non-relevance; $\mathcal{O}$ denotes opinion, $\overline{\mathcal{O}}$ non-opinion. With the notation [x] we refer to the class of documents labelled by $x = 1, 2, 3, 4$ in the *Qrels*.

Together with $\mathcal{C}^k_{\mathcal{O}}$, we introduce a random classifier $\mathcal{C}^{RC}_{\mathcal{O}}$ that classifies documents according to the a priori distribution of opinionated documents in the collection. It represents a good approximation of the random behaviour of a classifier. More precisely, this classifier assesses a document as opinionated with probability $P(\mathcal{O})$ and as not opinionated with probability $\Pr(\overline{\mathcal{O}}) = 1 - \Pr(\mathcal{O})$.

4.1 Filtering approach

As already stated, in the filtering approach documents classified as not opinionated are removed from the baseline. Note that while relevant documents contribute and improve the evaluation measure, if correctly classified, the not relevant ones do not contribute directly to this measure.

In conclusion if a not relevant document is classified as opinionated not being actually opinionated, then this misclassification will not affect the evaluation measure. Differently the removal of not relevant documents regardless of their real opinion orientation, always positively affects the ranking, even if misclassified.

For relevant documents instead the misclassification always negatively affects the ranking.

With this approach we can observe how hard is to overcome the baseline, i.e. we can identify how effective must be the opinion detection technique to improve the starting topic retrieval.

4.2 Re-ranking approach

Re-ranking techniques essentially are fusion models [9] that combine a relevance score $s_{\mathcal{R}}(d)$ and an opinion score $s_{\mathcal{O}}(d)$ (or two ranks derived from these scores) for a document d. The new score $s_{\mathcal{OR}}(d)$ is a function of the two non negative scores, $s_{\mathcal{R}}(d)$ and $s_{\mathcal{O}}(d)$:

$$s_{\mathcal{OR}}(d) = f(s_{\mathcal{R}}(d), s_{\mathcal{O}}(d)) \tag{6}$$

Given a classifier $\mathcal{C}^k_{\mathcal{O}}$, we define a new score $s^{\mathcal{C}}_{\mathcal{OR}}(d)$ based on the outcomes of $\mathcal{C}^k_{\mathcal{O}}$ according to which the baseline is re-ranked. $s^{\mathcal{C}}_{\mathcal{OR}}(d)$ is defined as follows:

$$s^{\mathcal{C}}_{\mathcal{OR}}(d) = \begin{cases} f(s_{\mathcal{R}}(d), s_{\mathcal{O}}(d)) & \text{if } d \in_{\mathcal{C}^k_{\mathcal{O}}} O \\ f(s_{\mathcal{R}}(d), 0) & \text{if } d \notin_{\mathcal{C}^k_{\mathcal{O}}} O \end{cases} \tag{7}$$

where $\in_{\mathcal{C}^k_{\mathcal{O}}}$ denotes the classifier outcome, that is when the document is assigned to a given class. Note when $k = 100\%$

and assuming that $f(\cdot,\cdot)$ is a not decreasing function of $s_{\mathcal{O}}(\cdot)$, i.e. $f(s_{\mathcal{R}}(d), x) \geq f(s_{\mathcal{R}}(d), x')$, $\forall x \geq x'$, the opinion MAP of any ranking based on $s_{\mathcal{OR}}(\cdot)$ does not exceed that based on $s^{\mathcal{C}}_{\mathcal{OR}}(\cdot)$.

All the above considerations can be further extended to the case in witch the $s_{\mathcal{OR}}(d)$ is based on the ranks of d instead of on its scores (of relevance and opinion).

5. EXPERIMENTATION RESULTS

In this paper we report the experimentation results for the filtering approach. The filtering process has been repeated 20 times for each baseline and for accuracy $k = 0.5$, 0.6, 0.7,0.8,0.9,1. Mean values of the MAPs are reported.

Table 2 reports, in decreasing order, the relevance MAPs (MAP_R) and the opinion MAPs (MAP_O) for each baseline.

Baselines		
	MAP_R	MAP_O
BL_4	0.4776	0.3542
BL_5	0.4247	0.2974
BL_3	0.4079	0.3007
BL_1	0.3540	0.2470
BL_2	0.3382	0.2657

Table 2: MAP of relevance (MAP_R) and opinion (MAP_O) of the five baselines.

In figure 1 MAP values are reported for each baseline as long as the accuracy of classifiers changes. The dotted lines represent the baselines opinion MAPs and the dot-dashed lines represent the baseline relevance MAPs. The MAP values of random classifier is also reported as the dashed lines in the graphs.

Analysing the MAP trend we can infer the following observations:

1. the baseline MAP_R is an upper bound for the MAP_0 obtained with a filtering approach;
2. the random classifier always deteriorate the performance of the baseline MAP_0.
3. the minimal accuracy needed to improve by filtering the baseline MAP_0 is very high, at least 80%;
4. there is a linear correlation between the MAP_0 achievable by a classifier with accuracy k and the accuracy itself.

First three remarks says that filtering strategy is very dangerous for MAP_0 performance, that is removing documents affects greatly the performance of the topical opinion retrieval.

From the above considerations, we may conclude that the opinion retrieval task is not easy and that having good results with a filtering approach requires a too high accuracy. The experimentation instead allows us to identify a plausible range for the MAP achievable by an opinion retrieval system: the classifier with accuracy 100% and the random classifiers obtains performance that can be considered as thresholds for the best and the worst opinion detection system. It is also evident that higher the baseline MAP is, higher the accuracy of classifier must be to introduce some benefits with a filtering approach with respect to relevance only retrieval.

6. CONCLUSIONS AND FUTURE WORKS

The opinion retrieval problem seems to be a relatively hard task: the combination of two variables like topic relevance and opinion, requires a deep analysis on their correlation. From the results of TREC competitions [8, 6, 10, 9], emerges the lack of exhaustive evaluations measures: the MAP, Precision at 10 and R-Precision are not sufficient alone to give a complete analysis on the systems performances.

Up to now we have studied only the filtering of documents by opinions. This strategy however requires a very high accuracy of the classification. We will compute the study with re-ranking approach starting from the approach used in [1, 2].

Our approach is able to provide an indicative accuracy of the opinion component of the topical opinion retrieval system. It also allows us to propose an evaluation framework, able to evaluate the effectiveness of opinion retrieval systems.

7. REFERENCES

[1] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. Fub, iasi-cnr and university of tor vergata at trec 2007 blog track. In *Proc. of the 16th Text Retrieval Conference (TREC)*, 2007.

[2] G. Amati, G. Amodeo, M. Bianchi, C. Gaibisso, and G. Gambosi. *A uniform theoretic approach to opinion and information retrieval*, in *Intelligent Information Access*, G. Armano, M. de Gemmis, G. Semeraro, and E. Vargiu (eds.) Studies in Computational Intelligence. Springer, to appear.

[3] J. Skomorowski and O. Vechtomova. Ad hoc retrieval of documents with topical opinion. In G. Amati, C. Carpineto, and G. Romano, editors, *ECIR*, volume 4425 of *Lecture Notes in Computer Science*, pages 405–417. Springer, 2007.

[4] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 345–354, Sydney, Australia, July 2006. Association for Computational Linguistics.

[5] G. Mishne. Multiple ranking strategies for opinion retrieval in blogs. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[6] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the trec-2007 blog track. In *Proc. of the 16th Text Retrieval Conference (TREC)*, 2007.

[7] Crag Macdonald and Iadh Ounis. The trec blogs06 collection : Creating and analysing a blog test collection. Technical report, University of Glasgow Scotland, UK, 2006.

[8] I. Ounis, M. de Rijke, C. Macdonald, G. A. Mishne, and I. Soboroff. Overview of the trec-2006 blog track. In *TREC 2006 Working Notes*, 2006.

[9] I. Ounis, C. Macdonald, and I. Soboroff. On the trec blog track. In *Proc. of the 2nd International Conference on Weblogs and Social Media (ICWSM)*, 2008.

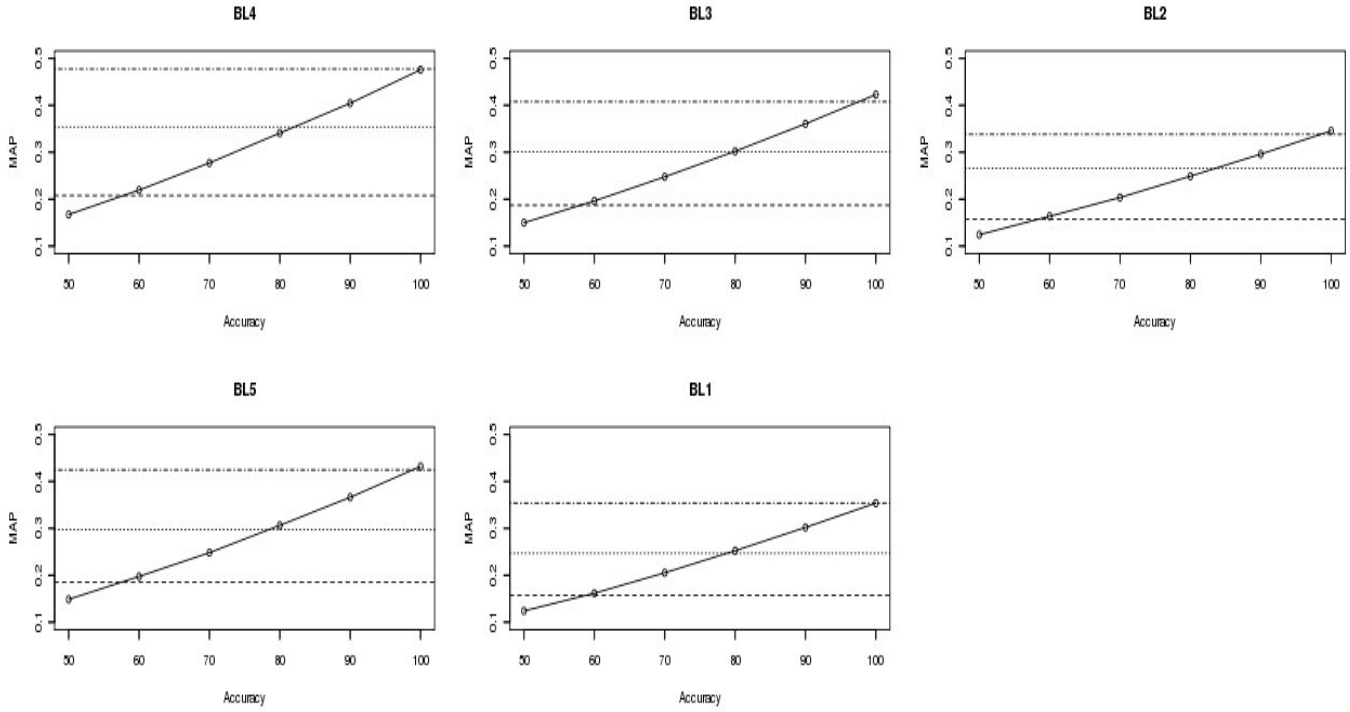


Figure 1: MAPs of opinion of the baselines filtered by $\mathcal{C}^k_{\mathcal{O}}$ and for $k = 0.5, 0.6, 0.7, 0.8, 0.9, 1$. The opinion MAPs (dotted lines) and relevance MAPs (dot-dashed lines) of the baselines are also reported. Finally dashed lines show the opinion MAPs for the baselines filtered by $\mathcal{C}^{RC}_{\mathcal{O}}$.

[10] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the trec-2008 blog track. In *Proc. of the 17th Text Retrieval Conference (TREC)*, 2008.

[11] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proc. of the ACL-02 conference on Empirical Methods in Natural Language Processing*, pages 79–86, 2002.

[13] Ian Soboroff and Donna Harman. Overview of the trec 2003 novelty track. In *TREC*, pages 38–53, 2003.

A Cluster Manipulation Paradigm for Mobile Web Search Interaction

Gloria Bordogna
CNR-IDPA
Via Pasubio 5
24044 Dalmine (BG)
Italy
gloria.bordogna@idpa.cnr.it

Alessandro Campi
Politecnico di Milano
DEI
Piazza L. da Vinci 32
20133 Milano
Italy
campi@elet.polimi.it

Stefania Ronchi
ronchi@polimi.it

Giuseppe Psaila
Università di Bergamo
Facoltà di Ingegneria
Viale Marconi 5
24044 Dalmine (BG)
Italy
psaila@unibg.it

ABSTRACT

This paper describes a new interaction paradigm well suited to perform web searches though a mobile device. The prototypal system that implements this novel interaction framework is named *Matrioshka*, that is a multi-modal system. In this paper we focus on the interaction framework and will introduce briefly an overview of the mobile version of *Matrioshka*. This framework is based on cluster manipulation operations. The results of a user request, yielded by one or more search engines, are organized into labelled clusters. Then, some manipulation operators can be applied to re-rank clusters or to combine them to generate new clusters. These facilities allow the user to capture the relevant documents hidden in the large set of retrieved ones in the first ranked clusters.

Keywords

Web searches, mobile information retrieval, results clustering, ranking strategies

1. INTRODUCTION

The large diffusion of Internet connections from anywhere at anytime has arisen the problem of more effective ways of searching the Web from mobile devices. In this paper, a mobile interaction framework for web meta-searching is proposed, whose definition is motivated by the observation that the visualization method based on the ranked list of web pages is too long to fit small screens such as those of mobile devices. Further, with the aid of a mobile keyboard, the usual way of interacting with search engines based on repeated cycles of query reformulation imposes too much burden to the user. At the same time, it is too expensive in terms of the high cost of mobile connections. In fact, if users do not find what they are looking for in the first one or two result pages, they are more keen to reformulate a new query than to analyze successive pages, or to submit the current query to another search engine.

To overcome these drawbacks, some search services such as *vivisimo*, *clusty*, *Snaket*, *Ask.com* (at [1]), *MS AdCenter Labs Search Result Clustering*, etc., proposed to cluster the results of Web searches. W.r.t. the ranked list, clustered results are more compact and offer an overview of the main topics dealt with in much more documents than those contained in the first few pages, that would be missed otherwise [8, 16, 11]. As far as we know, in the literature we found only one academic mobile search engine, named Credino [2] that exploits clustered results.

On the other side, one problem users encounter with such clustered results, is the inability of fully understanding the contents of the clusters. This is mainly due to the short and sometimes bad quality of the clusters' labels, which generally consist of a few terms, or individual short phrases, which are automatically extracted from the documents within clusters. Often, several clusters have similar labels, which differ just for a single term. To effectively explore the cluster contents, users have no other means than clicking on the cluster labels and browsing the clusters themselves. On a mobile device, this modality would again require too much scrolling.

The idea of our proposal is to maintain the result clustering paradigm, and to provide users with a language to manipulate clusters. Both several ranking criteria to differently order the clusters, and operators to combine the clusters themselves are defined whose final aim is to make possible the exploration of the retrieved contents.

The literature on mobile search engines mainly focuses on modelling the user context, considering primarily the user geographic location, in order to filter the retrieved results [10]; other topics are the summarization of documents [7], and the definition and use of data visualization schemes [13]. In [6] the clustering of retrieved results is proposed as a useful way of presenting the search results on small screens, but, to the best of our knowledge, only the mobile search engine *Credino* [2] performs clustering.

The manipulation language as a basis for a flexible interaction makes our proposal substantially different from *Credino* [2], where the focus is the clustering algorithm it adopts w.r.t. other clustering methods, and does not offer criteria to explore the cluster contents.

A motivation of utility of the manipulation language can be found in [12] which advocates the need of tools for giving the user more immediate control over the clusters of retrieved web documents. Our proposal can be particularly useful when groups of clusters with same or almost same labels are generated by distinct requests or by the same query submitted to distinct search engines. In such situations it

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

becomes necessary to explore the contents of the clusters and their relationships in terms of number of contained documents, relevance of contents, homogeneity of contents, or common and distinct contents with other clusters. This task an *exploratory* task, that may last for a long time, and may require to reuse the intermediate results several times. For this reason, storing of the intermediate results into a database is essential for successive manipulation. Furthermore, the local manipulation of results avoids the useless overloading of both the network and the search engines. In fact, in current practices, several modified queries are submitted to the search engines, trying to capture relevant documents in the first positions of the ranked list; note that most of these documents were already retrieved by the previous queries, although hidden to the user since they did not occur in the first positions.

In [4] and [5], we proposed and defined the operators for combining the clusters for revealing their implicit relationships. In [3] a prototypal mobile meta search system was proposed that allows easily using the combination operators.

In this paper we propose an extension of the manipulation language by introducing a ranking operator that makes possible the exploration of the cluster contents based on distinct properties of the clusters.

2. THE INTERACTION FRAMEWORK

Data Model. Here we describe the data model on which the proposed interaction framework is based. We start considering a *query q* submitted to a search engine; its result is a ranked list of documents, that we call *items*.

Definition 1: **Item** An *item i* represents (an *instance* of) a document retrieved by a web search. It is described by the following attributes: *uri*, which is the *Uniform Resource Identifier* of the ranked web document; *title* and *snippet* which are, respectively, the document title and snippet[1]; finally, *irank* is a score (in the range $[0, 1]$) that expresses the estimated relevance of the retrieved document w.r.t. the query. □

The same document (web page) may be represented by distinct items in distinct result lists. In facts, we assume that a document is uniquely identified by its *uri* [9], while it may have distinct *snippets*, *irank* and *title*, when retrieved by different search services (or by different queries). We assume that *irank* is a function of the position of the item in the query result list.

In our system, the results of a user request (or exploration) are not simply a ranked list of documents, but they are gathered in ordered *clusters*.

Definition 2: **Cluster** A *cluster c* is a set of items, having a rank. It is defined by two attributes: *label* is a set of terms that semantically synthesize the main content of the cluster; *crank* is a score (in the range $[0, 1]$) depending on some property of the cluster. □

A cluster *label* is automatically generated by a specific labelling algorithm on the basis of frequent terms in cluster items [3].

At this point we define the main element of the data model.

Definition 3: **Group** A *group g* is a non empty, ordered set of clusters. It is described by the following attributes: *label*, a set of terms that semantically synthesizes the main content of the group; *s*, the name of the search engines used to retrieve the items in the clusters of the group. □

Finally we define the users' History repository.

Definition 4: **History** A *history H* is a set of *items*. It can be the empty set, at the beginning of a search session, and it can be updated by explicit action of the user when he/she decides to save a retrieved document. □

Manipulating Clusters. The procedure that generates a group is initially activated by a search operator, named *CQuery*, that allows users to query a search engine (e.g., *Google*, *Yahoo*!, *MSN Search*) and to cluster the results. In the implementation we considered a maximum of N documents, with n $\geq$ 30, i.e., a number of documents greater than that retrieved in the first three pages, those usually analyzed by a common user.

On this basis, for each retrieved document, the operator builds an item i, whose *irank* value depends on the position of the document in the result list: $i.irank = (N - Pos(d) + 1)/N$ (where $Pos(d)$ is the position of the document in the query result list). In this way, a document in the first positions has a rank $r.irank$ very close to 1. This is done in order to achieve independence and comparability of the ranking produced by distinct search engines.

The ranked list obtained as a result by the search operator, is then clustered by applying the *Lingo* algorithm [14]. Lingo is used to perform a flat crisp clustering of the query results on the basis of their snippets and titles. Once clusters are obtained, they are labelled. Finally also the groups are labelled (see [3] for the labelling algorithm) to synthesize the most central contents retrieved by all their clusters.

Successively, one can decide either to explore the groups of clusters retrieved by a single query by applying some ranking operation described in 2.1 which evaluates a cluster property, or one can generate other groups by combining the obtained ones through the operators defined in Section 2.2.

2.1 Cluster Ranking Methods

Once the results of a query are obtained as a group of ranked clusters, in which the default *crank* score is computed as the average of the *irank* of its documents, the user has the possibility to re-rank the clusters based on the evaluation of some other clusters' property. This allows to obtain, in the first positions of the ranked list of clusters, those clusters that previously could appear in the last positions. This is the novel contribution of the paper w.r.t. our previous work: the user is this way provided with the possibility of evaluating groups by different perspectives.

The *cluster properties* that can be considered for the ranking are the following:

• *Relevance*: this is defined as the average of the relevance scores of documents belonging to the cluster and is the default property for the ordering of clusters; the relevance scores of clusters are the *irank* values computed as previously defined from the documents' positions in the ranked list returned by the search engine. Ordering clusters by decreasing values of their relevance means being interested primarily in the relevance of documents contained in the clusters.

• *Ponderosity*: this is defined as the cluster cardinality, and it measures how many documents belong to the clusters; the ranking of clusters in decreasing order of their ponderosity can be useful for users interested in high recall.

[1]The snippet is an excerpt of the document, made by a set of sentences that may contain the keywords of the query

• *Heterogeneity*: this is defined as the variance of the documents vectors, represented in the space of index terms extracted from their titles and snippets, and weighted by their relative frequency, w.r.t. the cluster centroid vector, defined as the average vectors of all the documents vectors belonging to the cluster. The greater the variance the more heterogeneous is the cluster: by choosing to rank clusters in increasing order of their heterogeneity means being interested in contents focalized on the specific meaning expressed by the label of the cluster, since the cluster label is generated from its centroid vector. This can be useful in target searches.
Conversely, by choosing to rank clusters in decreasing order of their heterogeneity means being more tolerant on the meaning expressed by the cluster label; this can be useful when one is unsure to have expressed by the query the actual information needs and wants to soften the selection conditions.
• *Novelty*: this is defined as the proportion of novel documents contained in the cluster w.r.t. previously already seen documents, that the user has saved in the *history* repository; choosing a novelty ranking means being interested in new documents on the topics of a search and can be useful in the context of bibliographic surveys.

In order to rank clusters of a group based on one of the above properties the operation *ClusterRank* is defined:

$$g' = ClusterRank(g, property, order)$$

in which g and g' are the input and output groups of clusters, *property* takes values in a set of strings {*Relevance*, *Ponderosity*, *Etherogeneity*, *Novelty*} denoting a cluster property; $order \in$ {*increasing*, *decreasing*} indicates the desired ordering, i.e., increasing and decreasing w.r.t. the value of the specified cluster property, respectively.
g' has the same label of g and contains the same clusters of g with the only difference that the clusters' *crank* scores are computed based on the specified *proverty* of the clusters:

$$crank_i = \frac{property(c_i)}{MAX_k(property(c_k))}$$

2.2 Combining Groups of Clusters

The system provides users with the possibility to interact with the results of search services organized in groups of clusters, in order to get more satisfactory and refined results to their needs. To this aim, the user can choose to apply different sequences of operators on selected groups, in order to recombine (modify, explore) their structure and content.

The operators that we are going to illustrate are formally defined in [4]; they are inspired by the operators provided by the Relational Algebra (i.e. intersection, join, union etc.), thought they are specifically defined for groups of clusters. They generate, starting from two input groups g_1 and g_2, one group g' that may contain one ore more clusters; it can also be empty, in the case no common items are detected.

First of all, we describe two basic operations that combine items belonging to two input clusters to get a new cluster.

We define two basic operations: **Cluster Intersection** and **Cluster Union**. They work on the *uri* of the items of two input clusters, assuming that *uri* is the document's unique identifier. The rationale of this assumption is the fact that the same document, retrieved by two different search services, may have different title and snippet, but maintains the same *uri*. Consider the intersection of two clusters c_1 and c_2, denoted as:

$$c' = ClusterIntersection(c_1, c_2).$$

The *irank* of $i' \in c'$, the cluster resulting from the intersection, is defined as the minimum *irank* value of i_1 and i_2.[2] In the case of cluster union, denoted as

$$c' = ClusterUnion(c_1, c_2),$$

the irank of i' is the maximum *irank* value of i_1 and i_2.[3] In both cluster intersection and union, the *title* and the *snippet* of the resulting items are obtained by selecting either $i_1.title$ or $i_2.title$, and either $i_1.snippet$ or $i_2.snippet$, respectively.

In particular, to obtain the *title* and the *snippet* of the items belonging to the clusters of the resulting groups we select as resulting *title* and *snippet*, those belonging to the document having the smallest (in the case of Cluster Intersection) or the greatest (in the case of Cluster Union) value of irank, without making any changes. The rationale of this choice is the fact that in the aggregation based on the intersection (union), we want to represent the document by its worst (best) representative, in accordance with the modelling of the AND and the OR within fuzzy set theory.

2.2.1 Group Operators

The first group operators we describe are not properly combination operators: they are the **Group Selection** and the **Group Deletion**. The *Group Selection* operator allows to select the clusters in a group. In the resulting group, the selected clusters maintain the original order.
Similarly, the *Group Deletion* operator allows the user to delete clusters. Like for the Cluster Selection operator, the original order is maintained in the resulting group.

The following operators combine and generate groups.

Group Intersection. Group Intersection is defined to support the straightforward wish of users to intersect clusters in two groups, to find more specific clusters. The assumption is that the more search services (or the more distinct queries) retrieve the same document, the more the document content is worth analyzing.

Definition 5: The *Group Intersection* operator generates a new group composed of all the combination of clusters in the original groups having a not empty intersection.
In particular, given g_1 and g_2 the groups of cluster to intersect, the resulting group g' is composed of all the clusters c' such that: $c' = ClusterIntersect(c_1, c_2)$ with $|c'| \neq 0$. □

Group Join A key operator of the language, closely related to the previous one, is the *Group Join*. It lets the user expand the original clusters in a group with clusters, possibly belonging to another group, that share one or more documents. The group Join operator can be used to explicit indirect correlations between the topics represented by the clusters in the two input groups. The basic idea underlying its definition is that if two clusters have a non empty intersection (i.e. have some common items), this means that the texts of their items are related with both topics represented by the clusters. This may hint the existence of an implicit relationship between the topics of the two clusters.
By merging the two overlapping clusters into a single one, the more general topic representing the whole content of the new cluster can be revealed, which subsumes, as more specific topics, those of the original clusters.

[2]This definition is consistent with the definition of the intersection operation between fuzzy sets [15].
[3]This is also consistent with the definition of union of fuzzy sets.

Definition 6: The *Group Join* operator allows the user to obtain, from two or more input groups, a resulting group composed by the union of all those pairs of original clusters that present a not empty intersection.
In particular, given g_1 and g_2 the input groups, for each pair of clusters $c_1 \in g_1$ and $c_2 \in g_2$, the cluster

$$\text{c'} = ClusterUnion(c_1, c_2) \in g',$$

if and only if $ClusterIntersection(c_1, c_2) \neq \emptyset$, with g' the resulting group. □

Group Refinement The *Group Refinement* operator is aimed at refining clusters in a group, based on clusters in another group. While the group join operator generates a cluster representing a more general topic than the topics in both the original clusters, the *refinement* operator can be regarded as generating clusters specializing the topics of the clusters in the first group on the basis of the topics of any cluster in the second group. The idea underlying this operator is that we want to collect, in a unique cluster, the items (that are considered by the user as more interesting) which belong to both a cluster c_1 of the first group g_1 and any of the second group g_2. This way, by eliminating some items from c_1, we generate a cluster representing a more specific topic w.r.t. c_1, but not necessarily more specific w.r.t. the clusters of the second group.

Definition 7: The *Group Refinement* operator allows the user to keep, from the original group g_1, only the clusters c_i containing documents presents in at least one of the clusters c_j of the most interesting group g_2.
In particular, given g_1 (group of clusters to refine) and g_2 (interesting group), and being c_1 a cluster such that $c_1 \in g_1$, for each cluster $c_j \in g_2$ we compute the cluster union of the intersections $\overline{c}_j$, $\overline{c}_j = ClusterIntersection(c_1, c_j)$.
If the union c' of $\overline{c}_j$ is not empty, then $c' \in g'$. □

The operators so far introduced constitute the core of our proposal; the others are sketched hereafter.

Group Union. The *Group Union* operator unites together two groups. It generates the resulting group g' in such a way it contains all clusters in the input groups g_1 and g_2.
Group Coalescing. Complex processing of retrieved documents may need to be performed by fusing all clusters in a group into one global cluster. The **Group Coalescing** operator generates a resulting group g' in such a way that g' contains only one cluster, obtained by uniting together all clusters in the input group g.
Reclustering. After complex transformations, it might be necessary to reapply the clustering method to a group. In fact, reclustering documents in a group may let new and unexpected semantic information emerge.
The *Reclustering* operator coalesces all clusters in the input group g and generates a new group g' in such a way that it contains all the clusters obtained by clustering all items.
The *Closure Property of Group Operators* holds: operators are defined on groups and generate groups [5].

3. THE MOBILE SYSTEM MATRIOSHKA

The interaction framework introduced in the previous section has been implemented in the mobile version of the prototypal system *Matrioshka*.

It is constituted by three main parts: the *client side* components handle the user interaction; the *server side* component interfaces the search engines and executes the clustering and the manipulation operations specified by the user; finally, the *Communication Layer* dispatches the messages between client and server. Specifically, the client provides a query editor for the user, the server either executes the queries and builds the groups of clusters or executes the operations on previously generated groups of clusters. Let us describe the functionality of each architectural component.

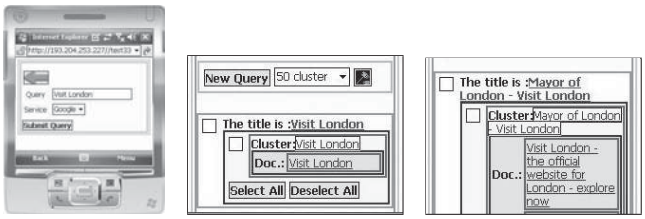


Figure 1: *Mobile Matrioshka*: the *interrogation panel* (left), Groups generated by the Group Intersection (center) and Group Join (right) operators.

On the **client side** the *Matrioshka* User Interface collects users requests, displays the results of queries and/or the application of manipulation operations. The Client-side components are *thin clients* compliant, and communicate with the server-side by exchanging XML messages. Specifically, the component for mobile devices (called *Mobile Matrioshka*), is a Javascript application based on the AJAX (Asynchronous JavaScript and XML) web development technique.

The **Server Side** exposes a web service interface, based on XML messages: it receives requests to perform queries on search services, or to apply the operators; it replies with groups of clusters. All the data received from the search engines, and those resulting from the operations, are stored in an XML native database; this way, the entire process is stored and can be accessed to carry on the exploratory task. The server side is entirely implemented in the Java Language. The interaction with search services usually exploits web service APIs provided by the search engines, otherwise the standard HTTP interaction model is exploited.

Document clustering is performed on the indexes extracted from the titles and snippets of retrieved documents (generated by using *Lucene* functions): the *Lingo* multilingual algorithm, provided by the *Carrot2* libraries is used.

The interpreter of the combination operators has been implemented from scratch.

The **Communication Layer** is a pool of JSP scripts, executed on top of the *Tomcat* web server. It carries out the client/server communication through XML format messages, according with AJAX web development techniques, and by the support of the Tomcat Java servlet container.

When the user logs into the system, a specific instance of the database is created, in which the entire exploratory process performed by the user will be stored. When logged-in, the user has the possibility to submit queries to the chosen search engine (as shown in the left-hand side of Figure 1).

In order to organize a trip to visit London, let us submit the query `"visit London"` to the search engines *Google*, *Yahoo!* and *MSN search*. Groups g_1, g_2, g_3 in Figure 2 are the resulting groups clusters; the three groups being generated by the same query `"Visit London"` have the same label.

Terminated the inspection of clusters in the groups, we can interactively ask for executing some operators, in an attempt of obtaining clusters with labels that more closely

g_1"Visit London"
cl.1: Visit London
cl.2: When to visit London
cl.3: Destination marketing
cl.4: London tourist information
cl.5: Visit London services
cl.6: The Royal Parks
cl.7: London Theater Guides

g_2"Visit London"
cl.1: Visit London
cl.2: Visit London-official web site
cl.3: Attractions in London
cl.4: London City Guide 2008
cl.5: Family-Visit London
cl.6: Visit London Organizers
cl.7: London Travel Maps
cl.8: Business-Visit London

g_3"Visit London"
cl.1: Travel - Visit London
cl.2: Visit London Organizers
cl.3: Special Offers - Visit London
cl.4: London Accommodation Guide
cl.5: Visit London Corporate
cl.6: London Maps - Visit London
cl.8: Places to go - Visit London

g_4"Visit London"
cl.1: Visit London
cl.2: Visit London-official website
cl.3: Visit London-official website

g_5"Mayor of London"
cl.1: Visit London
cl.2: London Accommodation Guide
cl.3: Mayor of London

Figure 2: Resp., resulting groups from the query *Visit London* submitted to Google (group g_1), Yahoo! (group g_2), and MSN live search engines (group g_3), *Group Intersection* and *Group Join* of groups g_1, g_2. .

meet our needs. At first, we ask to intersect the three groups to retrieve the most reliable documents. By observing clusters in the resulting group g_4, we then decide to request a join of the three original groups g_1 g_2 and g_3, in order to expand the contents obtained by the intersection (see the screen shots in Figure 1). A new group g_5 is generated with more populous clusters: these clusters are the union of the original clusters that share some common document. We can see that the obtained clusters are identified by labels which hints the presence of new correlated contents w.r.t. the labels of the clusters obtained by the intersections of the same groups (see groups g_4 vs group g_5 in Figure 2).

4. CONCLUSIONS

In this paper, we described a novel interaction framework for web searches implemented by the prototypal mobile version of the system *Matrioshka*.

The features that make this framework particularly suitable for mobile searches are several: first, it presents clustered results of the searches so as to better render them on the small screen of mobile devices; it makes available ranking and combination operators defined for clusters manipulation which allow easily exploring the retrieved results, thus alleviating network overloading caused by the submission of repeated refined queries to search engines. The large number of documents retrieved by such engines constitute a serious obstacle for users of mobile devices, who generally engages long trial and error query reformulation phases to retrieve relevant results in first few positions.

The operator provided by the interaction framework are the basis for complex exploratory tasks; users can issue operations through the mobile interface, but certainly they must be skilled users; certainly, generic users are in troubles. Currently we are performing an evaluation study to understand the effectiveness for end users, in order to define novel, more user friendly interaction paradigms on the client side, more suitable for generic users.

5. REFERENCES

[1] Ask.com clustering. www.ask.com/reference/dictionary/49514/cluster.

[2] The GRASS system. http://credino.dimi.uniud.it/.

[3] G. Bordogna, A. Campi, G. Psaila, and S. Ronchi. An interaction framework for mobile web searches results. *MOMM-2008*.

[4] G. Bordogna, A. Campi, G. Psaila, and S. Ronchi. A language for manipulating clustered web documents results. *CIKM-2008*.

[5] G. Bordogna, A. Campi, G. Psaila, and S. Ronchi. A flexibkle language for exploring clusterd search results. *Scalable Fuzzy Algs. for Data Mgmt. and Anal.*, 2009.

[6] G. Buchnan, M. Jones, and G. Marsden. Exploring small screen digital library access with the greenstone digital library. *Eur. Conf. on Digital Libraries*, 2003.

[7] O. Buyukkokten and et al. Efficient web browsing on handheld devices using page and form summarization. *ACM Trans. on Inf. Systems*, 20(1):82–115, 1999.

[8] H. Chen and S. Dumais. Bringing order to the web: Automatically categorizing search results. *SIGCHI Conf. on Human factors in computing systems*, 2000.

[9] T. Coates and al. Uris, urls, and urns: Clarifications and recommendations 1.0. *W3C Tech. report*, 2001.

[10] F. Crestani, M. Dunlop, M. Jones, S. Jones, and S. Mizzaro (eds.). Special issue on interactive mobile inf. access. *J. of Personal and Ubiquitous Comp.*, 2006.

[11] M. A. Hearst and J. O. Pederson. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. *Conf. on Research and Devel. in Inf. Retrieval*, 1996.

[12] A. V. Leouski and W. B. Croft. An evaluation of techniques for clustering search results. *Tech. Rep. of the Dept. of Computer Science f University of Massachusetts at Amherst*, IR-76:122–133, 1996.

[13] M. Noirhomme-Fraiture and al. Data visualizations on small and very small screens. *Conf. on Applied Stocastics Models and Data Analysis*, 2005.

[14] S. Osinski. An algorithm for clustering of web search results. Master's thesis, 2009, Poznan' Univ. of Tech.

[15] L. Zadeh. Fuzzy sets. *Information and control*, 1965.

[16] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. *WWW 1999*.

GrOnto: a GRanular ONTOlogy for Diversifying Search Results

Silvia Calegari
Gabriella Pasi
University of Milano-Bicocca
V.le Sarca 336/14, 20126
Milano, Italy
{calegari,pasi}@disco.unimib.it

ABSTRACT

Results diversification is an approach used in literature to cover the possible interpretations of the results produced by query evaluation. For diversifying search results we propose the GrOnto model. This model is based on a normalized granular view of an ontology: GrOnto allows to associate each result with the suited topical granules in order to categorize it based on the granular information.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval- *information filtering, search process*

1. INTRODUCTION

In last years, Web search engines have become the de-facto access point to the information available on the Internet. Usually people specify their information needs by writing queries with a limited number of terms (usually $2-3$ terms per query). However, short queries are very difficult to disambiguate: in fact a term may have several interpretations. One of the problems related to term disambiguation is how to diversify results produced as an answer to an ambiguous query. An interesting research topic that in recent years has attracted several researchers is results diversification. The focus is on how to produce a set of diversified results that cover the different possible interpretations of the query. The importance of result diversification has been recognized as a very important topic in Information Retrieval; the basic idea is that "*the relevance of a set of documents depends not only on the individual relevance of its members, but also on how they relate to one another*"[3]. The key aspect is that the relevance of a document has to consider also the semantics expressed by the terms it contains."*The focus is on how to diversify search results making explicit use of knowledge about the topics the query or the documents may refer to*" [1].

In a recent research work, a taxonomy of information is used to model the user's request [1]. The idea is to assign both query and documents to one or more categories of the taxonomy. The taxonomy adopted is the one provided by the ODP [1] ontology. Furthermore, it is assumed that *usage statistics have been collected on the distribution of user intents over the categories* ([6]). The aim of this approach is to minimize the risk of user dissatisfaction by computing a *quality value* for each document retrieved in response to a query as a combination of relevance and diversity.

In this paper a method for diversifying the results produced in response to a query is proposed. We do not use a statistical approach in order to diversify the results, but our method makes use of a semantic support offered by a granular view of an ontology [2] to the aim of producing a granular taxonomy of the results. By this method the information is classified at different topical levels (from a general topic to a specific topic).
In a granular ontology the concepts and instances are classified into granules. A granule is a chunk of knowledge made of different objects "*drawn together by indistinguishability, similarity, proximity or functionality*"[12]. A level is just the collection of granules of similar nature, and a granular information is a pyramidal information structure with different levels of clarifications.
The paper is organized as follows. In Section 2 an overview of the use of ontologies in Information Retrieval is presented. In Section 3 the definition of a normalized granular view of an ontology is reported. The approach proposed in this work, named GrOnto, for diversifying search results is defined in Section 4. At the end, in Section 5 some conclusions and future works are stated.

2. THE USE OF ONTOLOGIES IN INFORMATION RETRIEVAL

In the last decades ontologies have been used in different areas of research in Computer Science, among which Information Retrieval where they have been involved into several applications to different aims. For example, ontologies have been used: in distributed environments, for reranking the results to better satisfy the user's needs, to provide conceptual indexing and to disambiguate user's query. In distributed environment, significant works are SemreX [7] and Semantic Link Network (SLN)[13]. SemreX is a recent project that implements a multi-layer overlay network to map semantically correlated documents to clustered groups of neighbors. This semantic mapping is obtained by considering the ACM Topic Ontology. In SLN, an ontology has

[1]ODP: Open Directory Project, (http://dmoz.org)

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

been built as a self-organized semantic data model by defining semantic nodes, semantic links among nodes, and a set of relational reasoning rules; where each node identifies a resource.
In order to re-rank the results obtained after a search on the Web, generally, a user's profile is used. In the literature different strategies have been defined in order to build a user's profile by adopting the semantic support of an ontology. For example in [4] a user profile is built by considering past queries, and it is represented as a weighted graph by extracting the related terms from the ODP ontology.
In the conceptual indexing field of research, WordNet[2] synsets are used as terms for the representation of the documents. The concept detection phase consists in extracting concepts from documents that correspond to synsets in WordNet. In [8] the authors proposed some procedures to identify the correct sense of a word.
In this paper we are interested in the last field of research where the problem of disambiguation of the query is taken into account. Short queries are very difficult to disambiguate. Two main problems may arise: word synonymy (i.e., two words with the same meaning), and word polysemy (i.e., one word with multiple meanings). In the literature several strategies have been proposed in order to find a solution to this problem. Also ontologies have been involved in this field with the goal to provide a semantic support for reducing the ambiguity of the query. A way is to analyse the structure of the ontology to expand the terms written into the query with new meanings terms. The use of ontology reduces the possible (mis)interpretation of a query, but it needs to tune a query term to the right level in the hierarchy. Not only the IS-A relationship is used to discover the suited words [11], but also other important relationships such as, synonymy, meronymy and hypernyms are taken into account. For example in [9] the relationships considered are: *hyperonymy* and *synset*. For each term written in the query, a set of its synsets in WordNet is identified.

As reported in the Introduction of this paper, the results diversification is another strategy that can be adopted to solve the problem of ambiguous queries. We are interested in the situation where there is the necessity to individuate the different interpretations of a user's query. The focus is to produce a set of diversified results that cover at best these interpretations. One of pioneers works on diversification is that of Carbonell and Goldstein [3]. In their work, the diversification is obtained through the use of two similarity functions: one for measuring the similarity of the documents, and the other one for measuring the similarity between each document and a query. In more recent works a new approach has been explored to categorize both queries and documents by the use of a taxonomy [1, 14]. In these papers the taxonomy adopted is the one of the ODP ontology. The taxonomy is set by the IS-A relationship among categories; in fact in this context each concept of the ODP ontology represents a specific category.
In our paper we propose a method to diversify search results with the adoption of a new granular view of an ontology. Whereas in the previous works ([1, 14]) the taxonomy has been used only as a vocabulary for individuating the categories for queries and documents, now we consider an innovative ontology framework with a semantic expressiveness (i.e., instances and their properties) richer than the ODP ontology.

[2]http://wordnet.princeton.edu/

3. GRANULAR VIEW OF AN ONTOLOGY

This proposed method is based on the concept of a granular view (or granular perspective) of an ontology which has been defined in [2]. Given a domain ontology, the idea is to analyse the instances and their properties in order to discover new semantic associations among them. These semantic associations can be defined with the application of a rough methodology. The objective is to re-organize the ontology in a new taxonomy obtained after the analysis of the properties values assigned to the instances.
The rough structure used is known as Information Table [10]. For a domain ontology, an Information Table is induced as the structure:

$$\langle I, P, Val(I), F \rangle$$

where I is the set of the instances, P is the set of the properties, $Val(I)$ is the set of all the values assumed by the properties P, and F is the function that assigns to a pair (i, p) the value assumed by the instance $i \in I$ on the property $p \in P$. Thus, we can say that two instances are similar if they have the same values only for some properties. Formally, let $D \subseteq P$, then given two instances $i_1, i_2 \in I$, i_1 is similar to i_2 with respect to D and ϵ, with $\epsilon \in [0, 1]$, iff

$$\frac{|\{d_j \in D : F(i_1, d_j) = F(i_2, d_j)\}|}{|D|} \geq \epsilon \qquad (1)$$

This relation says that two instances are similar if they have at least $\epsilon|D|$ properties with the same value. For example, if we consider a Wine Ontology then a possible set of properties is $P := \{Location, Color, Sugar, Flavor, Body\}$. D is a subset of P defined as $D := \{Sugar, Flavor, Body\}$. In this case two instances belong to the same granule if they have at least $|(D-1)|$ properties with the same value, i.e. $\epsilon := \frac{|(D-1)|}{|D|} := \frac{2}{3}$. For example, *Longridge Merlot* and *Marietta Zinfandel* belong to the same granule by having two properties with the same value, i.e. $(flavor == moderate)$ and $(sugar == dry)$.

In [2] the instances are classified into granules at a different level of clarification. A key aspect is how to choose the granular levels from the non-granular ontology. The idea is to cluster the instances into granules by considering their similarity, i.e. by analysing the values of their properties (see Equation 1).

The granular view of an ontology is defined by following 3 steps. In order to clarify the construction of the new ontology, we refer to a very simple example. In this example, let us consider a small Wine Ontology which has 4 instances, and the set P of properties previously defined.
First step: definition of the tabular version of the ontology. In this table the rows are the instances and the columns are all the properties defined in the ontology. The selected instances and properties are the ones defined only by the IS-A relationships of the ontology domain. Table 1 reports the instances and the properties with their values of the small Wine Ontology analysed in this work.
Second step: It consists in the definition of the granular levels. As previously stated the granular levels have been chosen by analysing the properties values of the instances.

Table 1: A tabular version for the small Wine Ontology

Instances	Color	Sugar	Flavor	Body	Location
`Longridge Merlot`	Red	Dry	Moderate	Light	$Undefined$
`Marietta Zinfandel`	Red	Dry	Moderate	Medium	$Undefined$
`Lane Tanner Pinot Noir`	Red	Dry	Delicate	Light	$Undefined$
`Chateau-D-Ychem`	$Undefined$	$Undefined$	$Undefined$	$Undefined$	Bordeaux region

The tabular representation is used as support for this step. Thus, from the set of properties P two disjoint sets of granules are induced: $D_1 := \{Color, Flavor, Body, Sugar\}$ and $D_2 := \{Location\}$. Only $Location$ belongs to the first level with the instance $Chateau-D-Ychem$ at the second granular level. Whereas for D_1, the choice of the first granular level has to be made among the properties that belong to D_1. Also in this case we have to analyze the properties values assumed by the set of instances, and we can observe that the identification of the first granular level can be made arbitrarily between $Color$ and $Sugar$ since they assume the same values for all their instances. For this ontology, without loss of generality, we can consider $Color$ at the first granular level, and for the next level the similarity relation (i.e., Equation 1) to the D_1 set (without the property $Color$) can be applied. In this illustrative example $\epsilon := \frac{2}{3}$, that is, two instances belong to the same granule if they have at least two out of three properties with the same value. Figure 1 depicts the granular classification obtained where the circles are the properties values and the squares are the instances.

The **third step** is to solve the problem of redundancy of

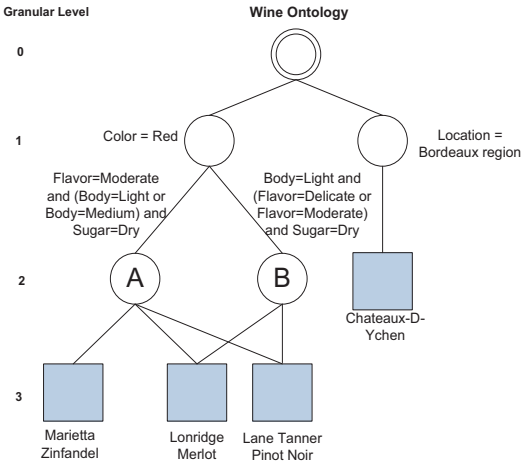


Figure 1: A granular view of a small Wine Ontology after the application of the rough methodology.

the information. Let us consider two granules G_i and G_j at the same granular level, we have that G_i is redundant with respect to G_j iff $G_j \supseteq G_i$. In [2] a normalisation process has been defined in order to obtain a normal form of the granular perspective. For example, if we examine the same example of Figure 1, we can observe that G_A and G_B belong to the same granular level, and that $G_A \supseteq G_B$. Indeed, the instances *Lonridge Merlot* and *Lane Tanner Pinot Noir* are completely included into G_B but they belong to G_A. In this normalisation process the granular subclass G_B inherits all the common instances from the granular superclass G_A (see Figure 2).

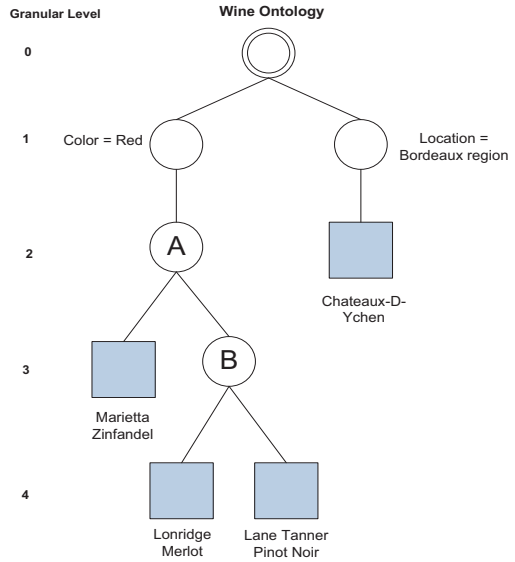


Figure 2: The granular view of the small Wine Ontology after the application of the normalisation process.

4. THE PROPOSED MODEL

When using a search engine a user formulates a query in order to retrieve the documents relevant to her/his information needs. In most cases the user writes short queries that are difficult to disambiguate. In fact, in several user's queries a query term could be interpreted with different meanings. We propose a solution to diversify search results that aims to increase the effectiveness of the system by reducing the ambiguity in the interpretation of results. As proposed in [1] we adopt a taxonomy of information where both queries and results may belong to more than one category. In particular we use the taxonomy corresponding to a normalized granular view of an ontology (see Section 3). The idea is to associate each result with the suited topical granules.

Generally, in search engines the evaluation of a user's query produces an ordered list of results. For diversifying search results the GrOnto model (see Figure 3) takes in input a ranked *list of results*, and the *granular ontology* to categorize each result. In other words, the normalized granular view of the ontology is used to apply a filtering on the search results. As reported in Section 1, in a *granular* ontology the granules are organized at different levels of clarifications. Thus the categorization of each result is performed by locating in the ontology the right granules with which it may be associated. Figure 4 shows the general structure of the approach where the list of results (left-hand side of Figure 4) is re-organized by the filtering strategy (right-hand side of Figure 4) based on the *granular* ontology structure. By applying the catego-

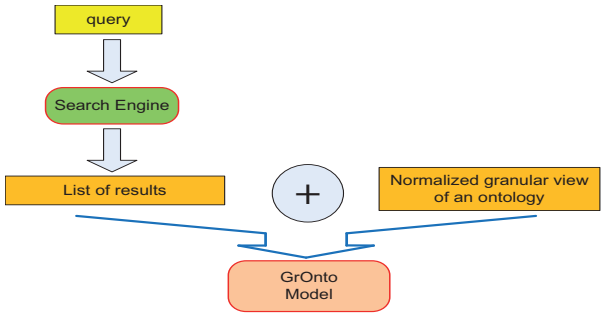


Figure 3: A simple schema of the GrOnto model.

rization process (explained here below), we obtain a representation of the results which reflects the classification into topics corresponding to the granular levels of the adopted ontology. Each retrieved document is associated with one

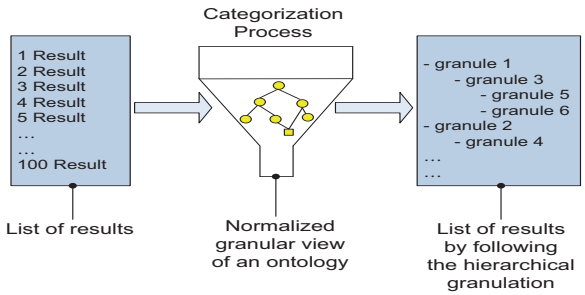


Figure 4: A Web search after the application of the GrOnto model.

or more granules of the ontology by a procedure explained here below.

As an example, let us consider the same vocabulary and structure of the Wine Ontology described in Section 3. The related set of concepts is $O := \{Red, Bordeaux\,region, Chateau-D-Ychen, MariettaZinfandel, Lonridge\,Merlot, Lane\,Tanner\,Pinot\,Noir\}$. During a search session a user is interested in finding, for instance, information about *red wines* and she/he writes the following short query *q:="red wines in France"*, and a list of results is displayed. The association of each result with granules of the *granular* ontology is obtained in two steps. Here below the process undertaken to categorize a search result is explained. We present these two steps in order to categorize the first result, obviously the same procedure is applied to the other search results.

Step 1: "Formal representation of each result". In order to formally represent the content of a result R_i proposed in response to a query, we assume that results are described by *Title* and *Snippet*. The $i-th$ result R_i is then associated with a set of terms, Res_i, extracted from the textual information, i.e. $Res_i := Title_i \cup Snippet_i$ where $Title_i$ and $Snippet_i$ are sets of terms included into the vocabulary of the *granular* ontology.

Thus, by analysing the first result R_1, we have: *Title:="Wines of France-A guide to French wines"* and *Snippet:="Discover the wines of France, their varieties, history and regions;... Lane Tanner Pinot Noir is a very famous red wine produced in..."*. From these two short texts, by considering the set O, we obtain that $Res_1 := \{Lane\,Tanner\,Pinot\,Noir, Red\}$, i.e. $Title_1 := \emptyset := Title \cap O$ and $Snippet_1 := \{Lane\,Tanner\,Pinot\,Noir, Red\} := Snippet \cap O$.

Step 2: "Association of each result R_i with granules of the granular tree". The output of Step 1 is a set of terms of the vocabulary O, named Res_i, for each retrieved document R_i. An element of Res_i is a granule of the ontology, and to this granule we can associate the $i-th$ result. Thus, for each granule the following structure: $< Results_j, card_{TOT_j} >$ is defined, where $Results_j$ is the set of the search result associated with the $j-th$ granule, i.e. $Results_j := \{R_i | granule_j \in Res_i\}$, and $card_{TOT_j}$ is the cardinality of all the results associated with the $j-th$ granule. This means that $card_{TOT_j} := |Results_j \cup (\bigcup_{child=0}^{n} Results_{child})|$ i.e., the cardinality of all the results individuated with the granule $j-th$ and the cardinality of the results associated with all its n sub-granules (children nodes).

By considering the same example of Step 1, we have that the first result R_1 has been formally represented as $Res_1 := \{Lane\,Tanner\,Pinot\,Noir, Red\}$ so that, the selected granules are *Lane Tanner Pinot Noir* and *Red*. Figure 5 depicts the situation after the application of Step 2 where the structure assigned with $granule_1$ is $< Results_1 := \{R_1\}, 1 >$, whereas for $granule_8$ is $< Results_8 := \{R_1\}, 1 >$. Thus, we have that the first result R_1 has been categorized with two topics (granules) at a different level of clarification.

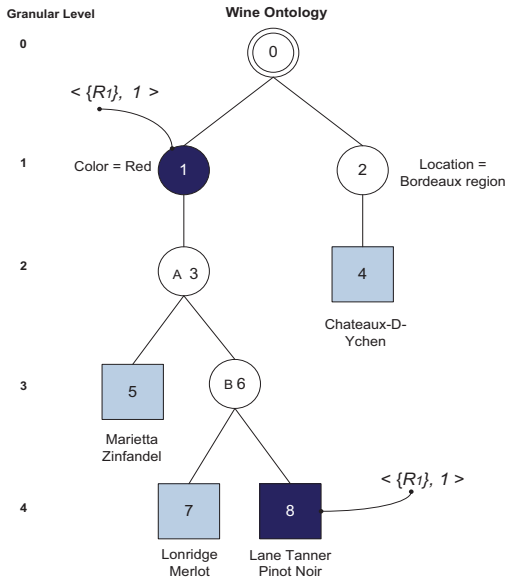


Figure 5: Example of the structure assigned to each granule identified with a result.

GrOnto on the Web.

Figure 6 depicts a prototype interface for the GrOntoS system. We have taken inspiration from *Clusty*[3] where the web-page structure is split into three parts: 1) a text area where the user can formulate her/his request by using the Yahoo! Search engine, 2) a profile used to visualize the portion of the normalized granular view of the ontology involved from the specific query, and 3) a web-page area devoted to the visualization of the results. In particular only the results categorized with a granule of the ontology are displayed

[3](http://clusty.com/)

one by one. Figure 6 reports a simple example where the small Wine Ontology of Section 3 is used to classify *ALL* the results obtained, for example, after the evaluation of the *q:=red wines in France*. A user can use the portion of the granular ontology in order to navigate the results by considering the categorization provided by the levels granular. In fact by clicking on an item of the portion of the granular ontology, all its results will be visualised. Furthermore, each item is enriched with the cardinality of the results associated with its topic, in this way the user is directed towards the category more numerous.

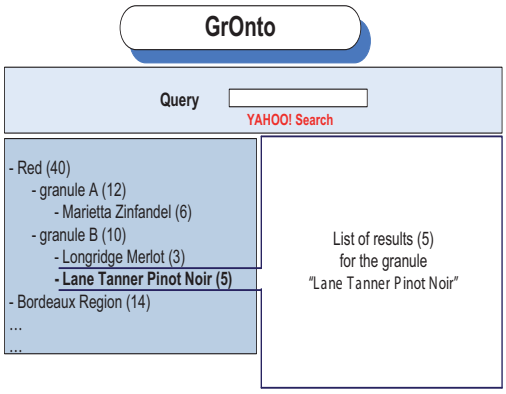


Figure 6: The interface model of the GrOnto model.

5. CONCLUSIONS

In this paper we have studied the problem of diversification of search results to disambiguate the user's query in a given domain of knowledge represented by a granular ontology. We have proposed a model, named Gronto, based on a semantic support for associating search result with one or more categories. A normalized granular view of an ontology is the semantic framework adopted in order to cover all the possibles meanings of a result. Generally, after the evaluation of a user's query an ordered list of results is obtained. GrOnto takes in input this list and the granular ontology, and thanks to the adoption of a filtering strategy a taxonomic organization of the results is achieved.

We are implementing the GrOnto model through a simple web service by adopting the representational state transfer (REST) paradigm [5].

The prosecution of this research activity will address the problem of applying the GrOnto approach to personalized ontologies, where the user interests will be represented by means of a granular ontology. To this aim we are also investigating the problem of defining personalized granular ontologies.

6. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14, New York, NY, USA, 2009. ACM.

[2] S. Calegari and D. Ciucci. Granular computing applied to ontologies. *International Journal of Approximate Reasoning*, 2009. In printing.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *In Research and Development in Information Retrieval*, pages 335–336, 1998.

[4] M. Daoud, L. Tamine-Lechani, M. Boughanem, and B. Chebaro. A session based personalized search using an ontological user profile. In S. Y. Shin and S. Ossowski, editors, *SAC*, pages 1732–1736. ACM, 2009.

[5] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 407–416, New York, NY, USA, 2000. ACM.

[6] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. Using the wisdom of the crowds for keyword generation. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 61–70, New York, NY, USA, 2008. ACM.

[7] H. Jin and H. Chen. Semrex: Efficient search in a semantic overlay for literature retrieval. *Future Generation Computer System*, 24(6):475–488, 2008.

[8] R. Mihalcea and D. I. Moldovan. Semantic indexing using wordnet senses. In *In Proceedings of ACL Workshop on IR & NLP*, pages 35–45, 2000.

[9] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *Workshop on Adaptive Text Extraction and Mining, (Cavtat Dubrovnik, Croatia, Sept 23)*, 2003.

[10] Z. Pawlak. Information systems - theoretical foundations. *Information Systems*, 6:205–218, 1981.

[11] E. M. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[12] L. Zadeh. Is there a need for fuzzy logic? *Information Sciences*, 178:2751–2779, 2008.

[13] H. Zhuge. Communities and emerging semantics in semantic link network: Discovery and learning. *IEEE Transactions on Knowledge and Data Engineering*, 21(6):785–799, 2009.

[14] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.

An IR-based approach to Tag Recommendation

Cataldo Musto
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
cataldomusto@di.uniba.it

Fedelucio Narducci
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
narducci@di.uniba.it

Marco De Gemmis
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
degemmis@di.uniba.it

Pasquale Lops
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
lops@di.uniba.it

Giovanni Semeraro
Dept. of Computer Science
University of Bari 'Aldo Moro'
Italy
semeraro@di.uniba.it

ABSTRACT

Thanks to the continuous growth of collaborative platforms like YouTube, Flickr and Delicious, we are recently witnessing to a rapid evolution of web dynamics towards a more 'social' vision, called Web 2.0. In this context collaborative tagging systems are rapidly emerging as one of the most promising tools. However, as tags are handled in a simply syntactical way, collaborative tagging systems suffer of typical Information Retrieval (IR) problems like polysemy and synonymy: so, in order to reduce the impact of these drawbacks and to aid at the same time the so-called tag convergence, systems that assist the user in the task of tagging are required.

In this paper we present a system, called STaR, that implements an IR-based approach for tag recommendation. Our approach, mainly based on the exploitation of a state-of-the-art IR-model called BM25, relies on two assumptions: firstly, if two or more resources share some common patterns (e.g. the same features in the textual description), we can exploit this information supposing that they could be annotated with similar tags. Furthermore, since each user has a typical manner to label resources, a tag recommender might exploit this information to weigh more the tags she already used to annotate similar resources. We also present an experimental evaluation, carried out using a large dataset gathered from Bibsonomy.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing: Indexing methods; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval: Information filtering

General Terms

Algorithms, Experimentation

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

Keywords

Recommender Systems, Web 2.0, Collaborative Tagging Systems, Folksonomies

1. INTRODUCTION

We are assisting to a transformation of the Web towards a more user-centric vision called Web 2.0. By using Web 2.0 applications users are able to publish auto-produced contents such as photos, videos, political opinions, reviews, hence they are identified as *Web prosumers*: *pro*ducers + con*sumers* of knowledge. Recently the research community has thoroughly analyzed the dynamics of *tagging*, which is the act of annotating resources with free labels, called *tags*. These systems provide heterogeneous contents (photos, videos, musical habits, etc.), but they all share a common core: they let users to post new resources and to annotate them with tags. Besides the simple act of annotation, the tagging of resources has also a key social aspect; the connection between users, resources and tags generates a tripartite graph that can be easily exploited to analyze the dynamics of collaborative tagging systems. Since folksonomies do not rely on a predefined lexicon or hierarchy they have the main advantage to be fully free, but at the same time they generate a very noisy tag space, really hard to exploit for retrieval or recommendation tasks without performing any form of processing.

This problem is a hindrance to completely exploit the expressive power of folksonomies, so in the last years many tools have been developed to assist the user in the task of tagging and to aid at the same time the tag convergence: we refer to them as tag recommenders.

This paper presents STaR, a tag recommender system implementing an IR-based approach that relies on a state-of-the-art IR model called BM25. In this work, already presented [5],within the ECML-PKDD 2009 Discovery Challenge[1], we tried to point out two concepts:

- resources with similar content should be annotated with similar tags;
- a tag recommender needs to take into account the previous tagging activity of users, increasing the weight of the tags already used to annotate similar resources.

[1]http://www.kde.cs.uni-kassel.de/ws/dc09

The paper is organized as follows. Section 2 analyzes related work. Section 3 explains the architecture of the system and how the recommendation approach is implemented. The experimental evaluation carried out is described in Section 4, while conclusions and future work are drawn in the last section.

2. RELATED WORK

Usually the works in the tag recommendation area are broadly divided into three classes: *content-based*, *collaborative* and *graph-based* approaches.

In the content-based approach, exploiting some Information Retrieval-related techniques, a system is able to extract relevant unigrams or bigrams from the text. Brooks et. al [2], for example, develop a tag recommender system that exploits TF/IDF scoring in order to automatically suggests tags for a blog post.

AutoTag [4] is one of the most important systems implementing the collaborative approach for tag recommendation. It presents some analogies with collaborative filtering methods. As in the collaborative recommender systems the recommendations are generated based on the ratings provided by similar users (called neighbors), in AutoTag the system suggests tags based on the other tags associated with similar posts.

The problem of tag recommendation through graph-based approaches has been firstly addressed by Jäschke et al. in [3]. The key idea behind their FolkRank algorithm is that a resource which is tagged by important tags from important users becomes important itself. Furthermore, Schmitz et al. [7] proposed association rule mining as a technique that might be useful in the tag recommendation process.

3. STAR: A SOCIAL TAG RECOMMENDER SYSTEM

STaR (Social Tag Recommender) is a content-based tag recommender system, developed at the University of Bari. The inceptive idea behind STaR is to improve the model implemented in systems like TagAssist [8] or AutoTag [4].

Although we agree that similar resources usually share similar tags, in our opinion Mishne's approach presents two important drawbacks:

1. the tag re-ranking formula simply performs a sum of the occurrences of each tag among all the folksonomies, without considering the similarity with the resource to be tagged. In this way tags often used to annotate resources with a low similarity level could be ranked first;

2. the proposed model does not take into account the previous tagging activity performed by users. If two users bookmarked the same resource, they will receive the same suggestions since the folksonomies built from similar resources are the same.

We will try to overcome these drawbacks, by proposing an approach firstly based on the analysis of similar resources capable also of leveraging the tags already selected by the user during her previous tagging activity, by putting them on the top of the tag rank.

Figure 1 shows the general architecture of STaR.

3.1 Indexing of Resources

Given a collection of resources (*corpus*) with some textual metadata (such as the title of the resource, the authors, the description, etc.), STaR firstly invokes the *Indexer* module in order to perform a preprocessing step on these data by exploiting Apache Lucene[2]. Obviously, the kind of metadata to be indexed is strictly dependent on the nature of the resources. Let U be the set of users and N the cardinality of this set, the indexing procedure is repeated $N+1$ times: we build an index for each user (*Personal Index*) storing the information on the resources she previously tagged and an index for the whole community (*Social Index*) storing the information about all the tagged resources by merging the Personal Indexes.

3.2 Retrieval of Similar Resources

STaR can take into account users requests in order to produce personalized tag recommendations for each resource. First, every user has to provide some information about the resource to be tagged, such as the title of the Web page or its URL, in order to crawl the textual metadata associated on it. Next, if the system can identify the user since she has already posted other resources, it exploits data about her (language, the tags she uses more, the number of tags she usually uses to annotate resources, etc.) in order to refine the query to be submitted against both the *Social* and *Personal* indexes stored in Lucene.

In order to improve the performances of the Lucene Querying Engine we replaced the original Lucene Scoring function with an Okapi BM25 implementation[3]. BM25 is nowadays considered as one of the state-of-the art retrieval models by the IR community [6].

Let D be a corpus of documents, $d \in D$, BM25 returns the top-k resources with the highest similarity value given a resource r (tokenized as a set of terms $t_1 \dots t_m$), and is defined as follows:

$$sim(r,d) = \sum_{i=1}^{m} \frac{n^r_{t_i}}{k_1((1-b)+b*l)+n^r_{t_i}} * idf(t_i) \qquad (1)$$

where $n^r_{t_i}$ represents the occurrences of the term t_i in the document d, l is the ratio between the length of the resource and the average length of resources in the corpus. Finally, k_1 and b are two parameters typically set to 2.0 and 0.75 respectively, and $idf(t_i)$ represents the inverse document frequency of the term t_i defined as follows:

$$idf(t_i) = log\frac{N - df(t_i) + 0.5}{df(t_i) + 0.5} \qquad (2)$$

where N is the number of resources in the collection and $df(t_i)$ is the number of resources in which the term t_i occurs. Given a user u and a resource r, Lucene returns the resources whose similarity with r is greater or equal than a threshold β. To perform this task Lucene uses both the *PersonalIndex* of the user u and the *SocialIndex*.

For example, we suppose that the target resource is represented by Gazzetta.it, one of the most famous Italian sport newspaper. Lucene queries the *SocialIndex* and it could returns as the most similar resources an online newspaper (Corrieredellosport.it) and the official web site of an Italian

[2]http://lucene.apache.org

[3]http://nlp.uned.es/ jperezi/Lucene-BM25/

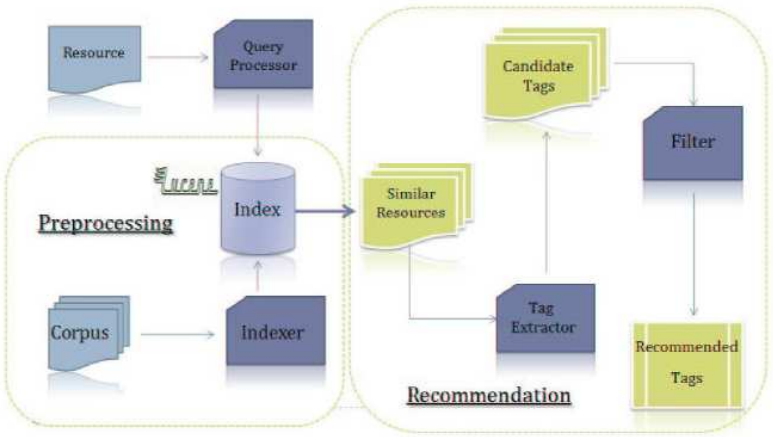


Figure 1: Architecture of STaR

Football Club (Inter.it). The *PersonalIndex*, instead, could return another online newspaper (Tuttosport.com).

3.3 Extraction of Candidate Tags

The role of the *Tag Extractor* is to produce as output the list of the so-called "candidate tags" (namely, the tags considered as 'relevant' by the tag recommender). In this step the system gets the most similar resources returned by the Apache Lucene engine and builds their folksonomies (namely, the tags they have been annotated with). Next, it produces the list of candidate tags by computing for each tag from the folksonomy a score obtained by weighting the similarity score returned by Lucene with the normalized occurrence of the tag. If the *Tag Extractor* also gets the list of the most similar resources from the user *PersonalIndex*, it will produce two partial folksonomies that are merged, assigning a weight to each folksonomy in order to boost the tags previously used by the user.

Figure 2 depicts the procedure performed by the *Tag Extractor*: in this case we have a set of 4 Social Tags (Newspaper, Online, Football and Inter) and 3 Personal Tags (Sport, Newspaper and Tuttosport). These sets are then merged, building the set of *Candidate Tags*. This set contains 6 tags since the tag *newspaper* appears both in social and personal tags. The system associates a score to each tag that indicates its effectiveness for the target resource. Besides, the scores for the Candidate Tags are weighted again according to SocialTagWeight (α) and PersonalTagWeight ($1-\alpha$) values (in the example, 0.3 and 0.7 respectively), in order to boost the tags already used by the user in the final tag rank. Indeed, we can point out that the social tag 'football' gets the same score of the personal tag 'tuttosport', although its original weight was twice.

3.4 Tag Recommendation

Finally, the last step of the recommendation process is performed by the *Filter*. It removes from the list of candidate tags those not matching specific conditions, such as a threshold for the relevance score computed by the Tag Extractor. Obviously, the value of the threshold and the maximum number of tags to be recommended are strictly dependent from the training data. In the example in Figure 2, setting a threshold $\gamma = 0.20$, the system would suggest the tags *sport* and *newspaper*.

4. EXPERIMENTAL EVALUATION

The goal of experimental session was to tune the system parameters in order to obtain the best effectiveness of the tag recommender. We exploited a large dataset gathered from Bibsonomy.

4.1 Description of the dataset

The dataset used for the experimental evaluation contains *263,004 bookmark* posts and *158,924 BibTeX* entries submitted by 3,617 different users. For each of the 235,328 different URLs and the 143,050 different BibTeX entries were also provided some textual metadata (such as the title of the resource, the description, the abstract and so on). We evaluated STaR by comparing the real tags (namely, the tags a user adopts to annotate an unseen resource) with the suggested ones. The accuracy was finally computed using classical IR metrics, such as Precision, Recall and F1-Measure.

4.2 Experimental Session

Firstly, we tried to evaluate the influence of different Lucene scoring functions on the performance of STaR. We randomly chose 10,000 resources from the dataset and we compared the results returned exploiting two different scoring functions (the Lucene original one and the BM25) in order to find the best one. We performed the same steps previously described, retrieving the most similar items using the two mentioned similarity functions and comparing the tags suggested by the system in both cases. Results are presented in Table 1. In general, there is a low improvement by adopting BM25 with respect to the Lucene original similarity function. We can note that BM25 improved the recall of bookmarks (+ 6,95%) and BibTeX entries (+1,46%).

Next, using the BM25 as scoring function, we tried to compare the predictive accuracy of STaR with different combinations of system parameters. Namely:

- the maximum number of similar documents retrieved by Lucene;
- the value of α for the *PersonalTagWeight* and *Social-*

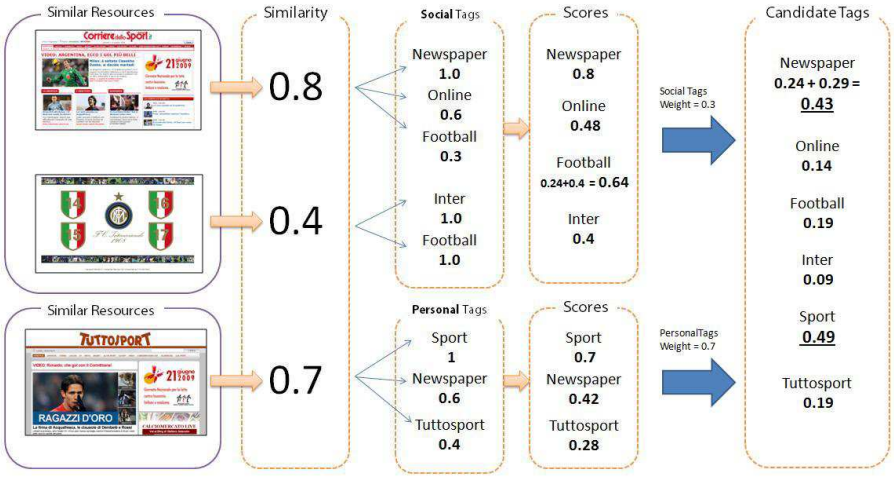


Figure 2: Description of the process performed by the Tag Extractor

Table 1: Results comparing the Lucene original scoring function with BM25

Scoring	Resource	Pr	Re	F1
Original	bookmark	25.26	29.67	27.29
Original	bibtex	14.06	21.45	16.99
BM25	bookmark	25.62	36.62	30.15
BM25	bibtex	13.72	22.91	17.16
Original	overall	16.43	23.58	19.37
BM25	overall	**16.45**	**26.46**	**20.29**

Table 2: Predictive accuracy of STaR over $50,000$ **bookmarks**

Approach	STW	PTW	Pr	Re	F1
Comm.-based	1.0	0.0	23.96	24.60	24.28
User-based	0.0	1.0	**32.12**	**28.72**	**30.33**
Hybrid	0.7	0.3	24.96	26.30	25.61
Hybrid	0.5	0.5	24.10	25.16	24.62
Hybrid	0.3	0.7	23.85	25.12	25.08
Baseline	-	-	35.58	10.42	16.11

TagWeight parameters;

- the threshold γ to establish whether a tag is relevant;
- which fields of the target resource use to compose the query.

Tuning the number of similar documents to retrieve from the *PersonalIndex* and *SocialIndex* is very important, since a value too high can introduce noise in the retrieval process, while a value too low can exclude documents containing relevant tags. By analyzing the results returned by some test queries, we decided to set this value between 5 and 10, depending on the training data.

Next, we tried to estimate the values for *PersonalTagWeight* (PTW) and the *SocialTagWeight* (STW). A higher weight for the Personal Tags means that in the recommendation process the systems will weigh more the tags previously used by the target user, while a higher value for the Social Tags will give more importance to the tags used by the community (namely, the whole folksonomy) on the target resource. These parameters are biased by the user practice: if tags often used by the user are very different from those used from the community, the PTW should be higher than STW. We performed an empirical study since it is difficult to define the user behavior at run time. We tested the system setting the parameters with several combinations of values:

i) PTW = 0.7 STW = 0.3;
ii) PTW = 0.5 STW = 0.5;
iii) PTW = 0.3 STW = 0.7.

Another parameter that can influence the system performance is the set of fields to use to compose the query. For each resource in the dataset there are many textual fields, such as title, abstract, description, extended description, etc. In this case we used as query the title of the webpage (for bookmarks) and the title of the publication (for BibTeX entries). The last parameter we need to tune is the threshold to deem a tag as relevant (γ).We performed some tests suggesting both 4 and 5 tags and we decided to recommend only 4 tags since the fifth was usually noisy. We also fixed the threshold value between 0.20 and 0.25. In order to carry out this experimental session we used the aforementioned dataset both as training and test set. We executed the test over $50,000$ bookmarks and $50,000$ BibTeXs. Results are presented in Table 2 and Table 3.

Analyzing the results, it emerges that the approach we called *user-based* outperformed the other ones. In this configuration we set PTW to 1.0 and STW to 0, so we suggest only the tags already used by the user in tagging similar resources. No query was submitted against the *SocialIndex*. The first remark we can make is that each user has her own mental model and her own vocabulary: she usually prefers to tag resources with labels she already used. Instead, getting tags from the *SocialIndex* only (as proved

Table 3: Predictive accuracy of STaR over 50,000 BibTeXs

Approach	STW	PTW	Pr	Re	F1
Comm.-based	1.0	0.0	34.44	35.89	35.15
User-based	0.0	1.0	**44.73**	**40.53**	**42.53**
Hybrid	0.7	0.3	32.31	38.57	35.16
Hybrid	0.5	0.5	32.36	37.55	34.76
Hybrid	0.3	0.7	35.47	39.68	37.46
Baseline	-	-	42.03	13.23	20.13

by the results of the community-based approach) often introduces some noise in the recommendation process. The hybrid approaches outperformed the community-based one, but their predictive accuracy is still worse when compared with the user-based approach. Finally, all the approaches outperformed the F1-measure of the baseline. We computed the baseline recommending for each resource only its most popular tags. Obviously, for resources never tagged we could not suggest anything. This analysis substantially confirms the results we obtained from other studies performed in the area of the tag-based recommendation [1].

5. CONCLUSIONS AND FUTURE WORK

Nowadays, collaborative tagging systems are powerful tools but they are affected from some drawbacks since the complete tag space is too noisy to be exploited for retrieval and filtering tasks. In this paper we presented STaR, a social tag recommender system. The idea behind our work was to discover similarity among resources exploiting a state-of-the-art IR-model called BM25. The experimental sessions showed that users tend to reuse their own tags to annotate similar resources, so this kind of recommendation model could benefit from the use of the user personal tags before extracting the social tags of the community (we called this approach user-based).

This approach has a main drawback, since it cannot suggest any tags when the set of similar items returned by Lucene is empty. We are planning to extend the system in order to extract significant keywords from the textual content associated to a resource (title, description, etc.) that has no similar items, maybe exploiting structured data or domain ontologies.

Furthermore, since tags usually suffer of typical Information Retrieval problem (polysemy, etc.) we will try to establish whether the integration of Word Sense Disambiguation algorithms or a semantic representation of documents could improve the performance of the recommender.

Anyhow, our approach resulted promising compared with already existing and state of the art approaches for tag recommendation. Indeed, our work classified in 6th position in the final results of the ECML-PKDD 2009 Discovery Challenge (id: 29723)[4]

6. REFERENCES

[1] P. Basile, M. de Gemmis, P. Lops, G. Semeraro, M. Bux, C. Musto, and F. Narducci. FIRSt: a Content-based Recommender System Integrating Tags for Cultural Heritage Personalization. In P. Nesi, K. Ng, and J. Delgado, editors, *Proceedings of the 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS 2008) - Workshop Panels and Industrial Applications, Florence, Italy*, Firenze University Press, pages 103–106, November 17-19, 2008.

[2] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM Press.

[3] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In Alexander Hinneburg, editor, *Workshop Proceedings of Lernen - Wissensentdeckung - Adaptivit?t (LWA 2007)*, pages 13–20, September 2007.

[4] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM.

[5] C. Musto, F. Narducci, M. de Gemmis, P. Lops, and G. Semeraro. STaR: a Social Tag Recommender System. In Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors, *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR Workshop Proceedings*, September 7 2009.

[6] S. E. Robertson, S. Walker, M. H. Beaulieu, A. Gull, and M. Lau. Okapi at trec. In *Text REtrieval Conference*, pages 21–30, 1992.

[7] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Őiberna, editors, *Data Science and Classification (Proc. IFCS 2006 Conference)*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin/Heidelberg, July 2006. Springer. Ljubljana.

[8] S. Sood, S. Owsley, K. Hammond, and L. Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[4]http://www.kde.cs.uni-kassel.de/ws/dc09/results

Context-Dependent Recommendations with Items Splitting

Linas Baltrunas
Free University of Bozen-Bolzano,
Piazza Università 1,
Bolzano, Italy
lbaltrunas@unibz.it

Francesco Ricci
Free University of Bozen-Bolzano,
Piazza Università 1,
Bolzano, Italy
fricci@unibz.it

ABSTRACT

Recommender systems are intelligent applications that help on-line users to tackle information overload by providing recommendations of relevant items. Collaborative Filtering (CF) is a recommendation technique that exploits users' explicit feedbacks on items to predict the relevance of items not evaluated yet. In classical CF users' ratings are not specifying in which contextual conditions the item was evaluated (e.g., the time when the item was rated or the goal of the consumption). But, in some domains the context could heavily influence the relevance of the item and this must be taken into account. This paper analyzes the behavior of a technique which deals with context by generating new items that are restricted to a contextual situation. The ratings' vectors of some items are split in two vectors containing the ratings collected in two alternative contextual conditions. Hence, each split generates two fictitious items that are used in the prediction algorithm instead of the original one. We evaluated this approach on semi-synthetic data sets measuring precision and recall while using a matrix-factorization algorithm for generating rating predictions. We compared our approach to the previously introduced reduction based method. We show that item splitting can improve system accuracy. Moreover, item splitting leads to a better recall than the reduction based approach.

1. INTRODUCTION

The Internet, interconnecting information and business services, has made available to on-line users an over abundance of information and very large product catalogues. Hence, users trying to decide what information to consult or what products to choose may be overwhelmed by the number of options. Recommender systems are intelligent applications that try to solve information overload problem by recommending relevant items to a user [2, 11]. Here an item is usually a descriptive information about a product such as a movie, a book or a place of interest. Recommender systems are personalized Information Retrieval systems where users make generic queries, such as, "suggest a movie to be watched with my family this night".

Collaborative Filtering (CF) is a recommendation technique that emulates a simple and effective social strategy called "word-of-mouth" and is now largely applied in the "social" web. For example, amazon.com recommends items that user could be interested to buy or delicious.com recommends the links that were tagged by alike users with commonly used tags. CF recommendations are computed by leveraging historical log data of users' online behavior [12]. The relevance of an item is usually expressed and modeled by the explicit user's rating. The higher is the rating that a user assigned to an item, the more relevant is the item for the user. CF assumes that the user's recorded ratings for items can help in predicting the ratings of like-minded users. We want to stress that this assumption is valid only to some extent. In fact, the user's general interests can be relatively stable, but, the exact evaluation of an item can be influenced by many additional and varying factors. In certain domains the consumption of the same item can lead to extremely different experiences when the context changes [1, 4]. Therefore, relevance of an item can depend on several contextual conditions. For instance, in a tourism application the visiting experience to a beach in summer is strikingly different from the same visit in winter (e.g., during a conference meeting). Here context plays the role of query refinement, i.e., a context-aware recommender system must try to retrieve the most relevant items for a user, given the knowledge of the current context. However, most CF recommender systems do not distinguish between these two experiences, thus providing a poor recommendation in certain situations, i.e., when the context really matters.

Context-aware recommender systems is a new area of research [1]. The classical context-aware reduction based approach [1] extended the classical CF method adding to the standard dimensions of users and items new ones representing contextual information. Here recommendations are computed using only the ratings made in the same context as the target one. For each contextual segment, i.e., sunny weekend, algorithm checks (using cross validation) if generated predictions using only the ratings of this segment are more accurate than using full data set. The authors use a hierarchical representation of context, therefore, the exact granularity of the used context is searched (optimized) among those that improve the accuracy of the prediction. Similarly, in our approach we enrich the simple 2-dim. CF matrix with a model of the context comprising a set of features either of the user, or the item, or the evaluation. We adopt the definition of context introduced by Dey, where "Context is any information that can be used to characterize the situation of an entity" [8]. Here, the entity is an item consumption that can be influenced by contextual variables

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

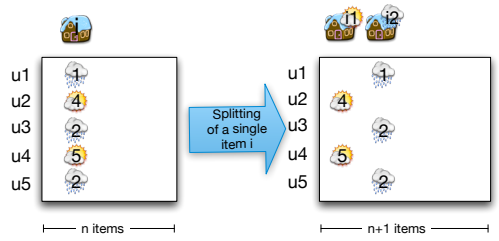


Figure 1: Item splitting

describing the state of the user and the item. In this paper we propose a new approach for using these contextual dimensions to pre-filter items' ratings. Actually, to be precise, the set of ratings for an item is not filtered but it is split into two subsets according to the value of a contextual variable, e.g., ratings collected in "winter" or in "summer" (the contextual variable is the season of the rating/evaluation). These two sets of ratings are then assigned to two new fictitious items (e.g. beach in winter and in summer).

This paper extends the results presented in [5, 6]. Here we evaluate the same item splitting technique in a different set of experiments, namely we measure precision and recall, whereas previously we used MAE. Also the nine semi-synthetical data sets are generated differently. Moreover, we extended our analyzes by studying the behavior of item splitting with respect to the various Information Gain thresholds.

2. ITEM SPLITTING

Our approach extends the traditional CF data model by assuming that each rating r_{ui} in a $m \times n$ users-items matrix, is stored (tagged) together with some contextual information $c(u,i) = (c_1, \ldots, c_n), c_j \in C_j$, describing the conditions under which the user experience was collected (c_j is a nominal variable). The proposed method identifies items having significant differences in the ratings (see later the exact test criteria). For each one of these items, our algorithm splits its ratings into two subsets, creating two new artificial items with ratings belonging to these two subsets. The split is determined by the value of one contextual variable c_j, i.e., all the ratings in a subset have been acquired in a context where the contextual feature c_j took a certain value. So, for each item the algorithm seeks for a contextual feature c_j that can be used to split the item. Then it checks if the two subsets of ratings have some (statistical significant) difference, e.g., in the mean. If this is the case, the split is done and the original item in the ratings matrix is replaced by the two newly generated items. In the testing phase, the rating predictions for the split item are computed for one of the newly generated item. For example, assume that an item i has generated two new items i1 and i2, where i1 contains ratings for item i acquired in the contextual condition cj = v, and i2 the ratings acquired in context cj $\bar{v}$, hence the two sets partition the original set of ratings. Now assume that the system needs to compute a rating prediction for the item i and user u in a context where $c_j = x$. Then the prediction is computed for the item i_1 if $x = v$, or i_2 if $x \neq v$, and is returned as the prediction for i.

Figure 1 illustrates the splitting of one item. As input, the item splitting step takes a $m \times n$ rating matrix of m users and n items and outputs a $m \times (n+1)$ matrix. The total number of ratings in the matrix does not change, but a new item is created. This step can be repeated for all the items having a significant dependency of their ratings on the value of one contextual variable. In this paper we focus on a simple application of this method where an item is split only into two items, using only one selected contextual variable. A more aggressive split of an item into several items, using a combination of features, could produce even more "specialized" items, but potentially increasing data sparsity. We note again, that for the same user, and different items, one can in principle obtain ratings in different contexts, as in our context model context depends on the rating. Therefore, items i_1 and i_2 could overlap, i.e., could be rated both by the same user in different contextual conditions. However, such situation are not very common.

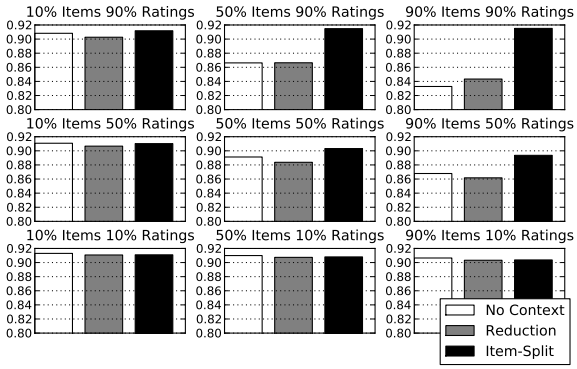
We conjecture that the splitting could be beneficial if the ratings within each newly obtained item are more homogenous, or if they are significantly different in the new items coming from a split. One way to accomplish this task is to define an impurity criteria t [7]. So, if there are some candidate splits $s \in S$, which divide i into i_1 and i_2, we choose the split s that maximizes $t(i,s)$ over all possible splits in S. A split is determined by selecting a contextual variable and a partition of its values in two sets. Thus, the space of all possible splits of item i is defined by the context model C. In this work we analyzed t_{IG} impurity criteria. $\boldsymbol{t_{IG}(i,s)}$ measures the information gain (IG), also known as Kullback-Leibler divergence [10], given by s to the knowledge of the item i rating: $t_{IG} = H(i) - H(i_1)P_{i_1} + H(i_2)P_{i_2}$ where H(i) is the Shannon Entropy of the item i rating distribution and P_{i_1} is the proportion of ratings that i_1 receives from item i. To ensure reliability of this statistic we compute it only for a split S that could potentially generate items each containing 4 or more ratings. Thus, algorithm never generates items with less than 4 ratings in the profile.

3. EXPERIMENTAL EVALUATION

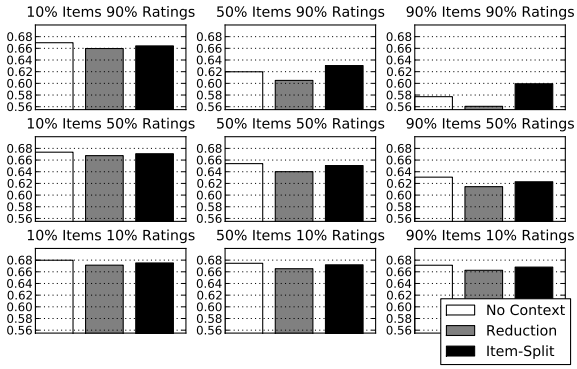
We tested the proposed method on nine semi-synthetic data sets with ratings in {1, 2, 3, 4, 5}. The data sets were generated using Yahoo![1] Webscope movies data set contains 221K ratings, for 11,915 movies by 7,642 users. The semi-synthetic data sets were used to analyze item splitting when varying the influence of the context on the user ratings. The original Yahoo! data set contains user age and gender features. We used 3 age groups: users below 18 (u18), between 18 and 50 (18to50), and above 50 (a50). We modified the original Yahoo! data set by replacing the gender feature with a new artificial feature $c \in \{0,1\}$ that was assigned randomly to the value 1 or 0 for each rating. This feature c is representing a contextual condition that could affect the rating. We randomly choose $\alpha * 100\%$ items from the data set and then from these items we randomly chose $\beta * 100\%$ of the ratings to modify. We increased (decreased) the rating value by one if $c = 1$ ($c = 0$) and if the rating value was not already 5 (1). For example, if $\alpha = 0.9$ and $\beta = 0.5$ the corresponding synthetic data set has 90% of altered items' profiles that contains 50% of changed ratings. We generated nine semi-synthetic data sets varying $\alpha \in \{0.1, 0.5, 0.9\}$ and $\beta \in \{0.1, 0.5, 0.9\}$. So, in these data set the contextual condition is more "influencing" the rating value as α and β increase.

In this paper we used matrix factorization ($FACT$) as the

[1]Webscope v1.0, http://research.yahoo.com/



(a) Precision



(b) Recall

Figure 2: Comparison of contextual pre-filtering methods.

rating prediction technique. We used the algorithm implemented and provided by Timely Development[2]. *FACT* uses 60 factors and the other parameters are set to the same values optimized for another data set (Netflix), so it might not be the best setting, but all the system variants that we compared used the same settings. To evaluate the described methods we used 5-fold cross-validation and measured precision and recall. The usage of precision and recall in recommender systems needs some clarification. These measures, in its purest sense, are impossible to measure as they would require the knowledge of the rating (relevance) of each item and user combination [9]. Usually there are thousands of candidate items to recommend (11K in our case) and just for a small percentage of them we know the true user's evaluation (typically less than 1%) . Herlocker et al. [9] proposed to approximate these measures by computing the prediction just for $user \times item$ pairs that are present in the ratings data set, and consider items worth recommending (relevant items) only if the user rated them 4 or 5. We computed the measures on full test set (of each fold), while trained the models on the train set. Please refer to [5] for additional experiments. These include the evaluation of other impurity criteria, the performance of the proposed method on the original Yahoo! data set, and experiments using other prediction methods such as user-based CF while computing Mean Absolute Error (MAE).

3.1 Context-aware Prediction Methods

To understanding the potential of item splitting in a context-dependent set of ratings we tested this approach on the semi-synthetical data sets described earlier, i.e., replacing the gender feature with a new contextual variable that does influence the ratings. The baseline method is FACT when no contextual information is considered. It is compared with the context-aware reduction based approach [1], and our item splitting technique. Figure 2 shows comparison of three methods for the nine semi-synthetic data sets. For each data set we computed precision and recall. We considered item as worth recommending if algorithm made a prediction greater or equal to 4. For all the nine data sets the algorithm splits an item if any split leads to an IG bigger than 0.01. The small IG threshold value led to a good results in our previous experiments [6] and it allows algorithm to split up to 15% of items (depending on the data set). In Subsection 3.3 we report result while choosing bigger values that typically decrease the impact of item splitting. As we expected, the smaller is the impact of the contextual feature c, the smaller is the improvement of the performance measure obtained by the methods that do use the context. In fact, item splitting improved the performance of baseline method for 4 data sets: $\alpha \in \{0.5, 0.9\}, \beta \in \{0.5, 0.9\}$. The highest improvement for precision of 9.9% was observed for the data set $\alpha = 0.9, \beta = 0.9$ where most items and most ratings were influenced by the artificial contextual feature. Increasing the value of α and β, i.e., increasing the number of items and ratings that are correlated to the value of the context feature, decreased the overall precision and recall of the baseline method. We conjecture, that the contextual condition plays the role of noise added to the data, even if this is clearly not noise but a simple functional dependency from a hidden variable. In fact, *FACT* cannot exploit the additional information brought by this feature and cannot effectively deal with the influence of this variable.

Reduction based approach increased precision by 1.3% only for $\alpha = 0.9, \beta = 0.9$ data set. This is the data set, where artificial contextual feature has highest influence on the ratings and 90% of items are modified. In [1] the authors optimized MAE when searching for the contextual segments where the context-dependent prediction improves the default one (no context). Here, we searched for the segments where precision and recall is improved and we used all better performing segments to make the predictions. For example, Figure 2(a) reports the precision of reduction based. To conduct this experiment, the algorithm first sought (optimized) the contextual segments where precision is improved (using a particular split of train and test data). Then, when it has to make a rating prediction, used either only the data in one of these segments, i.e., if the prediction is for a item-user combination in one of the found segments, or all the data, i.e., if the item-rating is in one contextual conditions where no improvements can be found with respect to the baseline. Note, that in all three data sets where $\alpha = 0.5, \beta \in \{0.1, 0.5, 0.9\}$ the results are similar to the baseline approach. In these cased the reduction base approach does consider the segments generated using the artificial feature. However, the data set was constructed in such a way that half of the items do not have ratings' dependencies on the artificial feature, and no benefit is observed.

These experiments show that both context-aware pre-filtering

[2]http://www.timelydevelopment.com

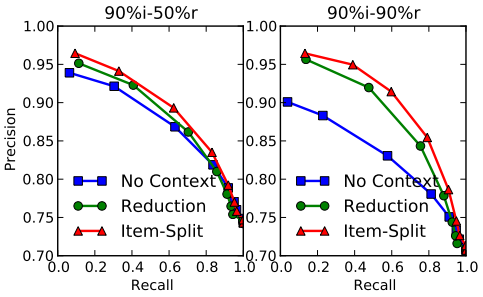


Figure 3: Precision/recall curves for two data sets.

approaches can outperform the base line *FACT* CF method, when the context influences the ratings. It is worth noting that item splitting is computationally cheaper and it performed better than reduction based. Note also that, accuracy could depend on the particular baseline prediction algorithm, i.e., FACT in our experiments. However, we choose FACT as it is now currently largely used, and in our previous experiments it outperformed traditional user-based CF method [5].

3.2 Precision Versus Recall

In this section we illustrate the precision/recall curves for the three selected methods. For this experiments we reused the three data sets: $\alpha = 0.1, \beta \in \{0.1, 0.5, 0.9\}$. As was done in the previous experiment, we set the IG threshold to 0.01. For the reduction based approach we optimized precision. The results can be seen in Figure 3. The left figure shows results for $\alpha = 0.9, \beta = 0.5$ data set and the right figure for $\alpha = 0.9, \beta = 0.9$. We skip the $\alpha = 0.9, \beta = 0.1$ data set, as for this data set all three methods perform similarly to each other. Each curve was computed by varying the threshold at which a recommendation is done. For example, all methods obtained the highest precision when recommending the items that were predicted as rating 5. In this case, we do not recommend the items that were predicted with a lower rating. Note that we always count recommendation as relevant if user rated the item 4 or 5. We set the threshold to values equal to {1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5}. Note, that previous experiment (see. Figure 2) was done with the recommendation threshold equal to 4. The recall is equal to 1 if we recommend all the items, i.e., those predicted with a rating of 1 and higher. Even at this level of recall, the precision is more than 70%. This can be explained by the high fraction of high ratings in the data set.

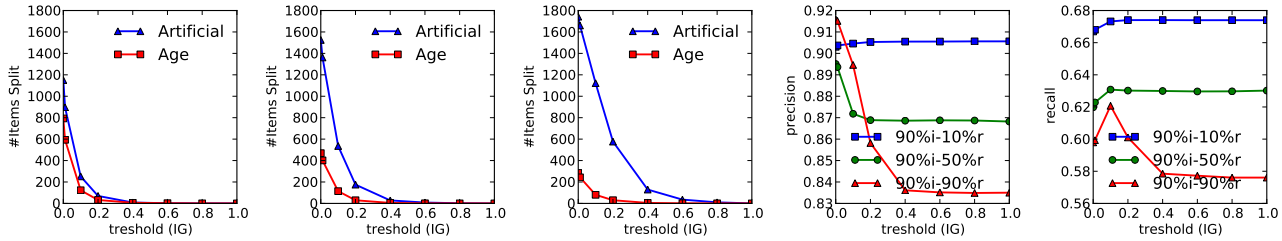
Recommender systems usually try to improve precision. Having recall as small as 0.01, we could still be able to recommend too many items for user to consume, i.e., approximately 119 items in our data set. Interestingly, as we can see it is also much harder to make precise recommendations than to obtain high recall. The curves for all three methods get flat when approaching precision 0.97. At this point we recommend only the items that were predicted with rating 5. This is the maximum possible predicted rating by *FACT* and precision can not be improved by varying the threshold at which recommendation is done. We also observe, that we can achieve higher maximum precision for item splitting method comparing to other methods. When $\alpha = 0.9$ and $\beta = 0.9$, the highest precision value for item split improves by 7% the baseline method. The improvement when $\alpha = 0.9$ and $\beta = 0.5$ is 2.7%. This experiment gives valuable insights into the behavior of reduction base approach. We see, that at each level of the recommendation threshold it shows a higher recall value than the other two methods. At the highest level of precision, reduction based approach is close to item splitting and gives improvement of 6.1% in precision for $\alpha = 0.9, \beta = 0.9$ data set and 1.3% for $\alpha = 0.9, \beta = 0.5$ data set. But, the precision/recall curve of reduction based is always below than that of item split.

In conclusion we want to note that considering both precision and recall, we see that both context-aware recommendation methods yields quite similar results. More noticeably, both methods outperforms baseline CF which does not take context into account.

3.3 Item Splitting for Various IG Thresholds

To better understand the item splitting method we further analyzed the prediction processes. We looked at the number of items the algorithm splits and also on which attribute the split was performed. For this purpose we varied the item splitting threshold parameter. For this experiment we used t_{IG} impurity measure and the three data sets: $\alpha = 0.9\beta \in \{0.1, 0.5, 0.9\}$. The summary of the results are shown in Figure 4. Figures 4(a), 4(b), 4(c) show the number of splits that the item split algorithm performs varying the IG threshold for the three considered data sets. When using $\alpha = 0.9, \beta = 0.1$ the algorithm chooses the artificial feature approximately twice as often as the age feature. More precisely, when the threshold is $IG = 0.2$ item split splits 101.8 items (on average in 5 folds); the artificial feature was chosen 69.8 and age feature was chosen 32 times. When the influence of artificial feature increases, a higher proportion of items are split using the artificial feature. For the $\alpha = 0.9, \beta = 0.9$ data set and $IG = 0.2$ it splits 576.8 items using the artificial feature and 29.8 using the age feature. Note, that despite IG favors attributes with many possible values [10] item splitting chooses the attribute having larger influence on the rating. We further observe that the number of split items is not large. For all three data sets we split no more than 2050 items (17%). This low number can be explained by looking at the size of items' profiles. Note that in the considered data sets the average number of ratings per item is 18.5. Algorithm splits item only if the newly generated item has at least 4 ratings. Therefore, item must have a minimum of 8 ratings to be considered for splitting. Lowering the minimum number of ratings in the item profile, could cause unreliable computation of statistics and was observed to decrease the overall performance.

Figures 4(d), 4(e) shows precision and recall accuracy measures for three data sets. We observe, that item splitting is only beneficial when context (i.e., artificial feature here) has an high influence on the rating. The best performance for the $\alpha = 0.9, \beta = 0.1$ data set, both for recall and precision, is obtained when no items are split. Each split of an item affects also the prediction for the items that are not split. Splitting an item is equivalent to create two new items and deleting one, therefore, it causes a modification of the data set. When CF generates a prediction for a target user-item pair all the other items' ratings, including those in the new items coming from some split, are used to build that prediction. In [5] we observed that we can increase the performance on split items, but at the same time the decrease of performance on the untouched items can cancel any benefit. When $\alpha = 0.9, \beta = 0.5$ the situation is dif-



(a) 90%i-10%r (b) 90%i-50%r (c) 90%i-90%r (d) precision (e) recall

Figure 4: Item splitting behavior for different thresholds.

ferent. We observe, that here splitting more items leads to an increase in precision and decrease in recall. Finally, for $\alpha = 0.9, \beta = 0.9$ splitting more items increase the precision and recall, and this is maximum when the IG threshold is equal to 0.1. In conclusion, we could regard item split as a more dynamical version of reduction based. Here the split is done for each item separately and using an external measure (such as IG) to decide if the split is needed. Using the IG criteria, splitting items is beneficial when context highly influences the ratings.

4. CONCLUSIONS AND FUTURE WORK

This paper evaluates a contextual pre-filtering technique for CF, called item splitting. Based on the assumption that certain items may have different evaluations in different contexts, we proposed to use item splitting to cope with this. The method is compared with a classical context-aware pre-filtering approach [1] which uses extensive searching to find the contextual segments that improve the baseline prediction. As a result we observed that despite the increased data sparsity, item splitting is beneficial, when some contextual feature separates the item ratings into two more homogeneous rating groups. However, if the contextual feature is not influential the splitting technique sometimes produced a minor decrease of the precision and recall. Item-splitting outperforms reduction based context-aware approach when *FACT* CF method is used. Moreover, the method is more time and space efficient and could be used with large context-enriched data bases.

The method we proposed can be extended in several ways. For instance one can try to split the users (not the items) according to the contextual features in order to represent the preferences of a user in different contexts by using various parts of the user profile. Another interesting problem is to find a meaningful item splitting in continuous contextual domains such as time or temperature. Here, the splitting is not easily predefined but have to be searched in the continuous space. Finally, item splitting could ease the task of explaining recommendations. The recommendation can be made for the same item in different context. The contextual condition on which the item was split could be mentioned as justifications of the recommendations. For example, we recommend you to go to the museum instead of going to the beach as it will be raining today. We would also like to extend our evaluation of the proposed algorithm. First of all, we want to use real world context-enriched data. Moreover, we want to evaluate precision and recall at top-N recommendation list. At the end, we want to develop a solution to be able to deal with missing contextual values.

5. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactins on Information Systems*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, editors, *RecSys*, pages 335–336. ACM, 2008.

[4] S. S. Anand and B. Mobasher. Contextual recommendation. In *Lecture Notes In Artificial Intelligence*, volume 4737, pages 142–160. Springer-Verlag, Berlin, Heidelberg, 2007.

[5] L. Baltrunas and F. Ricci. Context-based splitting of item ratings in collaborative filtering. In L. D. Bergman, A. Tuzhilin, R. Burke, A. Felfernig, and L. Schmidt-Thieme, editors, *RecSys*, pages 245–248. ACM, 2009.

[6] L. Baltrunas and F. Ricci. Context-dependent items generation in collaborative filtering. In G. Adomavicius and F. Ricci, editors, *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, 2009.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[8] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, February 2001.

[9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.

[10] J. R. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, January 1993.

[11] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[12] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324. Springer Berlin / Heidelberg, 2007.

Thinking of a System for Image Retrieval

Giovanna Castellano
Università degli Studi di Bari
"Aldo Moro"
via Orabona 4
Bari, Italy
castellano@di.uniba.it

Gianluca Sforza*
Università degli Studi di Bari
"Aldo Moro"
via Orabona 4
Bari, Italy
gsforza@di.uniba.it

Alessandra Torsello
Università degli Studi di Bari
"Aldo Moro"
via Orabona 4
Bari, Italy
torsello@di.uniba.it

ABSTRACT

Increasing applications are demanding effective and efficient support to perform retrieval in large collections of digital images. The work presented here is an early stage research focusing on the integration between text-based and content-based image retrieval. The main objective is to find a valid solution to the problem of reducing the so called semantic gap, i.e. the lack of coincidence existing between the visual information contained in an image and the interpretation that a user can give of it. To address the semantic gap problem, we intend to use a combination of several approaches. Firstly, a linking between low-level features and text description is obtained by a semi-automatic annotation process, which makes use of shape prototypes generated by clustering. Precisely, the system indexes objects based on shape and groups them into a set of clusters, with each cluster represented by a prototype. Then, a taxonomy of objects that are described by both visual ontologies and textual features is attached to prototypes, by forming a visual description of a subset of the objects. The paper outlines the architecture of the system and describes briefly algorithms underpinning the proposed approach.

Categories and Subject Descriptors

H [**Information Storage and Retrieval**]

General Terms

Image retrieval

Keywords

Content-based image retrieval, Semantic image retrieval

1. INTRODUCTION

By the end of the last century the question was not whether digital image archives are technically and economically viable, but rather how these archives would be efficient and informative. The attempt has been to develop intelligent and efficient human-computer interaction systems, enabling the user to access vast amounts of heterogeneous image sets, stored in different sites and archives. Additionally, the continuously increasing number of people that should access to such collections further dictates that more emphasis be put on attributes such as the user-friendliness and flexibility of any multimedia content retrieval scheme.

The very first attempts at image retrieval were based on exploiting existing image captions to classify images according to predetermined classes or to create a restricted vocabulary [5]. Although relatively simple and computationally efficient, this approach has several restrictions mainly deriving from the use of a restricted vocabulary that neither allows for unanticipated queries nor can be extended without re-evaluating the possible connection between each item in the database and each new addition to the vocabulary. Additionally, such keyword-based approaches assume either the pre-existence of textual annotations (e.g. captions) or that annotation using the predetermined vocabulary is performed manually. In the latter case, inconsistency of the keyword assignments among different indexers can also hamper performance. Recently, a methodology for computer-assisted annotation of image collections was presented [24].

To overcome the limitations of the keyword-based approach, the use of the visual content has been proposed, leading to Content-Based Image Retrieval(CBIR) approaches [6]. CBIR systems utilize the visual content of images to perform indexing and retrieval, by extracting low-level indexing features, such as color, shape, and texture. In this case, pre-processing of images is necessary as the basis on which features are extracted. The pre-processing is of coarse granularity if it involves processing of images as a whole, whereas it is of fine granularity if it involves detection of objects within an image [1]. Then, relevant images are retrieved by comparing the low-level features of each item in the database with those of a user-supplied sketch or, more often, a key image that is either selected from a restricted image set or is supplied by the user (query-by-example). Several approaches have appeared in the literature which perform visual querying by examples taking into account different facets of pictorial data to express the image contents, such as color [21], object shape [2], texture [14], or a combination of them [8, 18, 20]. Among these, search by matching shapes of image portions is one of the most natural way to pose a query in image databases.

Though many sophisticated algorithms have been designed to describe color, shape, and texture features, these algorithms cannot adequately model image semantics. Indeed, extensive experiments on CBIR show that low-level contents

*Corresponding author

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

often fail to describe the high-level semantic concepts in user's mind [25]. Also, CBIR systems have limitations when dealing with broad content image databases [16]; indeed, in order to start a query, the availability of an appropriate key image is assumed; occasionally, this is not feasible, particularly for classes of images that are underrepresented in the database. Therefore, the performance of CBIR systems is still far from user's expectations.

Summarizing, current indexing schemes for image retrieval employ descriptors ranging from low-level features to higher-level semantic concepts [23]. So far, significant work has been presented on unifying keywords and visual contents in image retrieval, and several hybrid methods exploiting both keywords and the visual content have been proposed [17, 12, 26]. Depending on how low-level and high-level descriptors are employed and/or combined together, different levels of image retrieval can be achieved. According to [7], three levels of image retrieval can be considered:

- Level 1: Low-level features such as color, texture, shape or the spatial location of image elements are exploited in the retrieval process. At this level, the system supports queries like *find pictures like this* or *find pictures containing blue squares.*

- Level 2: Objects of given type identified by low-level features are retrieved with some degree of logical inference. An example of query is *find pictures in which my father appears.*

- Level 3: Abstract attributes associated to objects are used for retrieval. This involves a significant amount of high-level reasoning about the meaning of the objects or scenes depicted. An example of query is *find pictures of a happy woman.*

Retrieval including both Level 2 and Level 3 together is referred to as semantic image retrieval. The gap between Level 1 and Level 2 is known as semantic gap, which is "the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation" [19]. Retrieval at Level 3 is quite difficult, therefore current systems mostly perform retrieval at Level 2, which requires three fundamental steps: (1) extraction of low-level image features, (2) definition of proper similarity measures to perform matching, (3) reducing the semantic gap. Clearly, step (3) is the most challenging one, since it requires providing a link between low-level features (visual data) and high-level concepts (semantic interpretation of visual data).

Currently, various approaches have been proposed to reduce the semantic gap between the low-level features of images and the high-level concepts that are understandable by human. According to [11], they can be broadly grouped into four main categories:

- Use of ontologies [15]. Ontologies can be used to provide an explicit, simplified and abstract specification of knowledge about the domain of interest; this is obtained by defining concepts and relationships between them, according to the specific purpose of the considered problem. This approach exploits the possibility to simply derive semantics from our daily language. Then, different descriptors can be related to the low-level features of images in order to form a vocabulary that provides a qualitative definition of high-level query concepts. Finally, these descriptors can be mapped to high level semantics, based on our knowledge. This approach works fine with small databases containing specifically collected images. With large collections of images with various contents, more powerful tools are required to learn the semantics.

- Automatic image annotation [22]. This approach consists in exploiting supervised or unsupervised learning techniques to derive high-level concepts from images. In particular, supervised learning techniques are used to predict values of a semantic category based on a set of training samples. However, supervised learning algorithms present some disadvantages strictly related to the nature of this kind of technique, that require a large amount of labeled data to provide effective learning results. This represents a problem when the application domain changes and new labeled samples have to be provided. Clustering is the typical unsupervised learning technique used for retrieval purpose. In this approach, images are grouped on the basis of some similarity measure, so that a class label is associated to each derived cluster. Images into the same cluster are supposed to be similar to each other (i.e. having similar semantic content). Thus, a new untagged image that is added to the database can be indexed by assigning it to the cluster that better matches with the image.

- Relevance feedback [13]. This approach concerns the possibility to learn the intentions of users and their specific needs by exploiting information obtained during their interactions with the system. In particular, when the system provides the initial retrieval results, the user judges these by indicating if they are relevant/irrelevant (and eventually the degree of relevance/irrelevance). Then, a learning algorithm is used to learn the user feedback, which will be exploited in order to provide results that better satisfy the user needs.

- Generating semantic templates [27]. This method is based on the concept of visual semantic template that includes a set of icons or objects denoting a personalized view of concepts. Feature vectors of these objects are extracted for query process. Initially, the user has to define the template of a concept by specifying, for example, the objects and their spatial and temporal constraints and the weights assigned to each feature for each object. Finally, through the interaction with users, the system move toward a set of queries that better express the concept in the user mind. Since this method requires the user to know the image features, it could be quite difficult for ordinary users.

Along with state-of-art directions in the field of IR, in this paper we present the idea of an IR system supporting retrieval at Level 2. Precisely, we intend to provide a solution to the problem of semantic gap in IR by designing a methodology based on a combination of several approaches, which is oriented to exploit both the visual and the semantic content of images. This is achieved making use of clustering and visual ontologies. In the following, all the approaches

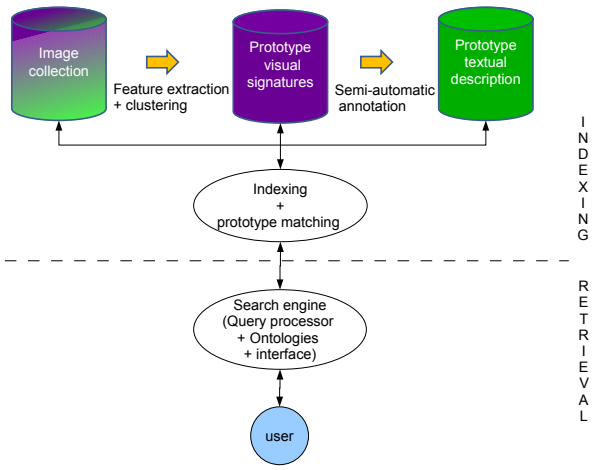


Figure 1: The system architecture.

underpinning the proposed IR methodology are briefly described and the architecture of the system is outlined.

2. OVERVIEW OF THE IR SYSTEM

The proposed system is intended to perform image retrieval by exploiting both the visual and the semantic content of images. As concerns the visual content, in this preliminary phase of the research we focus only on shape content. In fact, we aim to deal with specific domain images containing objects that have a distinguishable shape meaning. Therefore, we assume that indexing and querying are only based on shape matching. The system will allow the user to query the image database not only by shape sketches and by keywords but also by "concepts describing shapes". The general architecture of the proposed IR system is reported in fig. 1.

As it can be seen, several tasks are carried out in order to derive visual and textual features of shapes contained in images. These tasks are:

1. Feature extraction: detecting shapes in images;
2. Clustering: grouping similar shapes into prototypes;
3. Semi-automatic annotation: associating keywords to prototypes;
4. Search.

In the following we describe how each task is carried out.

2.1 Feature extraction

In the proposed system, each image in the database is stored as a collection of objects' shapes contained in it. In order to be stored in the database, every image is processed to identify objects appearing in it. Image processing starts with an edge detection process that extracts all contours in the image. Then, using the derived edges, a shape detection process is performed to identify different objects included in the image and determine their contours. Finally, Fourier descriptors are computed on each contour and retained as visual signatures of the objects in a separate database.

2.2 Clustering

Once all shapes have been detected from images and represented as visual signatures vectors, a set of shape prototypes is automatically defined by an unsupervised learning process that performs clustering on visual signatures (Fourier descriptors) of shapes, so as to categorize similar shapes into clusters. Each resulting cluster C_i is represented by a shape prototype $\mathbf{p}_i$, that is computed by averaging visual signatures of all shapes belonging to the cluster. We intend to apply a hierarchical clustering, in order to generate a hierarchy of prototypical shapes. Each node of the hierarchical tree is associated with one prototypical shape. Root nodes of the tree represent general prototypes, intermediate nodes represent general shapes, leaf nodes represent specific shapes.

During the interaction of the user with the system, the hierarchical tree is incrementally updated. Whenever a new shape is considered (i.e. each time a new image containing relevant object shapes is added to the database), we evaluate its matching against all existing prototypes, from root nodes to *pre-leafs*(final) nodes, according to a similarity measure defined on visual signatures. If the new shape matches a final prototype with a sufficient degree, then the corresponding prototype is updated by averaging the features of shapes that belong to the corresponding cluster [10]. Otherwise, a new prototype is created, corresponding to the new shape.

The use of shape prototypes, which represent an intermediate level of visual signatures, facilitates the subsequent tasks 3. and 4. Actually, prototypes facilitate the annotation process, since only a reduced number of shapes (the prototypical ones) need to be manually annotated. Secondly, the use of prototypes simplifies the search process. Indeed, since only a small number of objects is likely to match any single user query, a large number of unnecessary comparisons is avoided during search by performing matching with shape prototypes rather than with specific shapes. In other words, prototypes acts as a filter that reduces the search space quickly while discriminating the objects.

2.3 Semi-automatic annotation

Once shape prototypes have been derived, a semi-automatic annotation process is applied to associate text descriptions to identified object shapes. The process is semi-automatic since it involves a manual annotation only for prototypes: shapes immediately attached in the hierarchy are automatically annotated, since they inherit descriptions from their prototypes.

Every semantic class that is of interest in the considered image domain (e.g. for ours, glasses, bottles, etc.) will be described by a *visual ontology* (VO), which is intended as a textual description, made of concepts and relationships among them, of the visual content of a prototypical shape [9, 4]. We figure the lexicon used to define the VOs to be as much intuitive as possible, so as to evocate the particular shape it describes. We plan that the system will be supplied of a basic set of domain dependent VOs, one for each considered semantic class.

Of course, different prototypical shapes may convey the same semantic content (e.g., several different shapes may convey the concept of *glass*). We consider such prototypes to belong to the same semantic class. Shape prototypes belonging to the same semantic class will share about the same VO structure, obviously with the appropriate differences.

As an illustrative example, we sketch some possible relationships included in a VO that refers to the semantic class *glass*:

- *wine glass IS SPECIALIZATION OF glass*;
- *bottom IS PART OF wine glass*;
- *wavy shape IS PROPERTY OF bottom.*

The combined use of prototypes and VOs provides a powerful mechanism for automatic annotation of shapes. Every time the user adds a new shape to the database, the system associates the shape to the most similar prototype, which is related to a semantic class and linked to a VO. Thus the new shape inherits all the semantic descriptions associated to the selected prototype in an automatic fashion. Then, a feedback from the user is considered. Namely, the user may accept the choice operated by the system, or reject it. In the latter case, there are two possibilities: the user can select the proper prototype with the related VO from the existing ones, or, if no one can be associated to the shape, the user can create a new prototype (using the new shape) and manually annotate it by modifying the VO incorrectly assigned by the system previously.

2.4 Search

The engine mechanism is designed to allow users to submit sketch-based, text-based and concept-based queries.

The results of the sketch-based search emerge from a matching between the submitted sample shape and the created prototypes. Precisely, when the user presents a query in the form of an object sketch, the system formulates the query, performing feature extraction by translating that object into a shape model. The extracted query feature is submitted to compute similarity between the query and prototypes first. This is made by considering shapes as points of a feature space. Having characterized each shape as a vector of Fourier descriptors, we simply evaluate dissimilarity between two shapes in terms of Euclidean distance between two vectors of descriptors. Of course, other similarity measures can be considered, encapsulating the human perception of shape similarity (this is an interesting issue that we would like to deepen in future). After sorting the prototypes in terms of similarity, the system returns images containing objects indexed by the prototypes with highest similarities.

The results of the text-based search emerge from a matching between the submitted textual query and textual descriptions associated to prototypes. Namely, when a query is formulated in terms of keywords, the system simply returns images including the objects indexed by the prototypes labeled with that keywords. As before, high-matching prototypes are selected to provide shapes to be visualized as search results.

Finally, when both a visual and textual content are exploited by the user querying the image database, images returned from the two approaches separately, are merged together in a single output set.

3. FIRST STEPS TOWARD THE SYSTEM DEVELOPMENT

In this preliminary phase of the research, only the main functions for tasks 1. and 4. described above have been implemented in the system. For tests during the development

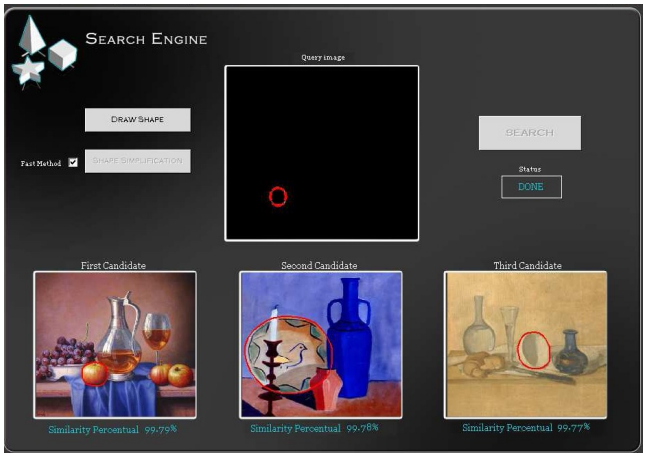


Figure 2: An initial search engine interface.

of the system, we considered an image database from the art domain. The database, used in other IR works [3] includes digitalized images representing still-object paintings by the Italian artist Giorgio Morandi.

As concerns task 1., various image processing tools that are necessary to extract shape features from the image objects have been developed, including edge detection methods, as well as enhancement and reconstruction functionalities. Basic image processing methods were included from the *ImageJ* image analysis software[1], such as thresholding methods (e.g. Canny, Prewitt and Sobel) for automatic detection of objects boundaries lying in images. Having the possibility to act on contrast and brightness properties, the user can adjust the image appearance to refine the extraction of the shapes of objects. The shape identification is made automatically through an edge following algorithm. When the result of shape identification is not satisfying, the user is given the possibility to correct boundaries or to manually draw boundaries directly on the image.

As concerns task 4., the retrieval graphical interface has been developed, that enables users to query the system and to inspect search results (fig. 2). Also, the computation of Euclidean dissimilarity measures for shape prototype matching has been included in the system.

Currently, the system provides also the interfaces for browsing the database and insert new images.

4. CONCLUSIONS

In this paper a preliminary proposal of an IR system has been presented. The system is intended to solve the problem of semantic gap by exploiting clustering and visual ontologies. The use of a visual ontology is motivated by the necessity of reproducing the capacity of a human in describing her visual perception by means of the visual concepts she possesses. From the point of human-computer interaction view, visual ontologies provide a bridge between low-level features of images and visual representation of semantic contained in images. Compared to symbolized ontology, visual ontologies can represent complex image knowledge in a more detailed and intuitive way, so that no expert knowledge is needed to process a complicated knowledge representation of images.

[1]http://rsbweb.nih.gov/ij

The binding created by visual ontologies between image objects and their description, enables the proposed IR system to perform a conceptual reasoning on the collection of images, also when treating with pure content-based queries. Thus, different forms of retrieval become possible with the proposed system:

1. text-based: queries are lexically motivated, i.e. they express objects by their names (keywords);
2. content-based: queries are perceptually motivated, i.e. they express objects by their visual apparency;
3. semantic retrieval: queries are semantically motivated, since they express objects by their intended meaning, i.e. in terms of concepts and their relationships.

Currently, we are continuing to develop the proposed IR system. To this aim, we are looking for the best appropriate clustering algorithm to derive significant shape prototypes and analyzing methods to create visual ontologies.

5. REFERENCES

[1] W. Al-Khatib, Y. F. Day, A. Ghafoor, and P. B. Berra. Semantic modeling and knowledge representation in multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):64–80, 1999.

[2] S. Arivazhagan, L. Ganesan, and S. Selvanidhyananthan. Image retrieval using shape features. *International journal of imaging science and engineering (IJISE)*, 1(3):101–103, 2007.

[3] A. D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:121–132, 1997.

[4] M. Bouet and M.-A. Aufaure. *Multimedia Data Mining and Knowledge Discovery*, chapter New Image Retrieval Principle: Image Mining and Visual Ontology, pages 168–184. Springer, 2007.

[5] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, and A. Pathria. Multimedia document presentation, information extraction, and document formation in minos: a model and a system. *ACM Trans. Inf. Syst.*, 4(4):345–383, 1986.

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[7] J. Eakins and M. Graham. Content-based image retrieval. University of Northumbria Technical Report, 1999.

[8] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244, 1996.

[9] S. Jiang, T. Huang, and W. Gao. An ontology-based approach to retrieve digitized art images. In *WI '04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 131–137, Washington, DC, USA, 2004. IEEE Computer Society.

[10] K.-M. Lee and W. Street. Cluster-driven refinement for content-based digital image retrieval. *Multimedia, IEEE Transactions on*, 6(6):817–827, 2004.

[11] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282, 2007.

[12] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 31–37, New York, NY, USA, 2000. ACM.

[13] S. MacArthur, C. Brodley, and C.-R. Shyu. Relevance feedback decision trees in content-based image retrieval. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVLŠ00)*, pages 68–72, 2000.

[14] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[15] V. Mezaris, I. Kompatsiaris, and M. Strintzis. An ontology approach to object-based image retrieval. In *ICIP 2003*, volume II, pages 511–514, 2003.

[16] R. Mojsilovic and B. Rogowitz. Capturing image semantics with low-level descriptors. In *Proc. of ICIP*, pages 18–21, 2001.

[17] M. Naphade, T. Kristjansson, B. Frey, and T. Huang. Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia systems. *Image Processing, International Conference on*, 3:536, 1998.

[18] P. Pala and S. Santini. Image retrieval by shape and texture. *Pattern Recognition*, 32:517–527, 1999.

[19] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[20] J. R. Smith and S. Chang. Local color and texture extraction and spatial query. In *Proc. of IEEE Int. Conf. on Image Processing*, volume 3, pages 1011–1014, Sep 1996.

[21] J. R. Smith and S. fu Chang. Tools and techniques for color image retrieval. In *IS&T/SPIE Proceedings, Storage & Retrieval for Image and Video Databases*, volume 2670, pages 426–437, 1996.

[22] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Transaction on Image Process*, 10(1):117–130, 2001.

[23] A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Trans. on Knowl. and Data Eng.*, 11(1):81–93, 1999.

[24] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4:260–268, 2002.

[25] X. S. Zhou and T. S. Huang. Cbir: from low-level features to high-level semantics. *Image and Video Communications and Processing 2000*, 3974(1):426–431, 2000.

[26] X. S. Zhou and T. S. Huang. Unifying keywords and visual contents in image retrieval. *IEEE MultiMedia*, 9(2):23–33, 2002.

[27] Y. Zhuang, X. Liu, and Y. Pan. Apply semantic template to support content-based image retrieval. In *Storage and Retrieval for Media Databases*, volume 3972, pages 442–449, 1999.

An Ontological Representation of Documents and Queries for Information Retrieval Systems

Mauro Dragoni
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013 Crema (CR), Italy
mauro.dragoni@unimi.it

Célia da Costa Pereira
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013 Crema (CR), Italy
celia.pereira@unimi.it

Andrea G.B. Tettamanzi
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

ABSTRACT

This paper presents a vector space model approach, for representing documents and queries, using concepts instead of terms and WordNet as a light ontology. This way, information overlap is reduced with respect to the classic semantic expansion techniques. Experiments undertaken on MuchMore benchmark showed the effectiveness of the approach.

1. INTRODUCTION

This paper presents an ontology-based approach for a conceptual representation of documents. Such an approach is inspired by a recently proposed idea presented in [9], and uses an adapted version of that method to standardize the representation of documents and queries. The proposed approach is somehow similar to the classic query expansion technique. However additional considerations have been taken into account and some improvements have been applied as explained below.

Query expansion is an approach used in Information Retrieval (IR) in order to improve the system's performance. It consists of the expansion of the content of the query by adding the terms that are semantical correlated with the original terms of the query [12]. Several works demonstrated the enhanced performance of IR systems that implement query expansion approaches [19] [3] [5]. However, the query expansion approach has to be used carefully because, as demonstrated in [8], expansion might degrade the performance of some individual queries. This is due to the fact that an incorrect choice of terms and concepts for the expansion task might harm the retrieval process by drifting it away from the optimal correct answer.

Document expansion applied to IR has been recently proposed in [2]. In that work a sub-tree approach has been implemented to represent concepts in documents and queries. However, when using a tree structure there is a redundancy of information because more general concepts may be represented implicitly by using only the leaf concepts they subsume. The smart idea behind the representation of documents by using concepts is that documents and queries may be represented in the same way. This way, the risk of omitting some related terms (as it may happen in the classical query expansion technique), is reduced. However, it is necessary to use a language resource that permits to cover a higher number of terms in order to avoid information loss.

This paper presents a new representation for documents and queries. The proposed approach exploits the structure of the well-known machine readable dictionary WordNet in order to reduce the redundancy of information generally contained in a concept-based document representation. The second improvement is the reduction of the computational time needed to compare documents and queries represented by using concepts. This representation has been applied to the ad-hoc retrieval problem. The approach has been evaluated on the MuchMore[1] Collection [4] and the results demonstrate its viability.

In Section 2 an overview of the environment in which ontology has been used is presented. Section 3 presents the tools used for this work. Section 4 illustrates the proposed approach to represent information, while Section 5 compares this approach with other two well-known approaches used in conceptual representation of documents. In Section 6 the results obtained from the test campaign are discussed. Finally, Section 7 concludes.

2. RELATED WORKS

An increasing number of recent information retrieval systems make use of ontologies to help the users clarify their information needs and come up with semantic representations of documents. Many ontology-based information retrieval systems and models have been proposed in the last decade. An interesting review on IR techniques based on ontologies is presented in [11], while in [16] the author studies the application of ontologies to a large-scale IR system for web purposes. A model for the exploitation of ontology-based knowledge bases is presented in [7]. The aim of this model is to improve search over large document repositories. The model includes an ontology-based scheme for the annotation of documents, and a retrieval model based on an adaptation of the classic vector-space model [15]. Another information retrieval system based on ontologies is presented in [14]. The authors propose an information retrieval system which has landmark information database that has hierarchical structures and semantic meanings of the features and

[1] http://muchmore.dfki.de

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

characteristics of the landmarks.

The implementation of ontology models has been also investigated by using fuzzy models [6].

In IR, the user's input queries usually are not detailed enough, so the satisfactory query results can not be brought back. Query expansion of IR can help to solve this problem. However, the common query expansion in IR cannot get steady retrieval results. Ontologies play a key role in query expansion research. A common use of ontologies in query expansion is to enrich the resources with some well-defined meaning to enhance the search capabilities of existing web searching systems.

In [18] the authors propose and implement query expansion method which combines domain ontology with the frequency of terms. Ontology is used to describe domain knowledge; logic reasoner and the frequency of terms are used to choose fitting expansion words. This way, higher recall and precision can be gotten as user's query results.

In [10] the authors present an approach to expand queries that consists in searching terms from the topic query in an ontology in order to add similar terms.

3. PRELIMINARIES

The roadmap to prove the viability of a concept-based representation of documents and queries consists in two main tasks:

- to choose a method that permits to represent all documents terms by using the same set of concepts;

- to implement an approach that permits to index and to evaluate each concept, in both documents and queries, with the appropriate weight.

To represent documents, the method described in Section 4 has been used, combined with the use of the WordNet machine-readable dictionary. From the WordNet database, the set of terms that do not have hyponymy has been extracted, each term is named "base concept". A vector, named "base vector", has been created and, to each component of the vector, a base concept has been assigned. This way, each term is represented by using the base vector of the WordNet ontology.

The representation described above has been implemented on top of the Apache Lucene open-source API. [2]

In the pre-indexing phase, each document has been converted in its ontological representation. After the calculation of the importance of each concept in a document, only concepts with a degree of importance higher than a fixed cut-value have been maintained, while the others have been discarded. The cut-value used in these experiments is 0.01. This choice has a drawback, namely that an approximation of representing information is introduced due to the discard of some minor concepts. However, we have experimentally verified that this approximation does not affect the final results.

During the evaluation activity, queries have been also converted into the ontological representation. This way, weights have to be assigned to each concept to evaluate all concepts with the right proportion. One of the features of Lucene is the possibility of assigning a payload to each term of the query. Therefore, to each element present in the concept-based representation of the query, its concept weight has been used as boost value.

[2]See URL `http://lucene.apache.org/`.

4. DOCUMENT REPRESENTATION

Conventional IR approaches represent documents as vectors of term weights. Such representations use a vector with one component for every significant term that occurs in the document. This has several limitations, for example:

1. different vector positions may be allocated to the synonyms of the same term; this way there is an information loss because the importance of a determinate *concept* is distributed among different vector components;

2. the size of a document vector have to be at least equal to the total number of words of the language used to write the document;

3. every time a new set of terms is introduced (which is a high-probability event), all document vectors must be reconstructed; the size of a repository thus grows not only as a function of the number of documents that it contains, but also of the size of the representation vectors.

To overcome these weaknesses of term-based representations, an ontology-based representation has been used [9].

An ontology-based representation has been recently proposed in [9] which exploits the hierarchical *is-a* relation among concepts, i.e., the meanings of words. For example, to describe with a term-based representation documents containing the three words: "animal", "dog", and "cat" a vector of three elements is needed; with an ontology-based representation, since "animal" subsumes both "dog" and "cat", it is possible to use a vector with only two elements, related to the "dog" and "cat" concepts, that can also implicitly contain the information given by the presence of the "animal" concept. Moreover, by defining an ontology base, which is a set of independent concepts that covers the whole ontology, an ontology-based representation allows the system to use fixed-size document vectors, consisting of one component per base concept.

Calculating term importance is a significant and fundamental aspect for representing documents in conventional information retrieval approaches. It is usually determined through term frequency-inverse document frequency (TF-IDF). When using an ontology-based representation, such usual definition of term-frequency cannot be applied because one does not operate by keywords, but by concepts. This is the reason why it has been adopted the document representation based on concepts proposed in [9], which is a concept-based adaptation of TF-IDF.

In this paper, an adaptation of the approach proposed in [9] is presented. The original approach was proposed for domain specific ontologies and does not always consider all the possible concepts in the considered ontology, in the sense that it assumes a cut at a given specificity level. Instead, the proposed approach has been adapted for more general purpose ontologies and it takes into account all independent concepts contained in the considered ontology. This way, information associated to each concept is more precise and the problem of choosing the suitable level to apply the cut is overcome.

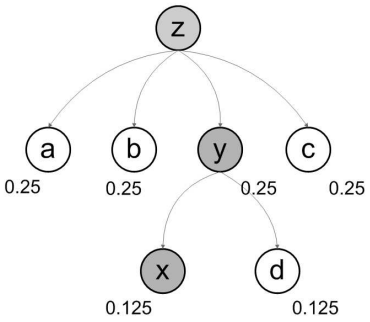


Figure 1: Ontology representation for concept 'z'.

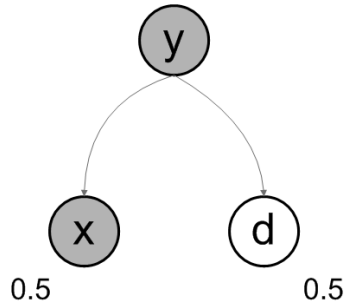


Figure 2: Ontology representation for concept 'y'.

The quantity of information given by the presence of concept z in a document depends on the depth of z in the ontology graph, on how many times it appears in the document, and how many times it occurs in the whole document repository. These two frequencies also depend on the number of concepts which subsume or are subsumed by z. Let us consider a concept x which is a descendant of another concept y which has q children including x. Concept y is a descendant of a concept z which has k children including y. Concept x is a leaf of the graph representing the used ontology. For instance, considering a document containing only "xy", the occurrence of x in the document is $1 + (1/q)$. In the document "xyz", the occurrence of x is $1 + (1/q(1 + 1/k))$. As it is possible to see, the number of occurrences of a leaf is proportional to the number of children which all of its ancestors have. Explicit and implicit concepts are taken into account by using the following formulas:

$$N(c) = \text{occ}(c) + \sum_{c \in \text{Path}(c,\ldots,\top)} \sum_{i=2}^{\text{depth}(c)} \frac{\text{occ}(c_i)}{\prod_{j=2}^{i} ||\text{children}(c_j)||}, \quad (1)$$

where $N(c)$ is the number of occurrences, both explicit and implicit, of concept c and $\text{occ}(c)$ is the number of lexicalizations of c occurring in the document. The value $N(c)$ is the weight associated with the concept c.

Given the ontology base $I = b_1, \ldots, b_n$, where the b_is are the base concepts, the quantity of information, $\text{info}(b_i)$, pertaining to base concept b_i in a document is:

$$\text{info}(b_i) = \frac{N_{\text{doc}}(b_i)}{N_{\text{rep}}(b_i)}, \quad (2)$$

where $N_{\text{doc}}(b_i)$ is the number of explicit and implicit occurrences of b_i in the document, and $N_{rep}(b_i)$ is the total number of its explicit and implicit occurrences in the whole document repository. This way, every component of the representation vector gives a value of the importance relation between a document and the relevant base concept.

A concrete example can be explained starting from the light ontology represented in Figures 1 and 2, and by considering a document D_1 containing concepts "$xxyyyz$".

In this case the ontology base is:

$$I = \{a, b, c, d, x\}$$

and, for each concept in the ontology, the vectors N_{doc} are:

$$\begin{aligned} z &= (0.25, 0.25, 0.25, 0.125, 0.125) \\ a &= (1.0, 0.0, 0.0, 0.0, 0.0) \\ b &= (0.0, 1.0, 0.0, 0.0, 0.0) \\ c &= (0.0, 0.0, 1.0, 0.0, 0.0) \\ y &= (0.0, 0.0, 0.0, 0.5, 0.5) \\ d &= (0.0, 0.0, 0.0, 1.0, 0.0) \\ x &= (0.0, 0.0, 0.0, 0.0, 1.0)\ , \end{aligned}$$

so the document vector associated to D_1 is:

$$D_1 = (2*\bar{x}) + (3*\bar{y}) + \bar{z} = (0.25, 0.25, 0.25, 1.625, 3.625). \quad (3)$$

In Section 5, a comparison between the proposed representation and other two classic concept-based representation is discussed.

5. REPRESENTATION COMPARISON

In Section 4 the approach used to represent information was described. This section shows the improvements obtained by applying the proposed approach and it illustrates a comparison between the proposed approach and other two approaches commonly used in conceptual document representation. The expansion technique is generally used to enrich information content of queries. However, in the last years some authors applied the expansion technique also to represent documents [2]. Like in [13] [2], we propose an approach that uses WordNet to extract concepts from terms.

The two main improvements obtained by the application of the ontology-based approach are illustrated below.

Information Redundancy.

Approaches that apply the expansion of documents and queries, use correlated concepts to expand the original terms of documents and queries. A problem with expansion is that information is redundant and there is not a real improvement of the representation of the document (or query) content. With the proposed representation this redundancy is eliminated because only independent concepts are taken into account to represent documents and queries. Another positive aspect is that the size of the vector representing document content by using concepts is generally lower than the size of the vector representing document content by using terms.

An example of technique that shows this drawback is presented in [13]. In this work the authors propose an indexing technique that takes into account WordNet synsets instead of terms. For each term in documents, the synsets associated to that terms are extracted and then used as token

for the indexing task. This way, the computational time needed to perform a query is not increased, however, there is a significant overlap of information because different synsets might be semantically correlated. An example is given by the terms "animal" and "pet", these terms have two different synsets, however, observing the WordNet lattice, the term "pet" is linked with an "is-a" relation with the term "animal". Therefore, in a scenario in which a document contains both terms, the same conceptual information is repeated. This is clear because, even if the terms "animal" and "pet" are not represented by using the same synset, they are semantically correlated because "pet" is a sub-concept of "animal". This way, when a document contains both terms, the presence of the term "animal" has to contribute to the importance of the concept "pet" instead of to be represented with a different token.

Computational Time.

When IR approaches are applied in a real-world environment, the computational time needed to evaluate the match between documents and the submitted query has to be considered. It is known that systems using the vector space model have higher efficiency. Conceptual-based approaches, such as the one presented in [2], generally implement a non-vectorial data structure which needs a higher computational time with respect to a vector space model representation. The approach proposed in this paper overcomes this issue because the document content is represented by using a vector and therefore, the computational time needed to compute document score is comparable to the computational time needed by using the vector space model.

6. EXPERIMENTS

In this section, the impact of the ontology document and query representation is evaluated. The evaluation method follows the TREC protocol [17]. For each query the first 1000 retrieved documents have been considered and the precision of the system has been calculated at different points: 5, 10, 15, and 30 documents retrieved. Moreover, the precision/recall graph has been calculated

The experimental campaign has been performed by using the MuchMore collection that consists of 7823 abstracts of medical papers and 25 queries with their relevance judgments. One of the particular features of this collection is that there are a lot of medical terms. This way, a term-based representation is more advantaged with respect to semantic representation, because specific terms present in documents (for example "Arthroscopic") are very discriminant. Indeed, by using a semantic expansion some problems may occur because, generally, the MRD and thesaurus used to expand terms do not contain some domain-specific terms.

The precision/recall graph showed in Figure 3 illustrates the comparison between the proposed approach (gray curve with circle marks), the classical term-based representation (black curve), and the synset representation method [13] (light gray curve with square marks). As expected, for all recall values, the proposed approach obtained better results than the term-based representation. The best gain of the concept-based representation is at recall levels 0.0, 0.2, and 0.4. While for recall values between 0.6 and 1.0, the concept-based precision curve lies with the other two curves.

A possible explanation for this scenario is that for documents that are well related to a particular topic the adopted

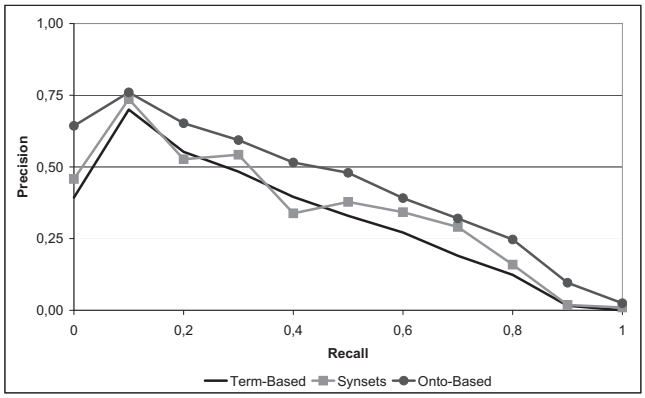


Figure 3: Precision/recall results.

ontology representation is able to improve the representation of the documents contents. However, for documents that are partially related to a topic or that contains many ambiguous terms, the proposed approach is not able to maintain an high precision of the results. At the end of this section some improvements that may be responsible of this fact are discussed.

In Table 1 the three different representations are compared for the Precision@X and MAP values. The results show that the proposed approach obtains better results for the all precision levels and also for the MAP value.

Systems	Precisions				
	P5	P10	P15	P30	MAP
Term-Based	0.544	0.480	0.405	0.273	0.449
Synset-Indexing [13]	0.648	0.484	0.403	0.309	0.459
Concept-Based	0.744	0.544	0.478	0.394	0.507

Table 1: Comparisons table between semantic expansion approaches.

An in-depth study of this first experiments campaign has been performed, and we have noticed that for some queries the concept-based representation obtained results that are below our expectations. By inspecting the implemented model, some issues have been noticed and are at now under analysis:

- Absence of some terms in the ontology: some terms, in particular terms related to specific domains (biomedical, mechanical, business, etc.), are not defined in the machine readable dictionary used to define the concept-based version of the documents. This way there is, in some cases, a loss of information that affects the final retrieval result.

- Proper names have not been considered: proper names of persons, geographical locations, industries, etc., are not present in the concept-based index. Observing the content of some documents and topics, proper names turn out to be a discriminant feature in some cases.

- Verbs and adjective are not present as well in the ontology: the concept representation of terms, described in Section 4, does not take into account verbs and adjectives.

This happens because verbs and adjectives are structured in a different way than nouns. The hyperonymy and hyponymy relations (that make MRD comparable with ontologies) are not defined for verbs and adjectives, therefore another approach will be studied and implemented to overcome this drawback.

- Term ambiguity: the concept-based representation has the problem of introducing an error given by not using a word sense disambiguation algorithm. Using such a method, concepts associated to incorrect senses would be discarded or weighted less. Therefore, the concept-based representation of each word would be finer, with the consequence of representing the information contained in a document with more precision.

Improving the actual model with the above features, would certainly yield significantly better results in the next experiments campaign. This positive view is motivated by the fact that, in spite of these issues, the preliminary goal of outperforming the precision of the term-based representation has been accomplished.

7. CONCLUSION

In this paper we have discussed an approach to index documents and to represent queries for information retrieval purposes which exploits a conceptual representation based on ontologies.

Experiments have been performed on the MuchMore Collection to validate the approach with respect to problems like term-synonymity in documents.

Preliminary experimental results show that the proposed representation improves the ranking of the documents. Investigation on results highlights that further improvement could be obtained by integrating WSD techniques like the one discussed in [1] to avoid the error introduced by considering incorrect word senses, and with a better usage and interpretation of WordNet to overcome the loss of information caused by the absence of proper nouns, verbs, and adjectives.

8. REFERENCES

[1] A. Azzini, M. Dragoni, C. da Costa Pereira, and A. Tettamanzi. Evolving neural networks for word sense disambiguation. In *Proc. of HIS '08, Barcelona, Spain, September 10-12*, pages 332–337, 2008.

[2] M. Baziz, M. Boughanem, G. Pasi, and H. Prade. An information retrieval driven by ontology: from query to document expansion. In D. Evans, S. Furui, and C. Soulé-Dupuy, editors, *RIAO*. CID, 2007.

[3] B. Billerbeck and J. Zobel. Techniques for efficient query expansion. In A. Apostolico and M. Melucci, editors, *SPIRE*, volume 3246 of *Lecture Notes in Computer Science*, pages 30–42. Springer, 2004.

[4] M. Boughanem, T. Dkaki, J. Mothe, and C. Soulé-Dupuy. Mercure at trec7. In *TREC*, pages 355–360, 1998.

[5] D. Cai, C. van Rijsbergen, and J. Jose. Automatic query expansion based on divergence. In *CIKM*, pages 419–426. ACM, 2001.

[6] S. Calegari and E. Sanchez. A fuzzy ontology-approach to improve semantic information retrieval. In F. Bobillo, P. da Costa, C. d'Amato, N. Fanizzi, F. Fung, T. Lukasiewicz, T. Martin, M. Nickles, Y. Peng, M. Pool, P. Smrz, and P. Vojtás, editors, *URSW*, volume 327 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[7] P. Castells, M. Fernández, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowl. Data Eng.*, 19(2):261–272, 2007.

[8] S. Cronen-Townsend, Y. Zhou, and W. Croft. A framework for selective query expansion. In D. Grossman, L. Gravano, C. Zhai, O. Herzog, and D. Evans, editors, *CIKM*, pages 236–237. ACM, 2004.

[9] C. da Costa Pereira and A. G. B. Tettamanzi. *Soft computing in ontologies and semantic Web*, chapter An ontology-based method for user model acquisition, pages 211–227. Studies in fuzziness and soft computing. Ed. Zongmin Ma, Springer, Berlin, 2006.

[10] M. Díaz-Galiano, M. G. Cumbreras, M. Martín-Valdivia, A. M. Ráez, and L. Ureña-López. Integrating mesh ontology to improve medical information retrieval. In *CLEF*, volume 5152 of *Lecture Notes in Computer Science*, pages 601–606. Springer, 2007.

[11] O. Dridi. Ontology-based information retrieval: Overview and new proposition. In O. Pastor, A. Flory, and J.-L. Cavarero, editors, *RCIS*, pages 421–426. IEEE, 2008.

[12] E. Efthimiadis. Query expansion. In M. Williams, editor, *Annual review of information science and technology*, pages Vol. 31, pp. 121Ű187. Information Today Inc, Medford NJ, 1996.

[13] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarrán. Indexing with wordnet synsets can improve text retrieval. *CoRR*, cmp-lg/9808002, 1998.

[14] T. Hattori, K. Hiramatsu, T. Okadome, B. Parsia, and E. Sirin. Ichigen-san: An ontology-based information retrieval system. In X. Zhou, J. Li, H. Shen, M. Kitsuregawa, and Y. Zhang, editors, *APWeb*, volume 3841 of *Lecture Notes in Computer Science*, pages 1197–1200. Springer, 2006.

[15] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[16] S. Tomassen. Research on ontology-driven information retrieval. In R. Meersman, Z. Tari, and P. Herrero, editors, *OTM Workshops (2)*, volume 4278 of *Lecture Notes in Computer Science*, pages 1460–1468. Springer, 2006.

[17] E. Voorhees and D. Harman. Overview of the sixth text retrieval conference (trec-6). In *TREC*, pages 1–24, 1997.

[18] F. Wu, G. Wu, and X. Fu. Design and implementation of ontology-based query expansion for information retrieval. In L. Xu, A. Tjoa, and S. Chaudhry, editors, *CONFENIS (1)*, volume 254 of *IFIP*, pages 293–298. Springer, 2007.

[19] J. Xu and W. Croft. Query expansion using local and global document analysis. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *SIGIR*, pages 4–11. ACM, 1996.

MOWIS: A system for building Multimedia Ontologies from Web Information Sources

Vincenzo Moscato, Antonio Penta, Fabio Persia, Antonio Picariello
University of Naples
Dipartimento di Informatica e Sistemistica
via Claudio 21, 80125, Naples
{vmoscato,a.penta,fabio.persia,picus}@unina.it

ABSTRACT

Defining ontologies within the multimedia domain still remains a challenging task, due to the complexity of multimedia data and the related associated knowledge. In this paper, we propose: i) a novel multimedia ontology model that combine both low level descriptors and high level semantic concepts; ii) an automatic construction of ontologies using the Flickrweb services, that provide images, tags, keywords and sometimes useful annotation describing both the content of an image and personal interesting information. Eventually, we describe an example of automatic ontology construction in a specific domain.

1. INTRODUCTION

Nowadays, a lot of repositories containing both multimedia and the related annotations or metadata are publicly available on the web. Such kind of information may be used for an automatically generation of multimedia knowledge, particularly suitable for a variety of applications, such as information retrieval, browsing, data mining and so on.

It is well known in the literature that despite the tons of papers produced about multimedia databases and knowledge representations, there is not yet an accepted solution to the problem of *how to represent, organize and manage multimedia data and the related semantics by means of a formal framework*.
Usually, a multimedia database is described by means of "flat" metadata, the most of the times using a predefined set of metadata (as in mpeg standard), or sometimes using small annotation in natural languages: such kind of structures are substantially inadequate to support complete retrieval by content of image documents.

It is the authors' opinion that there is still a great work to do with respect to the *intensional aspects* of a *multimedia ontology*:

- *what a multimedia ontology is*: is it a taxonomy, or a semantic network of metadata (tags, annotations)?
- *does a multimedia ontology support concrete data*: what is the role of rough data – image, video, audio data– if any?
- *what a multimedia semantic is*: how to define and capture the semantics of multimedia data?
- *how to build extensional ontologies*: once defined a suitable formal framework, can we automatically build the defined multimedia ontologies?

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

Throughout the rest of paper, we will try to give an answer to all the previous cited aspects; in particular the original contribution of this work is: (i) to propose a novel multimedia ontology framework, in particular related to the image domain; (ii) to propose a technique for building ontologies, that operates on large corpora of human annotated repositories, namely the Flickr [7] database, considering both low level image processing strategies and keywords and annotations produced by humans when they store the produced data.

We provide an algorithm for creating image ontology in a specific domain gathering together all this different information. We then provide an example of automatic construction of image ontology and a discussion of the encountered problems and the provided solutions. We concluded that the framework is promising and sufficiently scalable to different domains.

The remaind of paper is organized as follows. Section 2 outlines the related work related to the multimedia ontology topic. In Section 3 the process for building an Image Ontology is described. Section 4 details the system architecture with some implementation issues and a case study for our process is shown in Section 5. In Section 6 some discussions and conclusions are reported.

2. RELATED WORKS

In the last few years, several papers have been presented about multimedia systems based on knowledge models, image ontologies, fuzzy extension of ontology theories.

In almost all the works, multimedia ontologies are effectively used to perform *semantic annotation* of the media content by manually associating the terms of the ontology with the individual elements of the image or video domains [12], thus demonstrating that the use of ontologies can enhance classification precision and image retrieval performance.

Instead of creating a new ontology from the scratch, other approaches [3] extend *WordNet* to image specific concepts, using an annotated image corpus as an intermediate step to compute similarity between example images and images in the image collection.

For solving the uncertain reasoning problems, the theory of fuzzy ontologies is presented in several works, as an extension of ontologies with crisp concepts as in the paper [6], that presents a complete fuzzy framework for ontologies. While in [8], the authors introduce a description logic framework for the interpretation of image contents.

Multimedia semantic papers based on *MPEG-7* [9] are very interesting. The MPEG-7 framework consists of *Descriptors* (*Ds*) and *Descriptor Schemes* (*DSs*) that represent features for multimedia, and more complex structures grouping *Ds* and *DSs*, respectively.

In particular, the MPEG-7 standard includes tags that describe visual features (e.g. color), audio features (e.g. timbre), structure (e.g. moving regions and video segments), semantic (e.g. object and events), management (e.g. creator and format), collection organization (e.g. collections and models), summaries (e.g. hierarchies of key frames) and, even, user preferences (e.g. for search) of multimedia. In this way the standard includes descriptions of low-level media-specific features that can often be automatically extracted from the different media types.

Unfortunately, MPEG-7 is not currently suitable for describing top-level multimedia features, because: i) its XML Schema-based nature prevents the effective manipulation of descriptions and its use of URNs is cumbersome for the web; ii) it is not open to the web standards for representing knowledge.

Other efforts have been also done in order to translate the semantic of the standard in some knowledge representation languages [11]. All these methods perform a *one to one* translation of MPEG-7 types into OWL concepts and properties.

Finally, a very interesting work reported in [1] has been proposed in order to define a multimedia ontology. The authors try to define a novel multimedia ontology that takes into account the semantic of MPEG-7 standard. They started using some patterns derived from the foundational ontology *DOLCE* [10]. In particular they used two design patterns *Descriptions & Situations* (*D & S*) and *Ontology of Information Objects* (*OIO*). The obtained ontology already covers a very large part of the standard, while their modeling approach has the aim to offer even more possibilities for multimedia annotation than MPEG-7 since it is truly interoperable with existing web ontologies. This approach fits interoperability purposes, but some constraints on the image semantics are introduced.

3. BUILDING AN IMAGE ONTOLOGY

3.1 An Image Ontology Model

An ontology is usually referred as an "explicit specification of a conceptualization" which is, in turn, "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them".

Stressing its conceptual nature, an ontology may be considered as a *theory* used to represent relevant notion about domain modeling, a "domain" being classified in terms of concepts, relationships and constraint on them.

Let us consider the image domain: as usual in a given media, we detect symbols, objects and concepts; in a certain image we have a region of pixels (symbol) related to a portion of multimedia data; this region is an instance (object) of the certain concept.

In other words, we can detect concepts but we are not able to disambiguate among the instances without some specific knowledge.

A simplified version of the described vision process will consider only two main levels: *Low* and *High*. In fact, the knowledge associated to an image can be easily described at two different levels of analysis:

- *Low level* - the low-level descriptions of raw images;
- *High level* - general or domain-specific image content concepts that can be derivable or less from low-level ones.

It's the author's opinion that an image ontology has to take into account these specific characteristics, as captured by the following definition:

DEFINITION 1 (IMAGE ONTOLOGY). *An Image Ontology is a directed and labeled graph* $(\mathcal{V}, \mathcal{E}, \rho)$*, where:*

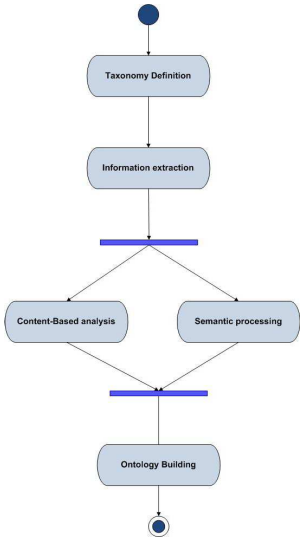


Figure 1: Image Ontology Building Process

1. $\mathcal{V}$ *is a finite set of nodes that can be of different kinds:*
   - low-level nodes *($\mathcal{V}_l$), corresponding to an image with the related properties:*
     - *content (e.g. texture, shape, color, objects, etc...) or more enhanced features;*
     - *metadata (e.g. author, title, description, tags, etc...);*
   - high-level nodes *($\mathcal{V}_h$), corresponding to* general concepts domain-specific concepts*, or* image content concepts *(that could be associated to low-level nodes);*
2. $\mathcal{E}$ *is a subset of* $(\mathcal{V} \times \mathcal{V})$*;*
3. ρ *is a function that associates to each couple of nodes a label indicating the kind of* relationship *between the two nodes* ρ_s*, and its reliability degree* $\rho_r \in [0, 1]$*:* $\rho : \mathcal{E} \rightarrow \langle \rho_s, \rho_r \rangle$.

Depending on the type of relationship in our model, we distinguish among:

- *similarity relationship*: relates between two low-level nodes (images) in function of their similarity degree, exploiting classical algorithms of image matching based on low-level features (e.g. color, texture, shape, etc...);
- *representativeness relationship*: relates between high-level and low-level nodes, containing those content features that better represents the associated concept, by means of clustering or classification algorithms that determine the probability that an image is a valid representative of the concept;
- *semantic relationship*: relates between two high-level nodes (example are those relationships such hypernym/hyponim, holonym/meronym, synonym, retrievable on lexical databases).

3.2 The Image Ontology building

The purpose of the image ontology building process (figure 1) is to perform the construction of the graph representing image ontology by a super-visioned approach.

The process is made of:

1. a definition of an initial taxonomy containing few high level nodes (related to the main concepts of a specific domain),

2. an extraction of useful information (images and annotations related to the taxonomy concepts) from several annotated web repositories,

3. a content-based analysis on the row-data and a semantic processing on the related textual annotations,

4. the ontology construction.

3.2.1 Taxonomy definition

Our image ontology building process is domain-oriented. Thus, during this step, it is necessary to define an initial taxonomy containing relevant concepts hierarchy of the considered domain that is represented by a subset of high level nodes.

3.2.2 Information extraction

The main objective of this task is to fetch images and the related textual annotations from web repositories, corresponding to the leaf high-level nodes of the image ontology, and to extract some useful low and high level information. Apposite communication API or web-services are exploited to obtain requested information.

In this paper we used Flickr as web image repository.

3.2.3 Content-Based analysis

The goal of such a task is to obtain a low-level description of images in terms of content features, using classical Computer Vision techniques.

We decided to use the salient points technique - based on the *Animate Vision* paradigm [2] - that exploits color, texture and shape information associated with those regions of the image that are relevant to human attention (*Focus of Attention*), in order to obtain a compact characterization (namely *Information Path*) that could be also used to evaluate the similarity between images, and for indexing issues.

An information path $\mathcal{IP}$=$\langle F_s(p_s;\tau_s)$,$h_b(F_s)$,$\Sigma_{F_s}\rangle$ can be seen as a particular data structure that contains, for each region $F(p_s;\tau_s)$ surrounding a given salient point (where p_s is the center of the region and τ_s is the the observation time spent by a human to detect the point), the color features in terms of HSV histogram $h_b(F_s)$, and the texture and shape features in terms of wavelet covariance signatures Σ_{F_s} (see [2] for more details).

Eventually, apposite super-visioned classification algorithms are exploited to determine content features [2].

3.2.4 Semantic processing

In this task the main objective is to discover textual *labels* that better reflect image semantic using NLP techniques and topic detection algorithms on the textual annotations coming from the considered image repositories. For what Flickr concerns, images usually have three main attached information: i) a *title*, ii) a content *description* and iii) a set of keywords, namely *tags*.

Titles in the majority of the cases contain text that summarizes the content of the images, while in other cases consist of automatically generated text that is not useful in the indexing process. *Descriptions* are very short and usually are not posted for retrieval purposes: they typically contain sentences concerning context of the picture, or user opinion. Finally, *Tags* are simple keywords users are asked (actually they may not insert any tag) to submit, that should describe the context of the image (e.g. among tags for a picture of an "elephant in an African landscape", you will probably see the words 'elephant', 'Africa' and 'landscape').

The simple use of tags does not improve the efficiency of indexing and searching contents. The absence of restrictions to the vocabulary from which tags are chosen can easily lead to the presence of *synonyms* (multiple tags for the same concept), *homonyms* (same tag used with different meaning), and *polysemies* (same tag with multiple related meanings). Also inaccurate or irrelevant tags result from the so called *'meta-noise'*, e.g. lack of stemming (normalization of word inflections), and from heterogeneity of users and contexts: hence an effective use of the tags requires these to be stemmed, disambiguated, and opportunely selected.

To these purposes, information coming from tags could be usefully analyzed in combination with titles and descriptions by suitable NLP technique that overcome the linguistic and semantic heterogeneity of such information, in order to extract a set of *relevant keywords* which more effectively represent image content.

In particular, the semantic processing, which is applied to the textual data attached to a given image can be decomposed into a set of sequential sub-tasks [13]: *meta-noise and named entity filtering*, *linguistic normalization*, *part of speech tagging*, *tokenization*, *word sense disambiguation* and *topic extraction*. Thus, the result of the semantic processing task is a set of *labels* (topics) with an associated *confidence* value - that represents the relative *importance* of the label (with respect to the other ones in the annotations) -, from the set of tags, title an descriptions.

3.2.5 Ontology building

As previously discussed, the obtained knowledge in terms of images, low-level characteristics and labels is then merged and translated in the shape of a graph representing image ontology.

In particular, in a first step, all images whose relevant labels are associated with a high confidence value to the high-level nodes, corresponding to the taxonomy leaves, will be represented by apposite low-level nodes; in addition, couple of image nodes, whose similarity (computed by means of the *Information Path Matching algorithm* [2]) is greater than a threshold will be linked by an edge having as reliability degree the related similarity measure.

In the successive step, previous images are clustered by used a *Balanced Expectation Maximazation* algorithm [2] applied in the feature spaces defined by the Information Path descriptors, in order to determine for the high-level nodes the set of images that better could represent the related concepts. Apposite edges (representative relationships) link such nodes with representatives of each cluster.

Eventually, by means of a *Learning Tag Relevance algorithm* [4], topics that are more relevant with respect to the content of images belonging to the same cluster (*winner topics*) are *promoted* to be image ontology high-level nodes. In particular, the tag relevance σ of a generic tag τ of the most significant image (*centroid*) of cluster C is computed by the following formula:

$$\sigma(\tau, C) = \sum_{i=1}^{m} |i_{df}(\tau) \cdot \frac{t_f(\tau.i) \cdot (a+1)}{t_f(\tau) + a \cdot (1 - b + b \cdot \frac{U_i}{\overline{U}})}| \quad (1)$$

where: $t_f(\tau, i)$ is the term frequency of topic τ with respect to the topics of all images belonging to C, $U_i, \overline{U}$ are the number of topics of $i-th$ image of C and the average number of tags related to all images belonging to C respectively, $i_{df}(\tau)$ is the inverse document frequency of τ in C. The winner topics, whose relevance is greater than a threshold, are finally inserted as high-level nodes in the ontology and *linked*, from one hand to the image node that corresponds to the cluster centroid and, from the other one, to those nodes which semantic distance (i.e. Wu/Palmer) is the minimum with respect to the current topic. If it is possible, the new ontology edge is labeled with the type of semantic relationship (e.g. hypernym/hyponim, holonym/meronym, etc...).

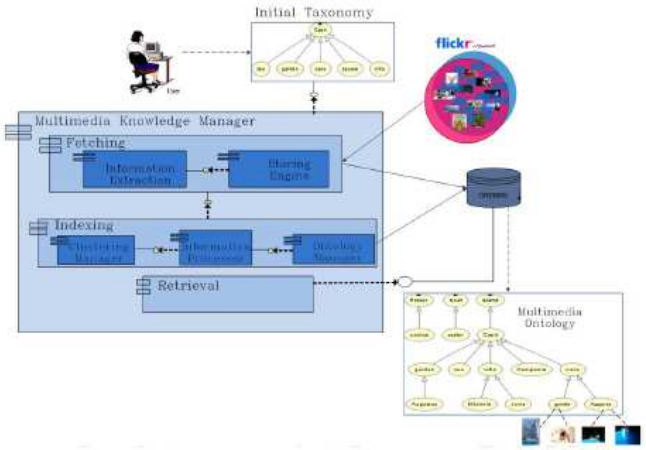


Figure 2: System Architecture

Thus, image ontology can be generated in an incremental way and in correspondence of pick-up operations from the Flickr repository.

4. THE SYSTEM ARCHITECTURE

The system architecture that supports the image ontology building process is shown in figure 2. User generates by an apposite graphical interface an *OWL* file coding the initial taxonomy containing relevant concepts of the considered domain. Such a file is then the input of the *Information Fetching* module that downloads images and the related annotations from the *Multimedia Repository*, using as search keywords the concepts related to leaf nodes of the taxonomy and some filters on users.

A *Storage Engine* module receives such information and stores image annotations (title, description, author, tags, labels, etc...) in a dedicated *RDF Database* and raw data together low-level characteristics in a *Image Database*. Each image is then identified in these databases by an *URI* (*Uniform Resource Identifier*).

Finally, the *Information Extraction* and *Information Processor* analyze both high level information stored into the RDF database and low level information contained into the Image database, in order to generate/update, by means of *Ontology Manager* and of *Clustering Manager*, and in according to the described process, a graph which represents final multimedia ontology.

For what implementation issues concerns, we notice that: (i) the initial taxonomy is generated by a JAVA desktop application that uses *Protégé* API; (ii) Flickr has been chosen as the multimedia repository; (iii) the Information Fetching module has been implemented as a JAVA application that exploits Flickr API; (iv) the RDF and Image Database have been realized by *Sesame* and *PostegreSQL* DBMS, respectively; (v) the Information Fetcing and Indexing packages have been implemented by apposite JAVA packages.

5. A CASE STUDY

This section describes a case study for our image ontology building process. In particular, we have built an ontology pertinent to *Capri*, a wonderful Italian island of the Sorrentine Peninsula, on the south side of the Gulf of Naples. A set of experts of natural and cultural attractions of Capri provided as initial taxonomy a graph reported containing the most relevant concepts in terms of high level nodes for the considered domain.

We used Flickr [7] as multimedia repository of annotated images. Flickr is one of the most popular web-based tagging system, that allows human participants to annotate a particular resource, such as web pages, blogs, images, with a freely chosen set of keywords, or tags, together with a short description of the content.

This kind of system has been recently termed *folksonomy* [5], i.e. a folk taxonomy of important and emerging concepts within user groups. The dynamic nature of these repositories assures the richness of the annotation; in addition, they are quite accurate, because they are produced by humans that want to share their images and the experience they have had, using tags and an annotation process.

The Flickr repository has been queried using as search keywords the *logical AND* between concepts reported in the leaf nodes of the taxonomy and the one corresponding to the root node and exploting some filters on user *id*s, in order to retrieve images really belonging to the domain. Each retrieved image undergoes a content-based analysis to determine the low-level description – i.e. the *IP* (*Information Path*) and content features. Moreover, in a first step we estimated similarity existing between each couple of different images by comparing their IPs by means of the *image path matching algorithm* [2].

All images belonging to the same concept are then clustered into different groups, which contain images that are more similar among themselves. We used as clustering procedure the *BEM algorithm* [2], that is recursively invoked to dynamically determine more fitting clusters without knowing a-priori the number of clusters themselves (that is usually proportional to number of images related to the current concept). Then we selected for each cluster the representative image as the closest one to all the other images of the cluster, and a suitable *representation probability* is associated to each representative image on the base of minimum and average distances.

The process is iterated for each taxonomy leaf concept and the ontology is incrementally built: images belonging to different topics could be linked on the base of their similarity values allowing to *merge* the multimedia knowledge in a unique graph. Thus, the more relevant tags are propagated in the ontology and linked to the other nodes.

We report in figure 3 a step by step complete example of the generation of *Capri* ontology.

6. CONCLUSION

In this paper we have addressed the problem of building a multimedia ontology in an automatic way using annotated image repositories. Our work differs from the previous papers presented in the literature for different reasons. First, we propose a notion of multimedia ontology, described by means of a graph and particularly suitable for managing the different levels of semantics of images. In addition, we obtain a dynamic generation of image ontologies using tags and annotations already produced by users in their social web networks.

Further works will be devoted to produce experimental results to evaluate the effectiveness of the produced ontologies with respect to other approaches by means of different criteria: *class match measure*, *density measure*, *semantic similarity measure*, *betweenness measure*.

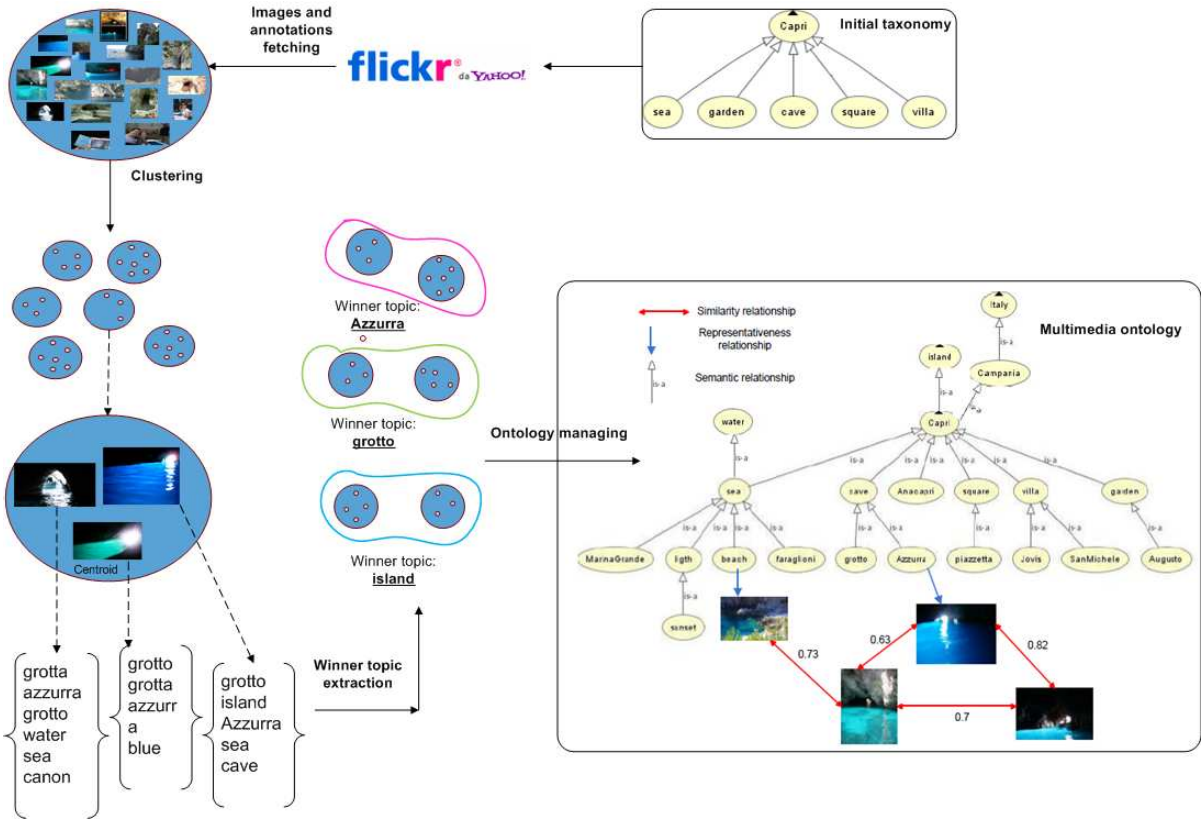


Figure 3: Bulding of the *Capri* Ontology

7. REFERENCES

[1] R. Arndt, R. Troncy, S. Staab, and L. Hardman. Adding formal semantics to mpeg-7: Designing a well-founded multimedia ontology for the web. Technical report, University of Koblenz, 2007.

[2] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello. Context-sensitive queries for image retrieval in digital libraries. *Journal of Intelligent Information Systems*, 31(1), 2008.

[3] Y. Chang and H. Chen. Approaches of using a word-image ontology and an annotated image corpus as intermedia for cross-language image retrieval. In *Proceedings of Cross-Language Evaluation Forum*, 2006.

[4] S. Golder and A. Hubemann. Usage patterns of collaborative tagging systems. *Information Science*, 2006.

[5] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. *ACM Multimedia*, 2007.

[6] C. Lee, Z. Jian, and L. Huang. A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man and Cybernetics,*, 35:859 – 880, 2005.

[7] K. Lerman and L. Jones. Social browsing on flickr. *CoRR*, abs/cs/0612047, 2006.

[8] R. Maller and B. Neumann. Ontology-based reasoning techniques for multimedia interpretation and retrieval. In Springer, editor, *Semantic Multimedia and Ontologies*, pages 55–98. Springer London, 2008.

[9] B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. 2002.

[10] C. Masolo and et al. The wonderweb library of foundational ontologies (wfol). Technical report, WonderWeb Deliverable 17, 2002.

[11] J. V. Ossenbruggen, F. Nack, and L. Hardman. That obscure object of desire: Multimedia metadata on the web, part 2. *IEEE Multimedia*, 12:54–63, 2005.

[12] G. Stamou and et al. Multimedia annotations on the semantic web. *Multimedia, IEEE*, 13:86 – 90, 2006.

[13] D. Trieschnigg and W. Kraaij. Tno hierarchical topic detection report at tdt 2004. *Proceedings of Corpus Linguistics 2005*, 99(7):1–8, 2004.

Natat in Cerebro: Intelligent Information Retrieval for "The Guillotine" Language Game *

Pierpaolo Basile
Dept. of Computer Science
University of Bari
via E. Orabona, 4
Bari, Italy
basilepp@di.uniba.it

Marco de Gemmis
Dept. of Computer Science
University of Bari
via E. Orabona, 4
Bari, Italy
degemmis@di.uniba.it

Pasquale Lops
Dept. of Computer Science
University of Bari
via E. Orabona, 4
Bari, Italy
lops@di.uniba.it

Giovanni Semeraro
Dept. of Computer Science
University of Bari
via E. Orabona, 4
Bari, Italy
semeraro@di.uniba.it

ABSTRACT

This paper describes OTTHO (On the Tip of my THOught), a system designed for solving a language game, called Guillotine. The rule of the game is simple: the player observes five words, generally unrelated to each other, and in one minute she has to provide a sixth word, semantically connected to the others. The system performs retrieval from several knowledge sources, such as a dictionary, a set of proverbs, and Wikipedia to realize a knowledge infusion process. The main motivation for designing an artificial player for Guillotine is the challenge of providing the machine with the cultural and linguistic background knowledge which makes it similar to a human being, with the ability of interpreting natural language documents and reasoning on their content. Our feeling is that the approach presented in this work has a great potential for other more practical applications besides solving a language game.

1. BACKGROUND AND MOTIVATION

Words are popular features of many games, and they play a central role in many language games. A *language game* is defined as a game involving natural language in which word meanings play an important role. Language games draw their challenge and excitement from the richness and ambiguity of natural language. In this paper we present a system that tries to play the *Guillotine* game. The Guillotine is a language game played in a show on RAI, the Italian National Broadcasting Service, in which a player is given a set of five words (clues), each linked in some way to a specific word that represents the unique solution of the game. She receives one word at a time, and must choose between two different proposed words: one is correct, the other one is wrong. Each time she chooses the wrong word, the prize money is divided by half (the reason for the name *Guillotine*). The five words are generally unrelated to each other, but each of them is strongly related to the word representing the solution. Once the five clues are given, the player has one minute to provide the solution. An example of the game follows: Given the five words *Capital, Pope, City, Colosseum, YellowAndRed*, the solution is *Rome*, because Rome is *Capital* of Italy, the *Pope* lives in Rome, Rome is a *City*, the *Colosseum* is in Rome and *YellowAndRed* is an alternative name for one of the Rome football teams. Often the solution is not so intuitive and the player needs different knowledge sources to reason and find the correct word.

OTTHO (On the Tip of my THOught) tries to solve the final stage of the *Guillotine* game. We assume that the five words are provided at the same time, neglecting the initial phase of choosing the words, that only concerns the reduction of the initial prize.

*The full version appears in [3]

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. OTTHO

Guillotine is a *cultural* and *linguistic* game, and for this reason we need to define an extended knowledge base for representing the *cultural* and *linguistic* background knowledge of the player. Next, we have to realize a reasoning mechanism able to retrieve the most appropriate *pieces of knowledge* necessary to solve the game.

2.1 The Knowledge Sources

After a deep analysis of the correlation between the clues and the solution, we chose to include the following knowledge sources, ranked according to the frequency with which they were helpful in finding the solution of the game:
1) Dictionary: the word representing the solution is contained in the description of a lemma or in some example phrases using that lemma;
2) Encyclopedia: as for the dictionary, the description of

an article contains the solution, but in this case it is necessary to process a more detailed description of information;
3) Proverbs and aphorisms: short pieces of text in which the solution is found very close to the clues.

These sources need to be organized and processed in order to model relationships between words. The modeling process must face the problem of the different characteristics of the several knowledge sources, resulting in a set of different heuristics for building the whole model on which to apply the reasoning mechanism. Since we are interested in finding relationships existing between words, we decided to model each knowledge source using the set of correlations existing between terms occurring in that specific source (a proverb, a definition in a dictionary, etc). Indeed, we used a *term-term matrix* containing terms occurring in the modeled knowledge source in which each cell contains the weight representing the degree of correlation between the term on the row and the one on the column. The computation of weights is different for each type of knowledge source.

For the dictionary, we used the on-line De Mauro Paravia Italian dictionary[1], containing 160,000 lemmas. We obtained a lemma-term matrix containing weights representing the relationship between a lemma and terms used to describe it. Because of the general lemma-definition organization of entries in the dictionary, we can fairly claim that the model is language-independent. Each Web page describing a lemma has been preprocessed in order to extract the most relevant information useful for computing weights in the matrix. The text of each Web page is processed in order to skip the HTML tags, even if the formatting information is preserved in order to give higher weights to terms formatted using bold or italic font. Stopwords are eliminated and abbreviations used in the definition of the lemma are expanded. Weights in the matrix are computed using a classical strategy based on a TF-IDF scheme, and normalized with respect to the length of the definition in which the term occurs and the length of the entire dictionary. A detailed description of the heuristics for modeling the dictionary is reported in [5].

As for the dictionary, a TF-IDF strategy has been used for defining the weights in the term-term matrix modeling the knowledge source of proverbs, a collection of $1,600$ proverbs gathered from the web[2].

The process of modeling Wikipedia is different from the one adopted for proverbs and dictionary, due to the huge amount of information to be processed. We adopted a more scalable approach for processing Wikipedia entries, by using models for representing concepts through vectors in a high dimensional space, such as the *Semantic Vectors* or `WordSpace` models [4]. The core idea behind semantic vectors is that words and concepts are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space (geometric metaphor of meaning). The basis of semantic vectors model is the theory of meaning called *distributional hypothesis*, according to which the meaning of a word is determined by the rules of its use in the context of ordinary and concrete language behavior. This means that words are semantically similar to the extent that they share *contexts* (surrounding words). If 'beer' and 'wine' frequently occur in the same context, say after 'drink', the hypothesis states that they are semantically related or similar.

[1] `http://old.demauroparavia.it/`
[2] `http://web.tiscali.it/proverbiitaliani` and `http://giavelli.interfree.it/proverbi_ita.html`

2.2 The Reasoning Mechanism

We adopt a *spreading activation model* [1], which has been used in other areas of Computer Science such as Information Retrieval [2] as reasoning mechanism for OTTHO. The pure spreading activation model consists of a network data structure of nodes interconnected by links, that may be labeled and/or weighted and usually have directions. In the network for "The Guillotine" game, nodes represent words, while links denote associations between words obtained from the knowledge sources. Spreading in the network is triggered by clues. The activation of clues causes words with related meanings (as modeled in the knowledge sources) to become active. At the end of the weight propagation process, the most "active" words represent good candidates to be the solution of the game.

3. BEYOND THE GAME

The system could be used for implementing an alternative paradigm for *associative retrieval* on collections of text documents [2], in which an initial indexing phase of documents can *spread* further "hidden" terms for retrieving other related documents. The identification of hidden terms might rely on the integration of specific pieces of knowledge relevant for the domain of interest. This might represent a valuable strategy for several domains, such as search engine advertising, in which customers' search terms (and interests) need to be matched with those of advertisers. Spreading activation can be also effectively combined with document retrieval for semantic desktop search.

4. REFERENCES

[1] A. M. Collins and E. F. Loftus. A Spreading Activation Theory of Semantic Processing. *Psychological Review*, 82(6):407–428, 1975.

[2] F. Crestani. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence*, 11(6):453–482, 1997.

[3] P. Lops, P. Basile, M. de Gemmis, and G. Semeraro. Language Is the Skin of My Thought": Integrating Wikipedia and AI to Support a Guillotine Player. In *AI*IA 2009*, LNCS 5883, pages 324–333. Springer, 2009.

[4] M. Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm University, Department of Linguistics, 2006.

[5] G. Semeraro and M. d. G. P. Lops, P. Basile. On the Tip of my Thought: Playing the Guillotine Game. In *IJCAI 2009*, pages 1543–154. Morgan Kaufmann, 2009.

Refreshing Models to Provide Timely Query Recommendations

Daniele Broccolo, Franco Maria Nardini, Raffaele Perego, Fabrizio Silvestri

ISTI - CNR
Pisa, Italy
{name.surname}@isti.cnr.it

ABSTRACT

In this work we propose a comparative study of the effects of a continuous model update on the effectiveness of well-known query recommendation algorithms. In their original formulation, these algorithms use static (i.e. pre-computed) models to generate recommendations. We extend these algorithms to generate suggestions using: a static model (no updates), a model updated periodically, and a model continuously updating (i.e. each time a query is submitted). We assess the results by previously proposed evaluation metrics and we show that the use of periodical and continuous updates of the model used for recommending queries provides better recommendations.

1. INTRODUCTION

The ocean of data on the web is continuously growing in size. Due to this reason web search engines are one of today's most used online applications to find what users need. According to Nielsen Online in October 2008 Google and Yahoo! answered more than 6 billions user searches in the US. In the latest years, web search engines have started to provide users with query recommendations to help them in refining queries and to quickly satisfy their needs. Query recommendation techniques are based on the knowledge about the behavior of past users of the search engine recorded in query logs. Basically, the behavior of many individuals is smarter than the behavior of few intelligent people.

We propose a new class of query recommender algorithms that we name "*incremental*" query recommender systems. These kind of systems update the model on which recommendations are drawn without the need for rebuilding it from scratch. That is, at regular intervals the recommender system updates the model on which suggestions are computed. In particular, we study a class of incremental recommenders where the model is updated for each received query. We study the effect on the performance (in terms of quality) of query recommender systems when varying the update interval. To do so, we propose an automatic evaluation mechanism to assess the effectiveness of query recommendation algorithms.

In this paper we aim at showing a novel class of query recommendation algorithms whose models are periodically updated as queries are submitted by users, and a comparison of four different query recommenders using new metrics. Due to space constraints we present a shortened version of our ongoing work.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. STATIC MODELS

To validate our hypothesis about the effects of continuous model updates on query recommender systems, we consider two well-known query recommendation algorithms and we define two new algorithms in order to continuously update the model on which recommendations are computed. The first one uses association rules for generating recommendations [3] (henceforth *AssociationRules*) while the second one uses click-though data [1] (henceforth *CoverGraph*). Hereinafter, we will refer to the original formulation of the two algorithms as "*static*", as opposed to the incremental version which will be called "*incremental*".

AssociationRules. Fonseca et al. uses association rules as a basis for generating recommendations [3]. The algorithm is based on two main phases. The first one uses query log analysis for session extraction, and the second one basically extracts association rules and identifies related queries. Each session is identified by all queries sent by an user in a specific time interval ($t = 10$ minutes). The problem of mining associations is to generate all the rules having a support greater than a specified minimum threshold (*minsup*). The rationale is that if distinct queries occurs simultaneously in many user sessions then those queries are considered to be related. Suggestions for a query q are simply computed by accessing the list of rules and by suggesting the q''s corresponding to rules with the higher support values.

CoverGraph. Baeza-Yates et al. use click-through data as a way to provide recommendations [1]. The method is based on the concept of *cover graph*. A *cover graph* is a bipartite graph of queries and URLs, where a query and an URL are connected if a user clicked in a URL that was an answer for a query. To catch the relations between queries, a graph is built out of a vectorial representation for queries. Each component of the vector is weighted according to the number of times the corresponding URL has been clicked on when returned for that query. Queries are then arranged as a graph with two queries being connected by an edge if and only if the two queries share a non-zero entry, that is if for two different queries the same URL received at least one click. Furthermore, edges are weighted according to the cosine similarity of the queries they connect. Suggestions for a query q are simply obtained by accessing the corresponding

node in the cover graph and extracting the queries at the end of the top scoring edges.

3. PERIODICALLY UPDATING MODELS

We argue that the use of "incremental" algorithms for query recommendation can provide better results from two main points of view: i) models age slowly (or do not age at all), and ii) they provide better recommendations for *bursty* topics [4]. However the trade offs are: the frequency of update (i.e. update frequency has to be tuned in order to maintain an high effectiveness of recommendations), and the computational cost for updating the model (the more frequently are the updates, the less responsive the recommender system). For these reasons to design a good incremental recommender algorithm is challenging.

We design two new recommender algorithms in order to allow the update of the models at regular time intervals. The two algorithms differ from the static versions by the way they manage and use data to build the model. To achieve the main challenges both the two algorithms implement LRU structures and use HashMaps to retrieve queries and links during the update phase. Doing so, a flexible, small and easy to maintain model is obtained.

4. EXPERIMENT

Assessing the effectiveness of recommender systems is a though problem. The evaluation can be made through *user-studies* and through automatic mechanisms.

We validate our proposals by means of an automatic evaluation methodology consisting in using previously proposed metrics. Due to space constraints, we show in the following experiments results with the *QueryOverlap* metric.

Let $S = \{q_1, \ldots, q_n\}$ be a user session of length n. Let $S_1 = \{q_1, \ldots, q_{\lfloor \frac{n}{2} \rfloor}\}$ be the set of queries in the first half of the session. For each $q_j \in S_1$, let $S_2 = \{q_{j+1}, \ldots, q_n\}$ be the $n - j$ most recently submitted queries in the session, and let $R_j = \{r_1, \ldots, r_m\}$ be the set of query recommendations returned for the query. We define *QueryOverlap* as:

$$QueryOverlap(q_j) = \frac{1}{K} \sum_{\substack{r_i \in R_j \\ s_k \in S_2}} [r_i = s_k] f(k)$$

where $[r_i = s_k]$ is 1 iff the i-th element of R is equal to the k-th element of S_2, and 0 otherwise. $f(k)$ is a weighting function allowing us to differentiate the importance of each recommendation depending on the position it occupies in the second part of the session and K is a normalization factor.

The most important experiments we have conducted is to measure the benefits of continuously updating models in query recommender systems. This test is conducted generating recommendations and assessing the effectiveness of query suggestions on different time slots. Here, we briefly discuss only results for the AssociationRules algorithm.

From the plots in Figure 1 it is evident that the effectiveness of the recommendations provided by both offline and online models becomes constant from a certain period of time. However the incremental versions (both quantized and continuously updating) produce sensitively better recommendations. This is due to the inclusion in the model of new and "fresher" data. Furthermore, except for an initial phase where the model is warming up, the number of useful suggestions of the continuously updating versions of

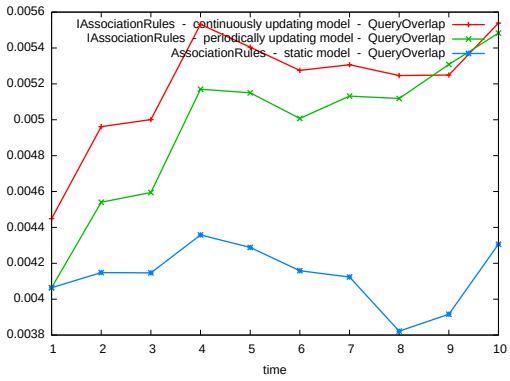


Figure 1: Comparing AssociationRules in two different implementations: static, and incremental (with periodical or continuous updates.

the algorithms is greater than the others throughout the entire observed period. The static and incrementally updating versions, indeed, produces more significant recommendations only in the very first intervals of the timeline. This is, obviously, due to both the "freshness" of the static models in the starting phases and to the cold-start problem in the continuously updating algorithms.

As a consequence of that, we prove that the aging effect on the models [2] affects the quality of the recommendations. "*Incremental*" algorithms provides a solution to this phenomenon.

5. CONCLUSIONS

In this work we propose a new class of query recommender algorithms that we name "*incremental*" query recommender systems. These kind of systems update the model on which recommendations are drawn, incrementally. In addition, we propose an automatic evaluation mechanism to assess the effectiveness of query recommendation algorithms.

Results show that continuously updating versions of the algorithms generate an higher number of useful suggestions with respect to the others throughout the entire observed period. This is a consequence of the aging effect on the models responsible for affecting the quality of the recommendations provided.

6. REFERENCES

[1] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *In Proc. KDD'07*, pages 76–85, New York, NY, USA, 2007. ACM.

[2] R. Baraglia, C. Castillo, D. Donato, F. M. Nardini, R. Perego, and F. Silvestri. Aging effects on query flow graphs for query suggestion. In *In Proc. CIKM'09.*, pages 1947–1950, New York, NY, USA, 2009. ACM.

[3] B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani. Using association rules to discover search engines related queries. In *In Proc. LA-WEB '03*, page 66, Washington, DC, USA, 2003. IEEE Computer Society.

[4] J. Kleinberg. Bursty and hierarchical structure in streams. In *In Proc. KDD'02*, pages 91–101, New York, NY, USA, 2002. ACM.

Yaanii: Effective Keyword Search over Semantic dataset

Roberto De Virgilio
Dipartimento di Informatica e Automazione
Universitá Roma Tre
Rome, Italy
devirgilio@dia.uniroma3.it

Paolo Cappellari
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
cappellari@ualberta.ca

Michele Miscione
Dipartimento di Informatica e Automazione
Universitá Roma Tre
Rome, Italy
miscione@dia.uniroma3.it

ABSTRACT

Nowadays data is disseminated in a number of different sources, from databases systems to the Web, from a traditional structured organization (relational) to a semi-structured (XML), up to the unstructured ones (text in Web documents). Although availability of data is constantly increasing, one principal difficulty users have to face is to find and retrieve the information they are looking for. To this aim keywords search based systems are increasingly capturing the attention of researchers. In this paper, we present Yaanii[1], a tool for the effective Keyword Search over semantic datasets. It is based on a novel keyword search paradigm for graph-structured data, focusing in particular on the RDF data model. While many techniques search the best answer trees, we propose an effective algorithm for the exploration and computation of all matching subgraphs. We provide a clustering technique that identifies and groups graph substructures based on template match. A scoring function, IR inspired, evaluates the relevance of the substructures and the clusters. A strong point of our approach is that the ranking supports the generation of Top-k solutions during its execution.

1. INTRODUCTION

Keyword-based search approaches have the huge benefit that users can ignore both the language and the structure of the data they are going to query. A keyword based search engine returns a list of candidate pages, documents or set of data that match keywords provided in input. Then a user has to dedicate time and efforts navigating each result returned from the engine in order to discover the desired information, i.e. the answer he is looking for. Therefore, attention around searching and query processing of graph-structured data continue to increase as the Web, XML documents and even relational database can be represented as a graph. Current approaches rely on a combination of IR and tree/graph exploration techniques whose goal is to rank results according to a relevance criterion. Keyword search on tree-structured data counts a good number of approaches already [4, 5]. Actual efforts [3, 6] focus on RDF data querying, given the great momentum of *Semantic Web* in which Web pages carry information that can be read and understood by machines in a systematic way. Simplifying, a generic approach first identifies the parts of the data structure containing the keywords of interest, possibly by using an indexing system or a database engine, then explores the data structure in order to discover a connection between such identified parts. The common exploration paradigm is similar to the triple of RDF, that is $\langle subject, property, object \rangle$. Candidate solutions, built out of found connections, are then all generated and finally ranked through a scoring function. To return the top-k best solutions, pruning techniques reducing the list of candidate solutions down to those whose score is above a threshold are implemented. In this framework to achieve efficiency above all, current algorithms compute the best answers only in an approximate way. This is because they use an exploration paradigm that is inefficient and the scoring function takes place only when solutions were generated all together. Moreover pruning techniques can have a sensible impact in both the quality of the solutions, as low scoring results are not shown or even computed, as well as on efficiency, as an early pruning reduces the space of candidate solutions to investigate.

In [2] we proposed a novel approach to keyword search in the graph-structure data in a RDF representation. The main contributions of our approach are:

- A clustering technique that identifies and groups graph substructures based on *template* match. The idea is to group paths with respect to the *template* (i.e schema) they correspond to. A solution is a composition of paths belonging to different clusters. In this way we avoid the exploration of overlapping solutions and we build cleaner results for the users, gaining in terms of computation cost. Usually, the most promising algorithms of an efficient solution for keyword based search are in PTIME class complexity. To this aim, in [1] we demonstrated how Yaanii is more efficient with respect to the others, presenting a quadratic complexity as upper-bound.
- An algorithm that ranks solutions while it builds the solutions. Unlike most of the approaches to keyword search, that first identify all the solutions and then rank them according to a function, our approach leverages on the clusters to assembly a solution starting with the most relevant path in the most relevant cluster. As a result, the most relevant solution is the first to come out of the algorithm, then decreasing monotonically to the less relevant solutions. This allows users to explore the returned solutions, starting with the most relevant, while the elaboration of remaining solutions is undergoing.

2. AN ARCHITECTURE OF REFERENCE

We implemented our approach into a tool, called Yaanii. A flexible architecture of the system was design, as shown in Figure 1.

[1] Yaanii, literally "path" in Sanskrit.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

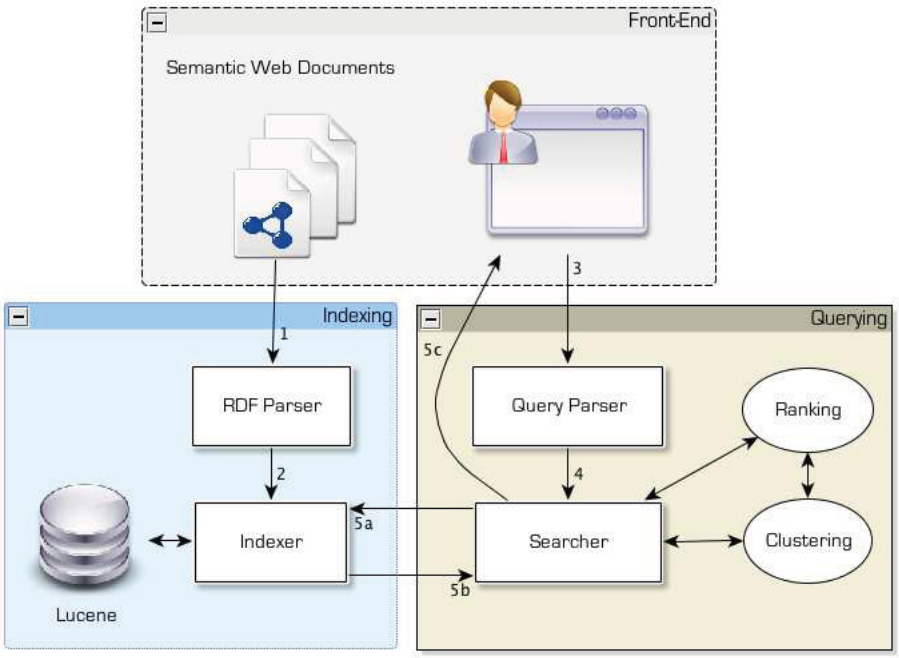


Figure 1: Architecture of Yaanii

It serves as a logical view of how the system looks like. This is a typical use scenario of the system:

1. The *RDF Parser* takes as input a collection of RDF Documents and parses them into triples. Here we use the Jena framework [2];

2. The *Indexer* builds an index on top of the triple collections to achieve structural information useful for the query process. Here the indexing is supported by Lucene[3] and WordNet [4]. The last allows query expansion;

3. A user performs a query through a *GUI helper*, handling events and the query itself;

4. The parsed query is given to the *Searcher* for processing;

5. The Searcher processes the query over the Indexed Resource Base and returns the search result to the caller. It communicates with the Indexer to extract the instances matching input keywords (i.e. informative paths), group them into clusters and compose elements from clusters into the final solutions (i.e. subgraph structures). Each structure (i.e. path, cluster and solution) is evaluated by a scoring function.

3. CONCLUSION AND FUTURE WORKS

We presented Yaanii, a tool for effective Keyword Search over semantic datasets. It is based on a clustering technique and a scoring function that support the generation of Top-k solutions during its execution in the first k steps.

From a theoretical point of view, future directions focus on improving the search algorithm of Yaanii to reach a linear time complexity. From a practical point of view, we would improve the indexing capabilities by embedding Lucene into a DBMS (e.g. Oracle) and provide a query-by-example interface to support the user to perform the query and navigate the results.

4. REFERENCES

[1] P. Cappellari, R. De Virgilio, M. Miscione. Keyword based Search over Semantic Data in Polynomial Time. In *Technical Report RT-DIA-160, Universitá Roma Tre, Rome, Italy*, 2009.

[2] R. De Virgilio, P. Cappellari, M. Miscione. Cluster-based exploration for Effective Keyword Search over Semantic Datasets. In *Proc. of the 28th International Conference on Conceptual Modeling (ER'09), Gramado, Brazil*, 2009.

[3] H. He, Wang, H., Yang, J., Yu, P.S. Blinks: ranked keyword searches on graphs. In *Int. Conf. on Management of Data (SIGMOD'07), China*, 2007.

[4] B. Kimelfeld, Sagiv, Y. Finding and approximating top-k answers in keyword proximity search. In *Int. Symposium on Principles of Database Systems (PODS'06), USA*, 2006.

[5] F. Liu, Yu, C.T., Meng, W., Chowdhury, A. Effective keyword search in relational databases. In *Int. Conf. on Management of Data (SIGMOD'06), USA*, 2006.

[6] T. Tran, Wang, H., Rudolph, S., Cimiano, P. Top-k exploration of query graph candidates for efficient keyword search on rdf. In *Int. Conf. on Data Engineering (ICDE'09), China*, 2009.

[2] http://jena.sourceforge.net/

[3] http://lucene.apache.org/

[4] http://wordnet.princeton.edu

Serendipitous Encounters along Dynamically Personalized Museum Tours*

Leo Iaquinta
Universitá degli Studi di Bari,
Dipartimento di Informatica
via E. Orabona, 4
Bari, Italy
iaquinta@di.uniba.it

Marco de Gemmis
Universitá degli Studi di Bari,
Dipartimento di Informatica
via E. Orabona, 4
Bari, Italy
degemmis@di.uniba.it

Pasquale Lops
Universitá degli Studi di Bari,
Dipartimento di Informatica
via E. Orabona, 4
Bari, Italy
lops@di.uniba.it

Giovanni Semeraro
Universitá degli Studi di Bari,
Dipartimento di Informatica
via E. Orabona, 4
Bari, Italy
semeraro@di.uniba.it

Piero Molino
Universitá degli Studi di Bari,
Dipartimento di Informatica
via E. Orabona, 4
Bari, Italy
piero.molino@gmail.com

ABSTRACT

Today Recommender Systems (RSs) are commonly used with various purposes, especially dealing with e-commerce and information filtering tools. Content-based RSs rely on the concept of similarity between items. It is a common belief that the user is interested in what is similar to what she has already bought/searched/visited. We believe that there are some contexts in which this assumption is wrong: it is the case of acquiring unsearched but still useful items or pieces of information. This is called serendipity. Our purpose is to stimulate users and facilitate these serendipitous encounters to happen. The paper presents a hybrid recommender system that joins a content-based approach and serendipitous heuristics in order to provide also surprising suggestions. The reference scenario concerns with personalized tours in a museum and serendipitous items are introduced by slight diversions on the context-aware tours.

1. BACKGROUND AND MOTIVATION

RSs allow a customized information access for targeted domains. They provide the users with personalized advices based on their needs, preferences and usage patterns. Sometimes RSs can only recommend items that score highly against the user's profile and, consequently, the user is limited to obtain advices only about items too similar to those she already knows. This drawback is referred as *over-specialization* and it prevents surprising finding from taking place. Indeed, the RSs are required to provide novel and even serendipitous advices. As explained by Herlocker [2], novelty occurs when the system suggests an unknown item that the user might have autonomously discovered. A serendipitous recommendation helps the user to find a surprisingly interesting item that she might not have otherwise discovered (or it would have been really hard to discover).

The idea of serendipity has a link with de Bono's "lateral thinking" [1] which consists not to think in a selective and sequential way, but accepting accidental aspects, that seem not to have relevance or simply are not sought for. This kind of behavior helps the awareness of serendipitous events, especially when the user is allowed to explore alternatives to satisfy her curiosity. Therefore the demonstrative scenario concerns personalized tours within a museum. Indeed, in addition to the "classical" recommendations that exploit the learned user profile, the system provides also programmatically supposed serendipitous recommendations and it arranges the whole of them in a personalized tour.

The serendipitous suggested items are selected exploiting the learned user profile so that they cause slight diversions on the personalized tour. Indeed the content-base recommender module allows to infer the most interesting items for the active user and a personalized tour is proposed according to the spatial layout, the user behavior and the time constraint. But the resulting tour potentially suffers from over-specialization and, consequently, some items can be found no so interesting for the user. Therefore the user starts to divert from suggested path considering other items along the path with growing attention. On the other hand, also when the recommended items are actually interesting for the user, she does not move with blinkers, i.e. she does not stop from seeing artworks along the suggested path. These are opportunities for serendipitous encounters. These considerations suggest to perturb the optimal path with items that are programmatically supposed to be serendipitous for the active user. Perturbing the optimal path with slight diversions does not compromise the system benefit to guide the user across the museum under a time constraint because the user behavior is constantly monitored and personalized tour

*The full version will appear in A. Lazinica (editor), *E-Commerce*, ISBN 978-953-7619-X-X, electronic version freely available at http://intechweb.org.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

eventually updated.

2. SERENDIPITOUS RECOMMENDATIONS

Toms [4] suggests four strategies to introduce the serendipity: 1) Role of chance or 'blind luck', implemented via a random information node generator; 2) Pasteur principle ("chance favors the prepared mind"), implemented via a user profile; 3) Anomalies and exceptions, partially implemented via poor similarity measures; 4) Reasoning by analogy, whose implementation is currently unknown.

In [3] we propose an architecture for content-based RSs that implements the "Anomalies and exceptions" approach to provide serendipitous recommendations alongside classical ones. The basic assumption is that serendipity cannot happen if the user already knows recommended items, because a serendipitous happening is by definition something new. Thus the lower is the probability that user knows an item, the higher is the probability that a specific item could result in a serendipitous recommendation. The probability that user knows something semantically near to what the system is confident she knows is higher than the probability of something semantically far. If we evaluate semantic distance with a similarity metric, like internal product which takes into account the item description to build a vector and compares it to other item vectors, it results that it is more probable to get a serendipitous recommendation providing the user with something less similar to her profile.

According to this idea, items should not be recommended if they are too similar to something the user has already seen. Following this principle, the basic idea underlying the proposed architecture is to ground the search for potentially "serendipitous" items on the similarity between the item descriptions and the user profile.

3. PERSONALIZED MUSEUM TOURS

RSs traditionally provide a static ordered list of items according to the user assessed interests, but they do not rely on the user interaction with environment. Besides, if the suggested tour simply consists of the enumeration of ranked items, the path is too tortuous and with repetitive passages that make the user disoriented, especially under a time constraint. Fig. 1 shows a sample tour consisting of the k most interesting items, where the k value depends on how long should be the personalized tour, e.g., it deals with the overall time constraint and the user behavior. Finally, different users interact with environment in different manner, e.g. they travel with different speed, they spend different time to admire artworks, they divert from the suggested tour. Consequently, the suggested personalized tour must be dynamically updated and optimized according to contextual information on user interaction with environment.

Once the personalized tour is achieved, as shown in Fig. 2, serendipitous disturbs are applied. Indeed, the previous personalized tour is augmented with some items that are along the path and that are in the ranked list of serendipitous items according to the learned user profile. The resulting path most likely has a worse fitness value and then a further optimization step is performed. However, the further optimization step should cut away exactly the disturbing serendipitous items, since they compete with items that are more similar with the user tastes. Therefore serendipitous items are differently weighed from the fitness function: their supposed stay time is changed. This implementation expedient also deals with the supposed serendipitous items should turn out not so serendipitous and the user should reduce the actual stay time in front of such items. Fig. 3 shows a "good enough" personalized tour consisting of the most interesting items and the most serendipitous ones. It is amazing to note that some selected serendipitous items are placed in rooms otherwise unvisited. More details and an empirical evaluation about serendipitous perturbations effects are presented in [3].

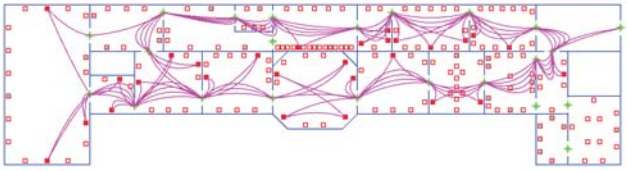


Figure 1: A sample tour consisting of the ranked k most interesting items

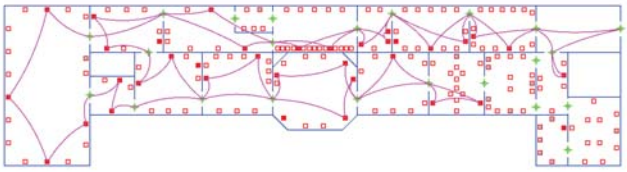


Figure 2: Optimized version of the tour in Fig. 1

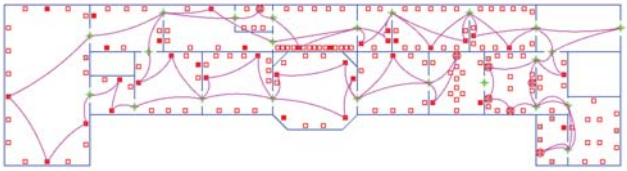


Figure 3: The "good enough" augmented version

4. REFERENCES

[1] E. De Bono. *Lateral Thinking: A Textbook of Creativity*. Penguin Books, London, 1990.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[3] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, and P. Molino. Can a recommender system induce serendipitous encounters? In A. Lazinica, editor, *E-Commerce*, pages 1–17. IN-TECH, Vienna, 2009.

[4] E. G. Toms. Serendipitous information retrieval. In *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, Zurich, 2000.

Manuzio: An Object Language for Annotated Text Collections

Marek Maurizio
Università "Ca Foscari" di Venezia
Via Torino 155
Venezia Mestre, Italy
marek@dsi.unive.it

Renzo Orsini
Università "Ca Foscari" di Venezia
Via Torino 155
Venezia Mestre, Italy
orsini@dsi.unive.it

1. INTRODUCTION

More and more large repositories of texts which must be automatically processed represent their content through the use of descriptive markup languages. This method has been diffused by the availability of widely adopted standards like SGML and, later, XML, which made possible the definition of specific formats for many kinds of text, from literary texts (TEI) to web pages (XHTML). The markup approach has, however, several noteworthy shortcomings. First, we can encode easily only texts with a hierarchical structure, then extra-textual information, like metadata, can be tied only to the same structure of the text and must be expressed as strings of the markup language. Third, queries and programs for the retrieval and processing of text must be expressed in terms of languages like XQuery [4]. In the XQuery data model, every document is represented as a tree of nodes; for this reason, in documents where parallel, overlapping structures exists, the complexity of XQuery programs becomes significantly higher.

Consider, for instance, a collection of classical lyrics, with two parallel hierarchies lyric > stanzas > verses > words, and lyric > sentences > words, with title and information about the author for each lyric, and where the text is annotated both with commentary made by different scholars, and with grammatical categories in form of tree-structured data. Such a collection, if represented with markup techniques, would be very complex to create, manage and use, even with sophisticated tools, requiring the development of complex ad-hoc software.

To overcome the above limitations due to the use of markup languages partial solutions exist (see for instance [3]), but at the expense of greatly increasing the complexity of the representation. Moreover, markup query languages need to be extended to take these solutions into consideration [1], making even more difficult to access and use such textual collections.

In the project "Musisque deoque II. Un archivio digitale dinamico di poesia latina, dalle origini al Rinascimento italiano", sponsored by the Italian MIUR, we have built a model and a language to represent repositories of literary texts with any kind of structure, with multiple and scalable annotations, not limited to textual data, and with a query component useful not only for the retrieval of information, but also for the construction of complex textual analysis applications. This approach fully departs from the markup principles, borrowing many ideas from the object-oriented models currently used in programming languages and database areas. A comprehensive description of the model, language, and system can be found in [5, 6]. The language (called Manuzio) has been developed to be used in a multi-user system to store persistently digital collections of texts over which queries and programs are evaluated. This abstract reports mainly the work done on the model and the language, since the system is still at its early stages of development with a prototypal implementation.

Appears in the Proceedings of the 1st Italian Information Retrieval Workshop (IIR'10), January 27–28, 2010, Padova, Italy.
http://ims.dei.unipd.it/websites/iir10/index.html

2. THE MANUZIO MODEL

The Manuzio model considers the textual information in a dual way: as a formatted sequence of characters, as well as a composition of logical structures called *textual objects*, similar to the content objects described in [2]. A *textual object* is a software entity with a state and a behavior. The state defines the precise portion of the text represented by the object, called the *underlying text*, and a set of *properties*, which are either *component* textual objects or *attributes* that can assume values of arbitrary complexity. The behavior is constituted by a collection of local procedures, called *methods*, which define computed properties or perform operations on the object. A textual object T is a *component* of a textual object T' if and only if the underlying text of T is a subtext of the underlying text of T'[1].

The Manuzio model can also represent aggregation of textual objects called *repeated textual objects*. Trough repeated textual objects it is possible to represent complex collections like "all the first words of each poem" or "all the first sentences of the abstracts of each article" in a simple and clean way. A repeated textual object is either a special object, called the *empty textual object*, or a set of textual objects of the same type, called its *elements*. Its underlying text is the composition of the underlying text of its elements.

Each textual object has a type, which represents a logical entity of the text, such as a word, a paragraph, a sentence, and so on. In the Manuzio model types are organized as a lattice where the greatest element represents the type of

[1]Differently from a substring, a subtext can comprise non-contiguous parts of a text.

the whole collection, and the least is the type of the most basic objects of the schema. Types can also be defined by inheritance, like in object-oriented languages. For instance, the types Novel and Poem are both subtypes of Work . An example of textual schema is given, by the means of a graphical notation, in Figure 1.

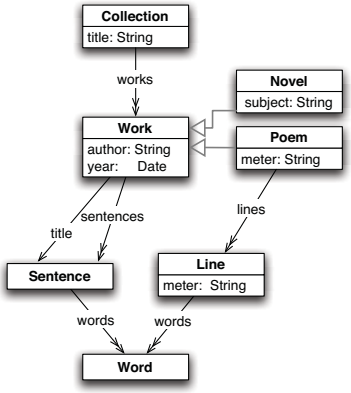


Figure 1: Example of Manuzio Model.

3. THE MANUZIO LANGUAGE

Manuzio is a functional, type-safe programming language with specific constructs to interact with persistently stored textual objects. The language has a type system with which to describe schemas as that illustrated in Figure 1, and a set of operators which can retrieve textual objects without using any external query language. A persistent collection of documents can be imported in a program and its root element can be referenced by a special variable `collection` of type `Collection`. From this value all the textual objects present in the collection can be retrieved through operators that exploit their type's structure: the *get* operator retrieve a specific component of an object, while the *all* operator retrieve recursively all the components and subcomponents of a certain type of an object. Other operators allow the creation of expressions similar to SQL or XQuery FLOWR expressions[2]. Since the queries are an integrated part of the language, they are subject to type-checking and can be used in conjunction with all the other language's features transparently.

The program in Source Code 1, for instance, assigns to a variable the first three sentences of each work of the collection. This portion of text can be subsequently refined or used in any retrieval context. In Source Code 2 a more complex example is shown, where an analysis of Shakespeare's plays extracts the top three "love speaking" characters in "A Midsummer Night's Dream". The results of such code are reported in Source Code 3.

```
let most_relevant_sentences =
        select all SENTENCE 1..3 of works of collection;
```

Source Code 1: Retrieve the most relevant sentences of each work.

[2]The full syntax and semantics of the Manuzio language can be found in [6].

```
let play =
    p in (get plays of collection)
    where p.title = "A Midsummer Night's Dream";
let loveSpeeches =
    s in (getall Speech of play)
    where some w in (getall Word of s)
    with (get stem of w) = "love";
let love_speech_count_by_speaker =
    select {speaker = s.speaker, n=(size of s.partition)}
    from s in (speeches groupby speaker);
output "The top 3 love spekaers are:";
output love_speech_count_by_speaker[1..3];
```

Source Code 2: Compute a new structure of the most love-speaking characters.

```
The top 3 love speakers are:
[{speaker="LYSANDER", n=17},
 {speaker="OBERON", n=13},
 {speaker="HERMIA", n=12}]
```

Source Code 3: Results of Source Code 2.

4. CONCLUSIONS AND FUTURE WORK

To evaluate the usefulness of our approach a first prototype of the Manuzio language has been developed by mapping the textual objects onto a relational database system. We are aware that a great deal of work on data representation and query optimization must yet be done to provide a satisfying performance for large collections of texts. However, we think that work on modeling and linguistics aspects of retrieval of texts and computations over them is very important, and prerequisite to enrich the solutions offered by research areas such as information retrieval and digital libraries. In particular, the possibility of taking into account structural information when making queries (for instance, by considering terms in titles, or excluding those in footnotes) could improve notably the quality of their results.

5. REFERENCES

[1] Alex Dekhtyar, Ionut E. Iacob, Kevin Kiernan, and Dorothy C. Porter. Extended xquery for digital libraries. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 378–378, New York, NY, USA, 2006. ACM.

[2] S.J. DeRose, D.G. Durand, E. Mylonas, and A.H. Renear. What is text, really? *ACM SIGDOC Asterisk Journal of Computer Documentation*, 21(3):1–24, 1997.

[3] Steven J. DeRose. Markup overlap: A review and a horse. In *Extreme Markup Languages*, 2004.

[4] H. Katz and D.D. Chamberlin. *XQuery from the experts: a guide to the W3C XML query language*. Addison-Wesley Professional, 2004.

[5] Renzo Orsini Marek Maurizio. A model and query language for literary texts. Technical Report CS-2009-4, Dipartimento di Informatica, Università Ca' Foscari di Venezia, 2009.

[6] Marek Maurizio. *Manuzio: an Object Language for Annotated Text Collections*. PhD thesis, Dipartimento di Informatica, Università Ca' Foscari di Venezia, 2009.

Author Index