Elena Simperl, Devika P. Madalli,
Denny Vrandecic, Enrique Alfonseca (Eds.)

DiversiWeb 2011

Proceedings of the 1st International Workshop on

Knowledge Diversity on the Web

Workshop at the
20th International World Wide Web Conference (WWW 2011)

Hyderabad, India
March, 2011

Organization committee

- Elena Simperl, AIFB, KIT, Germany
- Devika P. Madalli, Indian Statistical Institute, Bangalore, India
- Denny Vrandecic, AIFB, KIT, Germany
- Enrique Alfonseca, Google Zurich, Switzerland

Programm Committee

- Anthony Baldry, University of Pavia, Italy
- Jean-Yves Delort, Google, Switzerland
- Elena Demidova, L3S, Germany
- Reto Krummenacher, STI Innsbruck, Austria
- Paul Lewis, University of Southampton, UK
- Vincenzo Maltese, University of Trento, Italy
- Kunal Patel, Ingenuity Systems, USA
- Delia Rusu, JSI, Slovenia
- Katharina Siorpaes, STI Innsbruck, Austria
- Markus Strohmaier, TU Graz, Austria
- Thanh Duc Tran, KIT, Germany
- Mitja Trampus, JSI, Slovenia
- Andreas Wagner, KIT, Germany

Preface

DiversiWeb 2011,[1] the First International Workshop on Knowledge Diversity on the Web, co-located with WWW2011 in Hyderabad, India, provided an interdisciplinary forum for researchers and practitioners to present and discuss their ideas related to the challenges posed by diversity on the Web. We addressed a wide array of interdisciplinary questions, which need to be tackled in order to preserve the fragile balance between a world that is continually converging and growing together, the rich diversity of the global society, and the dangers of fragmentation and splintering.

The workshop was partially funded by EU ICT FP7 projects RENDER[2] and Living Knowledge[3]. We thank all authors for submitting to DiversiWeb 2011, the program committee for the indispensable help in the selection process for the workshop program, and VideoLectures.net for recording the workshop[4].

Elena Simperl, Devika P. Madalli, Denny Vrandecic, Enrique Alfonseca

DiversiWeb 2011 Organization Committee

[1] http://render-project.eu/diversiweb2011/
[2] http://render-project.eu/
[3] http://livingknowledge-project.eu/
[4] http://videolectures.net/diversiweb2011_hyderabad/

Table of Contents

Towards a Knowledge Diversity Model

Rakebul Hasan, Katharina Siorpaes,
Reto Krummenacher
Semantic Technology Institute (STI)
University of Innsbruck
A-6020 Innsbruck, Austria
firstname.lastname@sti2.at

Fabian Flöck
Institute of Applied Informatics and Formal
Description Methods
Karlsruhe Institute of Technology
D-76131 Karlsruhe, Germany
fabian.floeck@kit.edu

ABSTRACT

The Web is an unprecedented enabler for publishing, using and exchanging information at global scale. Virtually any topic is covered by an amazing diversity of opinions, viewpoints, mind sets and backgrounds. The research project RENDER works on methods and techniques to leverage diversity as a crucial source of innovation and creativity, and designs novel algorithms that exploits diversity for ranking, aggregating and presenting Web content. Essential in this respect is a knowledge model that makes accessible — cognitively to human users as well as computationally to the machine — the diversity in content. In this paper, we present a glossary of relevant terms that serves as baseline to the specification of the Knowledge Diversity Model.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introductory and Survey; I.2.4 [**Computing Methodologies**]: Artificial Intelligence—*Knowledge Representation Formalisms and Methods*

Keywords

Knowledge diversity, Glossary, Knowledge model

1. INTRODUCTION

The Web is a tremendous facilitator and catalyst for the publication, use and exchange of information, fostering a global network of news, stories and statements which represent an amazing diversity of opinions, viewpoints, mind sets and backgrounds. Its design principles and core technology have led to an unprecedented growth in mass collaboration; a trend that is also increasingly impacting business environments.

The RENDER project[1] aims at leveraging the diversity inherently unfolding through world wide scale publishing and collaboration by developing methods, techniques, software and data sets that make diversity accessible as an important source of innovation and creativity, and by designing novel algorithms that reflect diversity in the ways information is selected, ranked, aggregated, presented and used.

An important component for the capturing of diversity in online documents, is a comprehensive knowledge model for representing diversity that reflects the plurality of opinions and viewpoints on a particular topic. In a first step, the considered content such as articles, blog entries or news feeds are transformed into a semantic representation according to the knowledge model that is accessible both cognitively to human users as well as computationally to the machine. The semantic representation is then leveraged for improving the selection and ranking of content, and the presentation to users. In RENDER, selection and ranking will go beyond widely adopted approaches based on popularity or personalization, and take opinions and viewpoints into account when computing the relevance of results.

In this paper we present a glossary of terms relevant in the scope of knowledge diversity. Creating a shared understanding of terms and relationships between terms is an essential first step towards the specification of a conceptual model for knowledge diversity. In that sense, this paper provides the necessary baseline for the definition of a knowledge diversity ontology, which allows for formalizing, gathering, evaluating and processing diversity in various (written) online medias.

In a first section (Section 2) we provide three motivating scenarios for this work, which are derived from the project's showcases that are brought to RENDER by Google, Wikimedia, and Telefonica. Section 3 provides a glossary of terms such as diversity, opinion, sentiment, bias and many more. Section 4 presents a short overview of the related work. In Section 5 we take a quick look at next steps, at how the targeted knowledge model will be used and leveraged in the given scenarios and throughout the project, and conclude the paper.

2. MOTIVATING SCENARIOS

In the following we present three motivating business scenarios for the formalization of a knowledge diversity model.

2.1 Wikipedia

Despite efforts for a balanced coverage at Wikipedia, systemic biases influenced by the individual views of the more than 100'000 volunteer contributors have been introduced. The increasing complexity of the control processes for creating and editing articles that are put in place to overcome the problem of biases, negatively impacts the growth of Wikipedia. Edit conflict resolution, arbitration committees, banning policies, a complex hierarchy of contributors, editors and administrators is not sustainable. Effectively, recent statistics show that the number of new articles has been decreasing dramatically over the past years, while the number of edits is still growing steadily. Discovering missing con-

[1] render-project.eu

DiversiWeb 2011, March 28, Hyderabad, India..

tent from one language version of Wikipedia to another, or the detection of diverse viewpoints within a topic or article are urgently needed support to the editorial team for managing and encouraging large-scale participation and sustainable growth. Diversity-empowered services such as quality or reliability assessment of an article or a specific statement, conflict resolution, anomaly detection, and cross-lingual consistency checking are expected to considerably improve the way information is currently managed in Wikipedia.

2.2 Google News

The news aggregator service of Google (Google News) indexes several ten thousands of news Web sites which are summaries into more than forty regional issues in more than 15 languages. The considered news content is created by professional journalists and by Web users, and offers as such a rich diversity of information. Current ranking algorithms result in news summaries that are dominated by popular viewpoints or opinion holders such as large news agencies. Alternative opinions, or arguments from smaller publishers often disappear and do not reach the interested audience. Consequently, even though Google aims for wide and comprehensive news coverage, the presented view points are highly biased. Manual processing is costly and impractical, and techniques to automatically discover diverse opinions, viewpoints and discussions surrounding a topic are required to fully leverage the richness in news content. Diversity-aware ranking of news posts for covering the most diverse view points on a particular topic, and enriching these with data from other sources like blogs, tweets, and wiki pages is expected to considerably increase the interconnection between diversifying news and discussions on the Web.

2.3 Customer Relationship Management

Telefónica is one of the World's largest telecommunications companies by market share, operating in 25 countries with a global customer base exceeding 280 millions. The company maintains various different communication channels including call centers, Web sites and public forums and blogs to collect customer feedback about their products and services. This offers a massive amount of valuable user opinions coming from diverse sources, countries and socio-demographic groups that are currently only marginally exploited, as the technical support for automation is missing and manual processing is not feasible to the desired extent. Discovering and automatically evaluating customer reactions and discussions are expected to allow Telefónica to react more efficiently and effectively to trends, to make more precise forecasts, and to eventually improve future business decisions.

3. KNOWLEDGE DIVERSITY GLOSSARY

The first step towards our knowledge diversity model is to create a shared understanding of the relevant terms and relationships between them in the scope of knowledge diversity. In this section, we present a summary of definitions of possibly relevant terms to get a rough understanding of the key concepts in the scope of knowledge diversity. We do not attempt to define these concepts in this paper; instead we refer to the existing definitions of these concepts.

Agent is described in DOLCE+DnS Ultralite as an agentive object, either physical (e.g. a person), or social (e.g. a corporation, an institution, a community).[2] As an extension of this concept, an agent expressing an opinion of his own can be called an *opinion holder*.

Belief is given by Wikipedia as "the psychological state in which an individual holds a proposition or premise to be true".[3] WordNet defines belief as "any cognitive content held as true", or alternatively as "a vague idea in which some confidence is placed".[4]

Bias is defined by Wikipedia as "an inclination to present or hold a partial perspective at the expense of (possibly equally valid) alternatives".[5] The definition of bias by Giunchiglia *et al.* in [5] states that "bias is the degree of correlation between (a) the polarity of an opinion and (b) the context of the opinion holder". The context can be a variety of factors such as ideological, political, or educational background, ethnicity, race, profession, age, location, or time.

Data is definded by WordNet as "a collection of facts from which conclusions may be drawn".[6] Wikipedia states that "the term data refers to qualitative or quantitative attributes of a variable or set of variables". Furthermore, data is the lowest level of abstraction from which first information and then knowledge are derived.[7]

Diversity is described in the philosophical sense, according to [3], as "the relation that holds between two entities when and only when they are not identical". In the Cambridge Advanced Learner's Dictionary diversity is defined as: "when many different types of things or people are included in something".[8] In [5] diversity is given from a more knowledge diversity focused point of view as "the co-existence of contradictory opinions and/or statements (some typically non-factual or referring to opposing beliefs/opinions)". In the same paper, different dimensions of diversity are described such as: diversity of resources, diversity of topic, diversity of viewpoint, diversity of genre, diversity of language, geographical/spatial diversity, and temporal diversity.

Emotion is defined by Liu as "subjective feelings and thoughts" [7]. As Liu discusses, people use language expressions to describe their mental state (or feelings). According to [8], there are a large number of language expressions to depict the six types of emotions; i.e., *love*, *joy*, *surprise*, *anger*, *sadness* and *fear*. Similarly, people use a large number of opinion expressions to convey opinions with positive or negative sentiment.

Entity is described by Wikipedia as "something that has a distinct, separate existence, although it need not be a material existence".[9] In entity-relationship modelling, an entity is defined as "a thing which is recognized as being capable of an independent existence and which can be uniquely identified".[10]

[2] ontologydesignpatterns.org/ont/dul/DUL.owl

[3] en.wikipedia.org/wiki/Belief

[4] wordnetweb.princeton.edu/perl/webwn?s=belief

[5] en.wikipedia.org/wiki/Bias

[6] wordnetweb.princeton.edu/perl/webwn?s=data

[7] en.wikipedia.org/wiki/Data

[8] dictionary.cambridge.org/dictionary/british/diversity

[9] en.wikipedia.org/wiki/Entity

[10] en.wikipedia.org/wiki/Entity-relationship_model

Event is described in DOLCE+DnS Ultralite as "any physical, social, or mental process, event, or state". DOLCE+DnS Ultralite classifies events based on 'aspect' (e.g., stative, continuous, accomplishment, achievement, etc.), on 'agentivity' (e.g., intentional, natural, etc.), or on 'typical participants' (e.g., human, physical, abstract, food, etc.).

Fact, according to Liu, is the "objective expressions about entities, events and their properties" [7]. Wikipedia states that facts "refer to verified information about past or present circumstances or events which are presented as objective reality".[11] The Merriam-Webster Online Dictionary defines fact, *inter alia*, as 1) "the quality of being actual." 2) "something that has actual existence." or "An actual occurrence", 3. "a piece of information presented as having objective reality".[12]

Information is defined in [4] in terms of *data + meaning*:
σ is an instance of information, understood as semantic content, if and only if:
i) σ consists of n *data*, for $n \geqslant 1$;
ii) the data are *well formed*;
iii) the well-formed data are meaningful.

According to this definition, information is made of data and 'well formed' here means that data are rightly put together. Well formed and meaningful data are also known as *semantic content*. Information, understood as semantic content, has two major types: (a) *instructional* information, conveying the need for a specific action (b) *factual* information.

Information Object is described by DOLCE+DnS Ultralite as "a piece of information, such as a musical composition, a text, a word, a picture, independently from how it is concretely realized".

Knowledge is informally described in [2]. In a sentence like "John knows that Sara will come to the party", knowledge is "a relation between a knower, like John, and a proposition, that is, the idea expressed by a simple declarative sentence", like "Sara will come to the party". The proposition here are the abstract entities that can be *true* or *false*, right or wrong. More specifically, the sentences expressing the propositions, which are factual or non-factual, are *true* or *false*. The relationship between agents and propositions have different *propositional attitude* denoted by verbs like "know", "hope", "fear", "regret", and "doubt" etc. Brachman and Levesque do not consider the sentences involving knowledge that do not explicitly mention a proposition. For example, it is not clear if there is any useful proposition involved in the sentences like "John knows how to play guitar" or "John knows Bob well". Brachman and Levesque also discuss that the notion of *belief* is related to the notion of *knowledge*. People use the notion of *belief* if they do not want to claim that the judgement of an agent about the world is necessarily accurate.

Metadata is defined by Wikipedia as the "data providing information about one or more aspects of the data",[13]; e.g., means of creation of the data, purpose of the data, time and date of creation, creator or author of data, placement on a computer network where the data was created, or standards used. WordNet simplifies the meaning of metadata as "data about data".[14]

Object is described in DOLCE+DnS Ultralite as "any physical, social, or mental object, or a substance". The definition of objects by Liu states that "an object o is an entity which can be a product, person, event, organization, or topic [7]. It is associated with a pair, o: *(T, A)*, where T is a hierarchy of components (or parts), sub-components, and so on, and A is a set of attributes of o. Each component has its own set of sub-components and attributes".

Objectivity is the expression of facts [1]. Wikipedia moreover describes objectivity as "a proposition is generally considered to be objectively true when its truth conditions are mind-independent – that is, not the result of any judgements made by a conscious entity or subject".[15] WordNet defines it as the "judgment based on observable phenomena and uninfluenced by emotions or personal prejudices",[16] while according to [7] objective sentences express factual information about the world.

Object Feature represents the components and attributes of objects [7]. The term object feature is also referred simply as feature. Object features are used to simplify the complexity of hierarchical representation of the components of objects.

Opinion is defined by Wikipedia as "a subjective statement or thought about an issue or topic, and is the result of emotion or interpretation of facts".[17] Furthermore, "an opinion may be supported by an argument, although people may draw opposing opinions from the same set of facts". In [5], opinion is defined as "a statement, i.e. a minimum semantically self-contained linguistic unit, asserted by at least one actor, called the opinion holder, at some point in time, but which cannot be verified according to an established standard of evaluation. It may express a view, attitude, or appraisal on an entity. This view is subjective, with positive/neutral/negative polarity (i.e. support for, or opposition to, the statement)". Another definition of opinion, given by Liu [7], states that "an opinion on a feature f is a positive or negative view, attitude, emotion or appraisal on f from an opinion holder".

Opinion Expression is given by Liu as subjective expression that describes sentiments, appraisals or feeling toward entities, events and their properties [7]. More generally speaking, it could be said that opinion expressions are individual statements that contain an assessment of reality from the point of view of the *opinion holder*.

Opinion Holder, according to Liu [7], is "the person or organization that expresses the opinion"; see *Agent* above.

Polarity of Opinion on a feature f indicates if the opinion is positive, negative or neutral [7]. [5] describes polarity as the degree to which a statement is positive, negative or neutral. The polarity of an opinion is also known as sentiment orientation or semantic orientation [7].

Sentiment is defined in the American Heritage Dictionary

[11] en.wikipedia.org/wiki/Fact

[12] www.merriam-webster.com/dictionary/fact

[13] en.wikipedia.org/wiki/Metadata

[14] wordnetweb.princeton.edu/perl/webwn?s=metadata

[15] en.wikipedia.org/wiki/Objectivity_(philosophy)

[16] wordnetweb.princeton.edu/perl/webwn?s=objectivity

[17] en.wikipedia.org/wiki/Opinion

of the English Language as "a thought, view, or attitude, especially one based mainly on emotion instead of reason".[18] Sentiments can be seen as a way to express opinions. Hence, sentiments, as much as opinions, can be negative, positive or neutral [7].

Subjectivity refers to the subject and the perspective, feelings, beliefs, and desires of the subject [6]. Liu defines subjective sentences as the sentences which "express some personal feelings or beliefs" [7].

Text is defined by Dictionary.com, in the linguistic sense, as "a unit of connected speech or writing, especially composed of more than one sentence, that forms a cohesive whole".[19] The Free On-line Dictionary of Computing describes it as the "textual material in the mainstream sense", and in the computing sense as the "data in ordinary ASCII or EBCDIC representation", where ASCII and EBCDIC are computer codes for representing alphanumeric characters.[20]

Topic has three definitions in Wikipedia: "a.) the phrase in a clause that the rest of the clause is understood to be about, b.) the phrase in a discourse that the rest of the discourse is understood to be about, c.) a special position in a clause (often at the right or left-edge of the clause) where topics typically appear".[21] WordNet defines topic as "the subject matter of a conversation or discussion".[22]

4. RELATED WORK

Giunchiglia *et al.* consider knowledge diversity as an asset to improve navigation and search [5], however, they do not provide a representation model to represent the knowledge gathered using their technology. Liu introduces the core topics in the field of sentiment analysis and opinion mining, such as sentiment and subjectivity classification, feature-based sentiment analysis, sentiment analysis of comparative sentences, opinion search and retrieval, opinion spam and utility of opinions [7]. Liu provides definitions of the relevant concepts but the work is aimed at the processing of opinions, and not at representing opinions. Balahur and Steinberger provide their insight on sentiment analysis for the news domain [1], and as such argue the need for clearly defining the source and target of a sentiment. They provide guidelines on annotating news contents with different sentiments, however, they do neither discuss the representation of the captured knowledge.

The listed works present technologies and methodologies to gather different aspects of diversity, but they do not provide any representation model for this gathered knowledge. In contrast, our aim is to work towards developing a knowledge diversity model to represent the different aspects of diversity.

5. FUTURE WORK AND CONCLUSIONS

The goal of this paper was to collect a comprehensive glossary of terms that are relevant in the context of knowledge diversity. Aspects such as opinion, sentiment or bias are essential in understanding the diversity of news posts, Wikipedia articles, or customer feedback. Only when diversity can be computationally accessible to the machine, the capturing and interpretation of opinions and sentiments can be automated and results extracted at larger scale.

The intention is to derive a knowledge diversity model from the glossary presented in this paper. In the next step it will be necessary to determine the concrete questions that will have to be answered for the showcase scenarios, and to extract the definitions that cover these relevant aspects. Another important future work would be to determine the relationships among the aforementioned concepts. As an example, based on the definition presented in this paper we can conclude that sentiments are a way to express opinions. Subjectivity refers to the perspective, beliefs and feelings of a person. Bias is influenced by someone's personal opinion. A particular bias can influence the subjectivity of a sentence when it contains an opinion. Opinions are expressed by the opinion expressions. Opinion expressions are subjective statements contained in the information objects. The concepts and relationships can be seen as the baseline for the specification of the knowledge diversity ontology that yields the schema information for semantically capturing the diversity and context of the textual content considered. Context, also not part of the collected definitions above, is important to interpret diverse standpoints in view of their socio-demographic, spatio-temporal and historic relationship to each other. In many situation, taking the customer relationship management as an example, it is not only relevant to interpret diverging opinions and sentiments of customers but also to understand the situation of the opinion holders such as for example their country of residence. This allows for drawing further conclusions relevant for shaping the business.

Acknowledgments: The work presented in this paper is supported by the European Union's 7th Framework Programme (FP7/2007-2013) under Grant Agreement 257790.

6. REFERENCES

[1] A. Balahur and R. Steinberger. Rethinking Sentiment Analysis in the News: from Theory to Practice and back. In *1st Workshop on Opinion Mining and Sentiment Analysis*, 2009.

[2] R. Brachman and H. Levesque. *Knowledge representation and reasoning*. Morgan Kaufmann Publishers, 2004.

[3] J. Butterfield. *Collins English dictionary: Complete and unabridged*. HarperCollins Publishers, 2003.

[4] L. Floridi. *Information: A Very Short Introduction*. Oxford University Press, 2010.

[5] F. Giunchiglia, V. Maltese, D. Madalli, A. Baldry, C. Wallner, P. Lewis, K. Denecke, D. Skoutas, and I. Marenzi. Foundations for the representation of diversity, evolution, opinion and bias. Technical Report DISI-09-063, University of Trento, 2009.

[6] T. Honderich. *The Oxford Companion to Philosophy*. Oxford University Press, 2005.

[7] B. Liu. *Handbook of Natural Language Processing*, chapter Sentiment Analysis and Subjectivity, pages 627–666. CRC Press, 2010.

[8] W. Parrott. *Emotions in Social Psychology: Essential Readings*. Psychology Press, 2001.

[18] `www.houghtonmifflinbooks.com/ahd/`

[19] `dictionary.reference.com/browse/text`

[20] `foldoc.org/text`

[21] `en.wikipedia.org/wiki/Topic`

[22] `wordnetweb.princeton.edu/perl/webwn?s=topic`

Expressing Opinion Diversity

Andreea Bizău
Faculty of Mathematics and Computer Science
Babeș-Bolyai University
Cluj-Napoca, Romania
andreea.bizau@gmail.com

Delia Rusu
Artificial Intelligence Laboratory
Jožef Stefan Institute
Ljubljana, Slovenia
delia.rusu@ijs.si

Dunja Mladenić
Artificial Intelligence Laboratory
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

The focus of this paper is describing a natural language processing methodology for identifying opinion diversity expressed within text. We achieve this by building a domain-driven opinion vocabulary, in order to be able to identify domain specific words and expressions. As a use case scenario, we consider Twitter comments related to movies, and try to capture opinion diversity by employing an opinion vocabulary, which we generate based on a corpus of IMDb movie reviews.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: *Text analysis*.

General Terms

Algorithms, Design.

Keywords

Opinion mining, natural language processing, social networks.

1. INTRODUCTION

Information is expressed on the Web under a variety of forms, some of them more formal and standardized, like news articles, others more spontaneous, ad-hoc, like blogs or microblogs. One challenge is to tap into these sources, and allow for a diverse representation of information on the same topic, presenting different points of view, opinions, arguments.

In this work we are describing a natural language processing methodology for discovering the diversity of opinions expressed within text, which we deem to be an essential step to expressing and presenting diverse information on the Web. In this context, we consider an *opinion* as a subjective expression of sentiments, appraisals or feelings, and *opinion words* as a set of keywords/phrases used in expressing an opinion. As such, the *orientation* of an opinion word indicates whether the opinion expressed is positive, negative or neutral, while the totality of opinion words forms an *opinion vocabulary*. While opinion words can be analyzed in their base form (describe and convey the opinion directly) and comparative form (convey the opinion indirectly, by comparison with other entities), this research focuses only on base type opinion words.

In the context of the ever expanding world of social media and user generated content, instant access, world-wide coverage and diversity of perspective are the norm of the information flow. As an application of our approach, we propose to study the movie domain. There is a strong user interest in watching, tracking and discussing movies, generating highly diverse opinion content. Movies are subject to a variety of classifications, expanding the field of analysis. Moreover, the lifespan of a movie topic is longer than for usual topics, thus introducing a temporal dimension that can be further explored. Nowadays, accessing and assessing the public opinion has taken on a new form. Social networking encourages the exchange of information and sharing of opinions between individuals, friends and communities. Therefore, in our case study we directly address movie comments, as posted on Twitter, a popular social networking and microblogging website, and aim at identifying the diversity of opinions expressed in tweets related to movies. We determine a variety of polarized opinion words about a certain movie, and use these word frequency counts to obtain an overall aggregated opinion about the movie. Moreover, we can observe variations in opinions over time, related to a certain movie, by comparing the word frequency counts obtained from tweets belonging to a time interval (e.g. an hour, day, week).

The paper is structured as follows: in Section 2 we describe our algorithm for constructing a domain-driven opinion vocabulary, while Section 3 presents the Twitter movie comments use-case. The last section of the paper is dedicated to conclusions and future work.

2. DOMAIN DRIVEN OPINION VOCABULARY

We start from the idea that expressing opinions is dependent on the topic's context and we focus on the role of adjectives as opinion indicators; in the future we plan to broaden this line of work by including verbs and adverbs. The starting point is represented by a domain-specific corpus, from which we determine a small number of seed opinion words that we further extend, thus forming a domain-driven opinion vocabulary.

There are three main approaches to constructing an opinion vocabulary: manual, dictionary based and corpus based. The manual approach is not really in line with our work, as we are considering automatic, scalable approaches. The dictionary based approach provides a simple and efficient way of obtaining a good vocabulary. SentiWordNet [3] is a publicly available lexical resource. It provides tags of all WordNet [4] synsets with three numerical scores (objective, positive, negative), offering a general opinion vocabulary with good coverage. However, the dictionary-based approach cannot account for the domain specific orientation of words, nor can it identify domain specific words and expressions. As an example, consider the word *unpredictable*. In most situations it will express an undesirable quality (e.g. unpredictable car behavior), thus its orientation will be negative; but in the movie domain, an unpredictable plot is something desired and indicates a positive opinion. In order to account for domain specificity, we decided to employ a corpus based approach.

V. Hatzivassiloglou et al [6] showed the relevance of using connectives in gathering information about the orientation of conjoined adjectives. They emphasized that conjoined adjectives are of the same orientation, for most connectives, *but* reversing the relationship. The connectives are conjunctions used to join one or more adjectives together. In our algorithm we used a subset of the possible conjunctions (*and*, *or*, *nor*, *but*, *yet*), that cover many common syntactic patterns and are easier to correlate with the adjectives that they connect.

Other lines of research, like S.-M. Kim and E. Hovy [7] try to identify opinion expressions together with their opinion holder starting from a word seed list and use the WordNet synsets to determine the strength of the opinion orientation for the identified opinion words. M Gamon and A. Aue [5] extend the Turney-style [9] approach of assigning opinion orientation to the determined candidate words, working under the assumptions that in the opinion domain, opinion terms with similar orientation tend to co-occur, while terms with opposite orientation do not tend to co-occur at sentence level.

V Jijkoun et al [10] propose a different style of approach, by starting from an existing lexicon (clues) and focusing it. They perform a dependency parsing on a set of relevant documents, resulting in triplets (clue word, syntactic context, target of sentiment) that represent the domain specific lexicon. H. Kanayama and T. Nasukawa [11] apply the idea of context coherency (same polarity tend to appear successively) to the Japanese language. Starting from a list of polar atoms (minimum syntactic structure specifying polarity in a predicative expression), they determine a list of domain specific words using the overall density and precision of coherency in the corpus. Sinno Jialin Pan et al [12] propose a cross-domain classification method. Starting from a set of labeled data in a source domain and determining domain-independent words (features) that occur both in the source and the target domain, they construct a feature bipartite graph that models the relationship between domain-specific words and independent words. To obtain the domain specific words they use an adapted spectral clustering algorithm on the feature graph

Based on these premises, we propose a method to construct an opinion vocabulary by expanding a small set of initial (seed) words with the aid of connectives. The method consists of four steps, as follows:

1. Given a positive word seed list and a negative word seed list and making use of WordNet's synsets, we expand the initial seed lists based on the synonymity / antonymy relations.

The initial words will be assigned a score of 1 for positive words and -1 for negative words, respectively. We compute the orientation score for each newly found word by recursively processing the synsets for each seed word. A word can be found in synsets corresponding to different seed words, either in a synonymity or antonymy relations. Another factor we take into account is the *distance* between the seed word and the currently processed word, as provided by the WordNet hierarchy. From these two considerations, a more formal way to compute the score of a word (s_w) to be added to the seed list is:

$$s_w = \max(abs(s_{w,o}) \cdot sign(\max(s_{w,o})))$$

where

$$s_{w,o} = \begin{cases} f \cdot s_o, when\ w\ and\ o\ are\ synonyms \\ -f \cdot s_o, when\ w\ and\ o\ are\ antonyms \end{cases}$$

and *o* is a seed word, while *f* is a parameter for which we empirically assigned values between 0 and 1 (in our current implementation *f = 0.9*); in our future work we plan to determine its value by optimization.

The result of this step is an expanded seed word list together with their orientation score.

2. From a corpus of documents, we parse and extract all adjectives and conjunctions, constructing a set of relationships between the determined words. There can be two types of relationships, indicating if two or more words have the same context orientation (words connected by *and*, *or*, *nor*) or opposite orientation (words connected by *but*, *yet*). We will refer to them in the following algorithms as *ContextSame* and *ContextOpposite* relations, respectively.

```
$G = (\{\}, \{\})$
foreach document $d$ in corpus
 foreach sentence $s$ in $d$
     parseTree = GetParseTree(s)
     {w,c} = RetrieveWordsAndConjunctions(parseTree)
     ConstructRelationGraph(G, {w, c})
     HandleNegation(G, s)
```

Figure 1. The algorithm for constructing the relationship graph G.

Based on the determined relations, we can then construct a relationship graph *G(W, E)*, where

- *W={set of determined adjectives}* and
- $E=\{w_iw_j$, where w_i, w_j from W if there is a determined relationship between w_i and w_j, each edge having a positive weight for the *ContextSame* relationship and a negative weight for the *ContextOpposite* relationship}.

In what follows, we describe the algorithm for building the relationship graph G (see Figure 1).

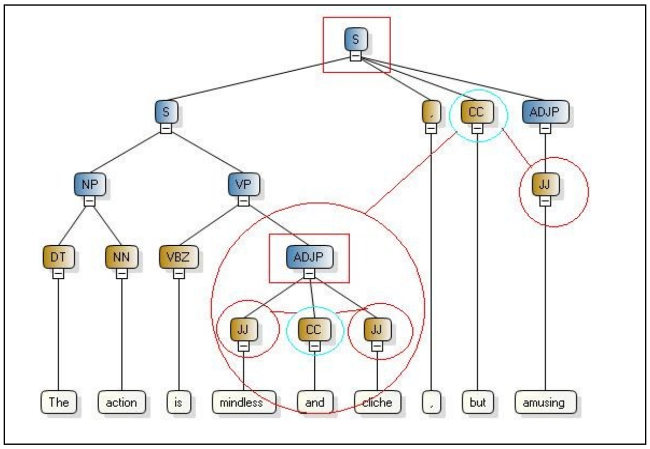


Figure 2. The parse tree and analysis of the sentence "The action is mindless and cliché, but amusing". We identify *mindless, cliché, amusing* as adjectives (having the JJ tags) connected by *and*, *but* (having the CC tags).

We used a maximum entropy parser[1] to retrieve a sentence's parse tree that we then analyze in the *RetrieveWordsAndConjunctions* procedure. We construct an adjective stack *w* and a conjunction stack *c* by extracting the relevant nodes according to their part-of-speech tags and group them together based on the common parent node between the adjective nodes and the conjunctions nodes. In the *ConstructRelationGraph*, we will add the nodes for each newly found adjective and add new edges to the relationship

[1] http://sharpnlp.codeplex.com/

graph *G* according to each conjunction's behavior. Each edge has an associated weight with values between 0 and 1, determined by optimization. We handle the presence of negation in the sentence by reversing the type of the relation, if a negation is detected. For example, considering the sentence "*Some of the characters are fictitious, but not grotesque*", the initial relation between *fictitious* and *grotesque* would be a *ContextOpposite* relationship, but the presence of the negation is converting it to a *ContextSame* relationship. We depict another example visually, in Figure 2.

3. The third step implies cleaning the resulting set of words and relationship graph by removing stop words and self-reference relations. Consider the example "*The movie has a good casting and a good plot*". The algorithm detects a *ContextSame* relationship between the adjective *good* and itself. Since there is no useful context information we can use, we do not want them to influence the results of the scoring done in the next step.

4. In the fourth step, we determine the orientation of the words extracted from the corpus by applying an algorithm on the relationship graph obtained in the previous steps, which was inspired by the well-known PageRank algorithm [2]. For this, we define two score vectors, a positivity score *sPos* and a negativity score *sNeg*, respectively. We choose the final score to be the sum of the positivity and negativity score. The sign of the score represents the word's orientation, that is, a positive score characterizes a positive opinion orientation, while a negative score characterizes a negative opinion orientation. The algorithm is presented in Figure 3, and described in what follows.

```
InitializeScoreVectors(sPos(W), sNeg(W))
do {
foreach word $w_i$ in $W$
    foreach relation $rel_{ij}$ in relationship graph $G$ that contains $w_i$
        if $rel_{ij}$ is a ContextSame relation
            sPos($w_i$) += weigth($rel_{ij}$) * prevSPos($w_j$)
            sNeg ($w_i$) += weigth($rel_{ij}$) * prevSNeg($w_j$)
        else if $rel_{ij}$ is a ContextOpposite relation
            sPos($w_i$) += weigth($rel_{ij}$) * prevSNeg($w_j$)
            sNeg ($w_i$) += weigth($rel_{ij}$) * prevSPos($w_j$)
    NormalizeScores(sPos($w_i$), sNeg($w_i$))
} while more than 1% of the words $w_i$ in $W$ change orientation
```

Figure 3. The algorithm for determining the orientation of words extracted from a corpus.

We initialize the score vectors based on the orientation scores of the expanded seed word list (see step 1). We will assign the corresponding positivity or negativity score sw_j for each adjective *w* found in the seed list. For the opposite score we assign a very small value (ε), in order to allow for meaningful values when computing the score for *ContextOpposite* relations.

A *ContextSame* relation enforces the existing positive and negative scoring of w_i proportionally with the scoring of w_j. A *ContextOpposite* enforces the negativity score of w_j with respect to the positivity of w_i, and the positivity score of w_j with respect to the negativity score of w_i.

3. USE CASE: TWITTER MOVIE COMMENTS

Concerning the movie domain, research was done in classifying movie reviews by overall document sentiment [8], but there are few lines of research connecting the movie domain with social media. Sitaram Asur and Bernardo A. Huberman [1] demonstrate how sentiments extracted from Twitter can be used to build a prediction model for box-office revenue.

Our aim is to see how well a domain specific vocabulary constructed from movie reviews performs when applied to analyzing tweets. We used a document corpus of 27,886 IMDb (Internet Movie Database) movie reviews[3] and constructed a movie domain specific vocabulary according to the approach presented in Section 2. We retrieved 9,318 words, from which 4,925 have a negative orientation and 4393 have a positive orientation. Table 1 shows a few examples of positive and negative adjectives extracted from the movie review corpus.

Table 1. Examples of adjectives that were extracted.

Positive words	Negative words
surprised, original, breathless, chilling, undeniable, disturbing, irresistible, speechless, stylized, amazed, provoking, shocking, undisputed, unforgettable, electrifying, enraptured, explosive, unanticipated, unforeseen, recommended	syrupy, uninspiring, forgettable, frustrating, mild, contrived, laughable, restrained, showy, preachy, amateur, dogmatic, edgeless, foreseeable, ordinary, standard, saleable, usual, predictable

Table 2. Top opinion words identified for the highest and lowest ranking movies in our search

Inception (2010)	Meet the Spartans (2008)
Positive words: good, great, awesome, amazing, favorite, fantastic, incredible, thrilling, different, speechless *Negative words*: bad, confusing, weird, stupid, dumb, boring, predictable, horrible, disappointing	*Positive words*: funny, awesome, great *Negative words*: bad, stupid, dumb, weird, silly, common, ridiculous, terrible

For our tests, we crawled 220,387 tweets, using the Twitter Search API[6], over a two month interval, keyed on 84 movies, spanning different genres and release dates. As search keywords we used the movie name and the *movie* tag, in order to increase the relevance of the results. We used a simple tokenizer to split the text of the retrieved tweets and kept the tokens that had a dictionary entry as adjectives. We then matched the tweet adjectives to our domain specific vocabulary. For all subsequent analysis we only considered adjectives that were used in tweets and also appeared in our vocabulary, since we were intersected to see the relevance of our vocabulary in terms of actual usage and frequency over time. Without actually classifying each tweet, we counted the frequency of positive and negative opinion words that we identified in the collection of tweets. An example of top opinion words that we identified for the highest and lowest

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6]http://search.twitter.com/api/

ranking movies are shown in Table 2. Table 3 presents a sample of the movies that we analyzed, showing for each movie the genre, number of tweets, our score obtained by counting the positive opinion words and the IMDb score. In Figure 4 we represent graphically the positive and negative opinion word counts for the movie *Inception*.

Table 3. A sample of the movies that we analyzed, showing for each movie the genre, number of tweets, our score obtained by counting the positive opinion words and the IMDb score.

Movie	Genre	Our score	IMDb score	Tweets
Inception (2010)	mystery, sci-fi, thriller	66.52	8.9	19,256
Megamind (2010)	animation, comedy, family	67.71	7.3	8,109
Unstoppable (2010)	drama, thriller	63.67	7	15,349
Burlesque (2010)	drama, music, romance	70.78	6.2	1,244
Meet the Spartans (2008)	comedy, war	40.67	2.5	44
Pootie Tang (2001)	comedy, musical	45.88	4.5	79
Matrix (1999)	action, sci-fi	56.65	8.7	1,947
Blade Runner (1982)	drama, sci-fi, thriller	56.65	8.3	407
Metropolis (1927)	sci-fi	66.23	8.4	419

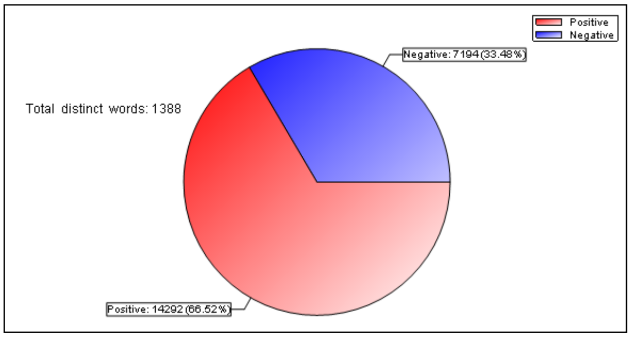


Figure 4. Word distribution for the movie *Inception* over 19,256 tweets.

In the cases presented in Table 3, there is a relationship between the number of positive opinion words and the rating from IMDb. One thing to notice is that in IMDb the movie ratings can be roughly grouped in three categories: ratings between seven and ten points accounting for good and very good movies, between five and seven points for average movies, and below five points for poor quality movies. Our positive opinion word count has a maximum of approximately 70 (or seven on a scale from zero to ten). In our future work we plan to conduct a series of experiments in order to determine if there exists a correlation between the two numbers: the IMDb rating and the number of positive opinion words. This involves collecting a higher number of movie related tweets (in the order of hundreds) in order to be able to report significant results.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented an approach to identifying opinion diversity expressed within text, with the aid of a domain-specific vocabulary. As a use case, we processed a corpus of IMDb movie reviews, extracted a set of adjectives together with their opinion orientation and used the generated opinion lexicon to analyze a different opinion source corpus, i.e. a tweet collection. For future work, we plan to further extend our algorithm to include opinion words expressed by verbs and adverbs, as well as more complex expressions. A second item point is carrying out a set of experiments in order to determine the correlation between positive opinion words for a given movie and the IMDb movie rating. Thirdly, from the lessons learned, we would look into applications in other domains like product reviews.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the Slovenian Research Agency and the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°257790.

6. REFERENCES

[1] Asur, S. and Huberman, B. A. 2010. *Predicting the Future With Social Media*. In Proceedings of the ACM International Conference on Web Intelligence.

[2] Brin, S. and Page, M. 1998. *Anatomy of a large-scale hypertextual Web search engine*. In Proceedings of the 7th Conference on World Wide Web (WWW).

[3] Esuli, A. and Sebastiani, F. 2006. *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining*. In Proceedings of the 5th LREC.

[4] Fellbaum, Ch. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

[5] Gamon, M and Aue, A. 2005. *Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms*. In Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP.

[6] Hatzivassiloglou, V. and McKeown, K. 1997. *Predicting the semantic orientation of adjectives*. In Proceedings of the 35th Annual Meeting of the ACL.

[7] Kim, S-M. and Hovy, E. 2004. *Determining the sentiment of opinions*. In Proceedings of COLING.

[8] Pang, B. and Lee, L. 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In Proceedings of EMNLP.

[9] Turney, P. D. 2002. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. In Proceedings of the 40th Annual Meeting on ACL.

[10] Jijkoun, V., de Rijke, M. and Weerkamp, W. 2010. *Generating Focused Topic-Specific Sentiment Lexicons*. In Proceedings of the 48th Annual Meeting of the ACL.

[11] Kanayama, H. and Nasukawa, T. 2006. *Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis*. In Proceedings of the EMNLP.

[12] Pan, S. J., Ni, X., Sun, J-T, Yang, Q. and Chen, Z. 2010. *Cross-Domain Sentiment Classification via Spectral Feature Alignment*. In Proceedings of the World Wide Web Conference (WWW).

Scalable Detection of Sentiment-Based Contradictions

Mikalai Tsytsarau
University of Trento
Trento, Italy
tsytsarau@disi.unitn.eu

Themis Palpanas
University of Trento
Trento, Italy
themis@disi.unitn.eu

Kerstin Denecke
L3S Research Center
Hannover, Germany
denecke@L3S.de

ABSTRACT

The analysis of user opinions expressed on the Web is becoming increasingly relevant to a variety of applications. It allows us to track the evolution of opinions or discussions in the blogosphere, or perform product surveys. The aggregation of sentiments and analysis of contradictions is another important application, which becomes effective since we are able to capture the diversity in sentiments on different topics with more precision and on a large scale. Though, there is still a need for a scalable way of sentiment aggregation with respect to the time dimension, which preserves enough information to capture contradictions.

In this paper, we are focusing on the problem of finding sentiment-based contradictions at a large scale. First, we define two types of contradictions, depending on the distributions of opposite sentiments over time. Second, we introduce a novel measure of contradiction based on the mean value and the variance of sentiments among different texts. Third, we propose a scalable method for identifying both types of contradictions at different time scales. We evaluate the performance of our method using synthetic and real-world datasets, as well as a user-study. The experiments demonstrate the effectiveness of the proposed method in capturing contradictions in a scalable manner.

1. INTRODUCTION

During the recent years we have been witnessing the Internet becoming an open platform, where people can express their opinions and can be heard. There are many services that allow people to publish information and opinions, such as blogs, wikis, forums, social networks and others. They all represent a rich source of opinionated information on different topics, which can be analyzed and exploited in various applications and contexts. Sentiment analysis can be used, for example, to learn about a customer's attitude to a product or its features, or to reveal people's reaction to some event. Such problems require a scalable analysis and some form of sentiments aggregation to produce a representative result.

The problem of contradictions, or sentiment diversity on some topic, has been studied in the context of different research areas, having a slightly varying notion in each case. For instance, in Information Retrieval opposite opinions and sentiments introduce noise to the fact-centric search and must be avoided [14]. In contrast, conflicting sentiments is one of the desired targets of mining of product reviews. Recently proposed methods can aggregate opinions expressed in customer reviews and extract a representative summary of sentiments on a feature-by-feature basis; or they can capture and aggregate sentiments on some topic among different texts [8].

Although aggregated sentiments do represent some information on contradiction, this information may be biased. For example, if two opposite sentiment values are averaged, the result may have a neutral polarity. The information about the contradiction is then lost. On the other hand, representative sentiments (which best describe opposite opinions) are likely to capture the meaning of contradiction, but not its level. Therefore, this problem essentially requires a consistent definition and new methods to deal with it.

In this paper, we introduce a framework[1] that defines the concepts of aggregated sentiment, sentiment variance and contradiction with respect to the time dimension, and formulates relevant problems of contradiction discovery. We say that we have a contradiction when there are conflicting opinions for a specific topic, which is a form of sentiment diversity. This kind of contradiction can occur at one specific point of time or throughout a certain time period. Furthermore, a contradiction can occur within one text when an author presents different opinions on the same topic, or across texts when different authors express different opinions on the same topic. We further extend this framework of contradiction detection by focusing on its performance and effectiveness for large-scale datasets.

Our method operates on sentence-level sentiments, which are represented in a continuous scale. This allows us to exploit different approaches for sentiment detection, which can be plugged in our framework. The use of mean and variance for contradiction detection allows our method to be fast and linearly scalable on the number of texts, which is an important feature for large-scale analysis. Tests on real datasets, as well as a user-study, demonstrate that our approach is able to efficiently and effectively identify contradictions.

The main contributions of this work can be summarized as follows.

- We formally define the problem of contradiction detection, and further describe two variations of the problem, namely, *synchronous* and *asynchronous* contradictions.
- We present an approach for contradiction detection, which is based on fine-grained sentiment extraction. Moreover, we describe techniques that enable this approach to scale to very large data collections.
- We experimentally evaluate the proposed approach using several synthetic and real datasets. The results show the effectiveness and scalability of our solution. In addition, we perform a user-study that demonstrates the usefulness of the proposed framework.

The remainder of this paper is structured as follows. In Section 2 we discuss the related work, and in Section 3 we formally define the problem. We present our approach for detecting and storing contradictions in Section 4 and Section 5, respectively, and the experimental evaluation in Section 6. We discuss our experiences in Section 7, and conclude in Section 8.

[1] Some preliminary ideas have appeared as a poster [16].

2. RELATED WORK

In the past few years, we have witnessed an increasing research interest in the area of blog analysis and specifically in opinion mining [13]. Contradiction analysis is a rather new research area. In particular, contradictions in opinions as considered here, have not been addressed before. Harabagiu et al. [6] present a framework for contradiction analysis that exploits linguistic information such as negation or antonymy as well as semantic information, such as types of verbs. De Marneffe et al. [3] introduce a classification of contradictions consisting of seven types that are distinguished by the features that contribute to a contradiction (e.g., antonymy, negation, numeric mismatches). They define contradictions as a situation where 'two sentences are extremely unlikely to be true', and describe a contradiction detection approach to their textual entailment application [12]. Ennals et al. [5] describe an approach that detects contradicting claims by checking whether some particular claim entails (i.e., has the same sense as) one of those that are known to be disputed. For this purpose, they have aggregated disputed claims from Snopes.com and Politifact.com into a database. Additionally, they populated this database by selecting explicit statements of contradiction or negation from web texts.

The above approaches are based on linguistic analysis and textual entailment. In contrast, our approach is based on statistical principles and intended for a large-scale operation, where pairwise comparisons of texts may not be computationally efficient. In addition, we are considering a time dimension for contradiction, which allows us to introduce such new types as, for example, change of opinion (asynchronous contradiction). To the best of our knowledge, this problem has not been studied so far.

Problems related to the identification and analysis of contradictions have also been studied in the context of social networks and blogs. A recent work by Liu et al. [10] introduces a system that allows to compare contrasting opinions of experienced blog users on some topic. In contrast, we take into account the opinions of all web users, regardless of their expertise. Clustering accuracy as an indicator of blogosphere topic convergence was proposed by [17]. By analyzing how accurate clustering is in different time intervals, one can estimate how correlated, or diverse, blog topics are. Such an approach can also be adapted to opinion contradictions as well, by replacing topic feature vectors by sentiment feature vectors. Our work goes beyond trend analysis by automatically recognizing contradictions regarding some topic within and across documents.

Analysis of product reviews is another opinion mining task that is close to contradiction analysis. A system for mining the reputation of products in the Web is described in [11]. A similar approach is proposed by the Opinion Observer system [9] that focuses on summarizing the strengths and weaknesses of a particular product. Even though the above studies consider both positive and negative opinions, they do not aggregate these two classes. In our approach, we describe an effective way for performing this aggregation, which leads to more insights on the user opinions.

Chen et al. [2] study precisely the problem of conflicting opinions on a corpus of book reviews, which they classify as positive and negative. Their main goal is to identify the most predictive terms for the above classification task, and visualize the results for manual inspection. However, the results are only used to visualize opposite opinions without further aggregation. It is up to the user to visually inspect the results and draw some conclusions. In contrast, we propose a systematic and automated way of performing sentiment aggregation, revealing contradictions, and analyzing the evolution of these contradictions over time.

3. PROBLEM DEFINITION

The problem we want to solve in this paper is the efficient detection of contradicting opinions[2] (on specific topics).

Usually, a particular source of information covers some general topic T (e.g., *health, politics*) and has a tendency to publish more texts about one topic than another. Yet, within a text, an author may discuss several topics. When using the term 'text' we refer either to the entire web document or its individual sentences. With the term sentence we assume a particular piece of text expressing an opinion about a certain topic, which can not be split into smaller parts without breaking its meaning. For each of the topics discussed in some text, we wish to identify the sentiment expressed towards it. In this study, we restrict ourselves to identifying and recording the intensity of these sentiments, which we represent as numbers. In the following, we refer to sentiment polarity simply as *sentiment*.

DEFNINITION 1 (SENTIMENT). *The sentiment S with respect to a topic T is a real number in the range $[-1, 1]$ that indicates the polarity of the author's opinion on T expressed in a text. Negative and positive values represent negative and positive opinions respectively, while the absolute value of sentiment represents the strength of the opinion.*

Apart from computing sentiments for individual texts, we also need to compute the polarity on some topic aggregated over multiple texts (that may span different authors, as well as time periods).

DEFNINITION 2 (AGGREGATED SENTIMENT). *The Aggregated Sentiment μ_S expressed in a collection of documents $\mathcal{D}$ on topic T, is defined as the mean value over all individual sentiments assigned in that collection. μ_S is defined on the same range of $[-1, 1]$ as sentiments and calculated as follows: $\mu_S = \frac{1}{n}\sum_{i=1}^{n} S_i$, where n is the cardinality of $\mathcal{D}$.*

By comparing the sentiment values of different collections of texts, contradictions are identified as follows.

DEFNINITION 3 (CONTRADICTION). *There is a contradiction on a topic, T, between two groups of documents, $\mathcal{D}_1, \mathcal{D}_2 \subset \mathcal{D}$ in a document collection $\mathcal{D}$, where $\mathcal{D}_1 \bigcap \mathcal{D}_2 = \varnothing$, when the information conveyed about T is considerably more different between $\mathcal{D}_1$ and $\mathcal{D}_2$ than within each one of them.*

In the above definition, we purposely not specify exactly what it means for a sentiment value to be very different from another one. We define contradiction on a *pairwise* basis, where we evaluate the disagreement between two groups of documents in a collection. In this case, the similarity of information within each group serves as a reference point, providing a basic disagreement level. This definition can lead to different implementations, and each one of those will have a slightly different interpretation of the notion of contradiction. We argue that our definition captures the essence of contradictions, without trying to impose any of the specific interpretations. Nevertheless, in Section 4, we propose a specific method for computing contradictions, which incorporates many desirable properties.

When identifying contradictions in a document collection, it is important to also take into account the time in which these documents were published. Let $\mathcal{D}_1$ be a group of documents containing some information on topic T, and all documents in $\mathcal{D}_1$ were published within some time interval t_1. Assume that t_1 is followed by time interval t_2, and the documents published in t_2, $\mathcal{D}_2$, contain a conflicting piece of information on T. In this case, we have a special

[2] For the rest of this document we will use the terms *sentiment* and *opinion* interchangeably.

type of contradiction, which we call *Asynchronous Contradiction*, since $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to two different time intervals. Following the same line of thought, we say that we have a *Synchronous Contradiction* when both $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to a single time interval, t.

In order to detect contradicting opinions in collections of texts, we first need to determine all the different topics and then calculate the corresponding sentiments.

PROBLEM 1 (SINGLE-TOPIC CONTRADICTION DETECTION). *For a given time interval τ, and topic T, identify the time regions of a predefined size w, where a contradiction level for T is exceeding some threshold ρ.*

The time interval, τ, is user-defined. As we will discuss later, the threshold, ρ, can either be user-defined, or automatically determined in an adaptive fashion based on the data under consideration. We can also determine all the topics in a dataset that are involved in contradictions, as follows.

PROBLEM 2 (ALL-TOPICS CONTRADICTION DETECTION). *For a given time interval τ, identify topics T, which have high contradiction level, or large number of contradicting regions above some threshold.*

The latter problem is interesting if we want to consider the popularity of certain web topics. Frequent contradictions may indicate "hot" topics, which attract the interest of the community. Due to space limitations, in this paper we only discuss a solution to the first problem, since a solution to the second one is its direct extension. Though, the approach we propose in this work is general, and can lead to solutions for several other variations of the above problem, such as detection of topics with periodically repeating contradictions or with the most frequently alternating *Aggregated Sentiment*.

4. CONTRADICTION DETECTION

Given the problems described before, we propose a three step approach to contradiction analysis, that includes:

- Detection of topics for each sentence,
- Detection of sentiments for each sentence-topic pair, and
- Analysis of sentiments for topic across multiple texts.

Steps one and two can be achieved using existing methods, or adaptations of existing methods. We will refer to these steps as 'preprocessing' and describe them briefly in the following. The focus of this paper is then the contradiction detection approach.

4.1 Preprocessing

For identifying topics per sentence, we apply the Latent Dirichlet Allocation (LDA) algorithm [1], which we extended to work on the sentence level [4]. So sentences are considered as input documents for the LDA and assigned with several most probable topics.

Then, for each sentence-topic pair we assign a continuous sentiment value in the range [-1;1] that indicates a polarity of the opinion expressed regarding the topic. For the sentiment assignment step, we use an existing tool for fine-grained opinion analysis [7]. Nevertheless, this tool can be replaced by any other suitable one that calculates continuous sentiment values at a sentence level. Then we average sentiments over text's sentences having the same topic, to get one sentiment value for each topic in a text.

Based on the analysis described so far, we can now describe our approach for contradiction detection with respect to different topics. In the following paragraphs, we first propose a novel contradiction measure, and then describe two simple approaches aiming at detecting contradictive periods in time.

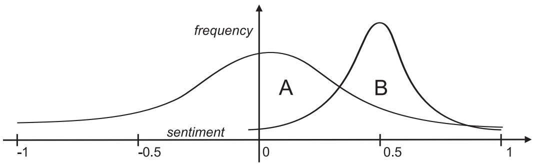


Figure 1: Example of two possible sentiment distributions.

4.2 Measuring Contradictions

In order to be able to identify contradicting opinions we need to define a measure of contradiction. Assume that we want to look for contradictions in a shifting time window[3] w. For a particular topic T, the set of documents $\mathcal{D}$, which we use for calculation, will be restricted to those, that were posted within the window w. We denote this set as $\mathcal{D}(w)$, and n as its cardinality, $n = |\mathcal{D}(w)|$.

In this example, a value of aggregated sentiment μ_S close to zero implies a high level of contradiction because positive and negative sentiments compensate each other. A problem with the above way of calculating topic sentiment arises when there exists a large number of documents with very low sentiment values (neutral documents). In this case, the value of μ_S will be drawn close to zero, without necessarily reflecting the true situation of the contradiction. Therefore, we suggest to additionally consider the variance of the sentiments along with their mean value. The sentiment variance σ_S^2 is defined as follows:

$$\sigma_S^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_S)^2 \qquad (1)$$

According to the above definition, when there is a large uncertainty about the collective sentiment of a collection of documents on a particular topic, the topic sentiment variance is large as well.

Figure 1 shows two example sentiment distributions. Distribution A with μ_S close to zero and a high variance indicates a very contradictive topic. Distribution B shows a far less contradictive topic with sentiment mean μ_S in the positive range and low variance. For example, a group of documents with μ_S close to zero and a high variance (distribution A on the Figure 1) will be very contradictive, and another group with sentiment μ_S shifted to negative or positive with low variance is likely to be far less contradictive (distribution B on the Figure 1). We note that neither the mean nor the variance can be used independently to identify contradictions. For example, a fairly large variance among sentiments does not lead to a contradiction when only positive or negative sentiments are present. Moreover, a zero mean value may occur even when all posts are neutral, which once again does not indicate a contradiction. When assuming a large number of neutral sentiments in the collection, we have two opposite trends: the average sentiment moves towards zero and sentiment variance decreases. If these trends will compensate each other, the neutral documents would not affect the contradiction value much.

Evidently, we need to combine mean and variance of sentiments in a single formula for computing contradictions. Then, the contradiction value C can be computed as:

$$C = \frac{\sigma_S^2}{(\mu_S)^2} \qquad (2)$$

where μ_S is squared so that its units are the same as of σ_S^2.

This formula captures the intuition that contradiction values should be higher for topics whose sentiment value is close to zero, and sentiment variance is large. Nevertheless, the contradiction values

[3] Without loss of generality, in this work we consider windows of days, weeks, months, and years.

generated by this formula are unbounded (i.e., they can grow arbitrarily high as μ_S approaches zero), and does not account for the number of documents n. This latter point is important, because in the extreme where $\mathcal{D}(w)$ contains only two documents with opposite values, C will be very high, and will compare unfavorably to the contradiction value of a different set of T documents with a much higher cardinality.

Incorporating to the contradiction formula the observations made above, we propose the following final formula for computing contradiction values:

$$C = \frac{\vartheta \cdot \sigma_S^2}{\vartheta + (\mu_S)^2} W \qquad (3)$$

In the denominator, we add a small value, $\vartheta \neq 0$, which allows to limit the level of contradiction when $(\mu_S)^2$ is close to zero. The nominator is multiplied by ϑ to ensure that contradiction values fall within the interval $[0; 1]$. Figure 2(c) shows how a contradiction value depends on ϑ in the denominator. Smaller ϑ values emphasize contradiction points with μ_S close to zero, for example changes of opinion. Larger ϑ values mask this difference, making levels of contradictions more equal. In this study, we used a value of ϑ set at 5% of the expected value of squared sentiment mean, which was effective for its purpose, exhibiting a stable behavior across datasets, without distorting the final results.

W is a weight function aiming to compensate the contradiction value for the varying number of documents that may be involved in the calculation of C. The weight function is defined as:

$$W = (1 + exp(\frac{\overline{n} - n}{\beta}))^{-1} \qquad (4)$$

where the constant $\overline{n}$ reflects the average number of topic documents in the window, and β is a scaling factor. This weight function provides a multiplicative factor in the range $[0; 1]$ Using W we can effectively limit C when there is a minor number of documents, as well as when this same number of documents increases significantly. What W achieves is essentially a normalization of the contradiction values across different sets of documents, allowing them to be meaningfully compared to each other.

Figure 2 shows the operation of the proposed contradiction function. To demonstrate this, we generated a time series of sentiments for a period of 8000 time units composed of 8000 normally distributed points, half of which follow a custom trend with dispersion 0.125 and another half with dispersion 0.25 and median 0 is acting like noise. Time stamps of all points followed the Poisson distribution with parameter $\lambda = 1$ time units. We have chosen these distributions because they are simple but still resemble the real data. The graph at the top (Figure 2(a)) shows generated sentiments. The bold line in this graph depicts the custom trend, showing an initial positive sentiment that later changes to negative (at time instance t_1), which represents a change of sentiment. There is also a point around time instance t_2, where the sentiments are divided between positive and negative, a situation representing a simultaneous contradiction. Using this dataset, we verify the ability of the C function to capture the planted contradictions.

As can be seen in Figure 2(b), μ_S closely captures the aggregate trend of the raw sentiments. The following two graphs in the figure show the contradiction value, calculated using a sliding window of size 500 and 1000 time units. When we use a window of small size (Figure 2(c)), C correctly identifies the two contradictions at points t_1 and t_2, where the values of C are the largest. Using a larger window has a smoothing effect in the values of C (Figure 2(d)). Nevertheless, we can still identify long-lasting contradictions: In this case, the largest value of C occurs at time instance t_1, corresponding to a change of sentiment that manifests itself across the entire dataset.

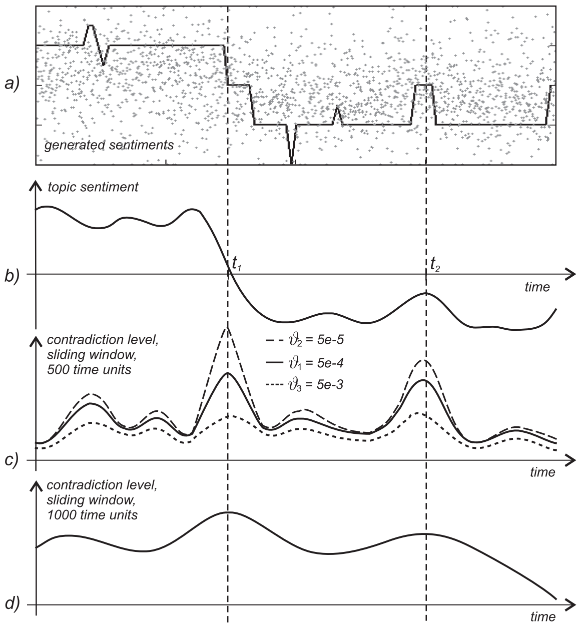


Figure 2: Example of contradiction values computed from a synthetic dataset with two planted contradictions.

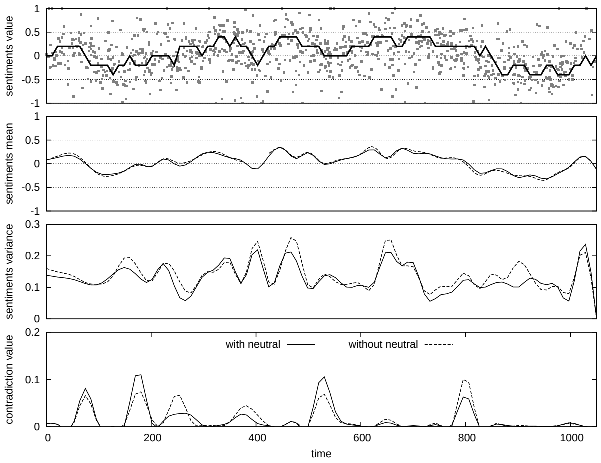


Figure 3: The effect of neutral sentiments on contradiction.

Subjective sentences take a considerably small part in the text when compared to objective statements. So neutral sentiments usually shift the aggregate sentiment towards zero, masking contradictions. Our contradiction formula is designed to compensate such effects by exploiting the sentiment variance.We demonstrate such behavior on another synthetic dataset shown in Figure 3. The bottom graph shows that the proposed formula can successfully identify the main contradicting regions, both with or without neutral sentiments.

5. STORING CONTRADICTIONS

So far we have described a technique for processing web documents to extract sentiments on various topics, and subsequently to use this information in order to identify contradictions. But our final goal is to identify contradictions in large collections of documents, what requires scalable methods. To this end, we demonstrated the need to analyze sentiment information on each topic across different time windows. Assuming this requirement, scalability may be achieved by storing pre-computed values for windows of different size. We now turn our attention to the problem of organizing all these data in a way that will allow the efficient detection of contradictions in large collections of data that span very long time intervals.

An important observation is that the Formula 3 that calculates the contradiction values is based on the mean and variance of the topic sentiment. Remember that aggregated sentiment and sentiment variance can be written as the following:

$$\mu_S = \frac{1}{n}\sum_{i=1}^{n} S_i; \qquad \sigma_S^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_S)^2 = \frac{1}{n}\sum_{i=1}^{n} S_i^2 - \mu_S^2$$

In the formula above, n is the number of documents published on topic T in a specific time window (see Definition 2).

We now define the first- and second-order moments of the topic sentiment as $M_1 = \sum_{i=1}^{n} S_i$ and $M_2 = \sum_{i=1}^{n} S_i^2$, respectively. Based on the above discussion, and using the sums M_1 and M_2, we can rewrite Formula 3 as follows:

$$C = \frac{nM_2 - M_1^2}{\vartheta n^2 + M_1^2} W \tag{5}$$

The above form of the contradiction values formula gives us additional flexibility, since we can now compute the contradiction of a large time window by composing the corresponding values from the smaller windows contained in the large one. We can therefore build data structures that take advantage of this property.

In the next paragraphs, we describe such a data structure, and we show how it can be used to identify contradictions. We also demonstrate that it can be easily maintained in an incremental fashion when new documents are added in the system.

5.1 TimeTree for Contradictions

The need to analyze contradictions at different time granularities predicts a hierarchical structure for contradiction storage. There is a number of ways to organize contradiction values by time. The first solution is to store a time-tree structure for each topic separately. It allows to achieve a scalability on the number of topics, and has a good performance when looking for contradictions at a single topic, but also brings larger update costs, because for each text the storage needs to be parsed as many times as there are topics in that text. Also it makes all-topic queries extremely ineffective, because for each topic we need to navigate through a time structure to find the right interval. The second solution that we propose is to store contradiction values for different topics under the same time-tree structure.

We introduce the TimeTree for managing the information on sentiments and contradictions. The TimeTree is organized around the sentiment moments, M_1 and M_2, and a hierarchical segmentation of time, as outlined in Figure 4. In this example, the time windows are organized on days, weeks, months, and years (though, other hierarchical time decompositions are applicable as well). Using this kind of structure, we can answer queries on *adhoc* time intervals, by dynamically computing the contradiction values based on Formula 5. In the following, we will refer to the levels of the TimeTree as the different *granularities* of the time decomposition, the root node having granularity 0.

Each node in the TimeTree corresponds to a time window, and summarizes information for all documents, whose timestamp is contained in this time window.

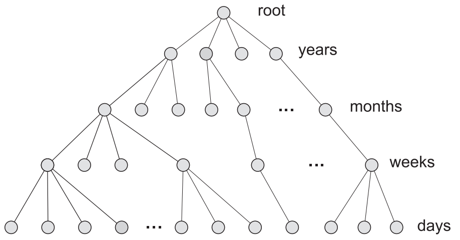


Figure 4: Logical representation of the TimeTree.

5.2 Querying for Contradictions

When trying to detect contradictions, we would like to identify those that have a contradiction value above some threshold. The intuition is that these contradictions are going to be more interesting than the rest in the same time interval. An obvious solution in this case is to define some fixed threshold, ρ, and only report the contradictions above this threshold. We refer to this solution as *fixed threshold*. However, by adopting the above solution, we cannot normalize the threshold to better fit the nature of the data within each time window (that may vary over time and across topics).

In order to address this problem, we propose an *adaptive threshold* technique, which computes a different threshold for each topic and time window as follows. The adaptive threshold ϱ_w for a topic T in time window w is based on the contradiction value C_{w_p} that has been calculated for T in the parent time window of w, w_p, and is defined for each time window and topic as $\varrho_w = p \cdot C_{w_p}$, $0 < p < 1$. In our experience with real datasets, p values between $0.5 - 0.7$ work well. In this work, we use $p = 0.6$.

Note that we cannot achieve the same result by using *top-k* queries (though, they can be complementary to our approach). The reason is that adaptive threshold does not impose a strict limit on the number of contradictions in the result, and can thus report the entire set of interesting contradictions within some time interval.

5.3 Updating the Contradictions

As discussed earlier, the nature of the contradiction function (Formula 5) and the TimeTree nodes allows us to incrementally maintain the TimeTree in the presence of updates. When new collections or individual documents are analyzed, their contribution to the contradiction of the corresponding topics and time windows in the TimeTree can be easily taken into account by updating the set of relevant $\{n, M_1, M_2\}$ values in the nodes of the tree.

In order to reduce update costs, we propose first to accumulate several updates and then submit them in a batch. When new documents arrive, as a preprocessing step, they are aggregated in time windows of the finest granularity of the TimeTree. Then, these aggregated values are used to update the counts and topic sentiment moments of all TimeTree nodes containing respective time windows.

The update cost for each batch of aggregated documents depends on the depth of the TimeTree, d, and the number of topics, $|T|$ (in the worst case), that participate in the time windows relevant to the update. Thus, the complexity can be expressed as $O(d \cdot |T|)$

6. EXPERIMENTAL EVALUATION

As mentioned earlier, the contradiction detection problem has not been considered before. Therefore, no annotated data set is available to measure the quality of the proposed approach in terms of accuracy. Anyway, we applied the algorithm to real world data sets and run several experiments with settings and results described in this section. The objectives of these experiments are to: Analyze the quality of the approach; Study its usefulness from a user perspective; Study the performance of the introduced approach.

6.1 Corpus Description

Our algorithms are applied to a data set of drug reviews collected from the DrugRatingz website[4], a data set of comments to YouTube videos from L3S [15] and a dataset with comments on postings from Slashdot, provided for the CAW2 workshop[5].

The first dataset contains 2701 positive, 352 neutral and 1616 negative reviews for 477 drugs. These reviews are provided by persons that took a specific drug. They describe their personal experience with the drug including contra-indications that occurred.

The second dataset contains approximately 6 million comments to YouTube videos, with an average number of comments per each video of five hundred. Unlike texts in review datasets which usually contain opinions specific to a topic, some of these comments contain information irrelevant to a topic, thus introducing extra noise to sentiment detection.

Our third dataset, Slashdot, is from a popular website for people interested in reading and discussing about technology and its ramifications. It publishes short story posts which often incite many readers to comment on them and provoke discussions that may trail for hours or even days. It contains about 140,000 comments under 496 articles, covering the time period from August 2005 to September 2006. Compared to usually brief comments on YouTube videos, comments from the latter dataset may span for several paragraphs and typically contain many objective statements.

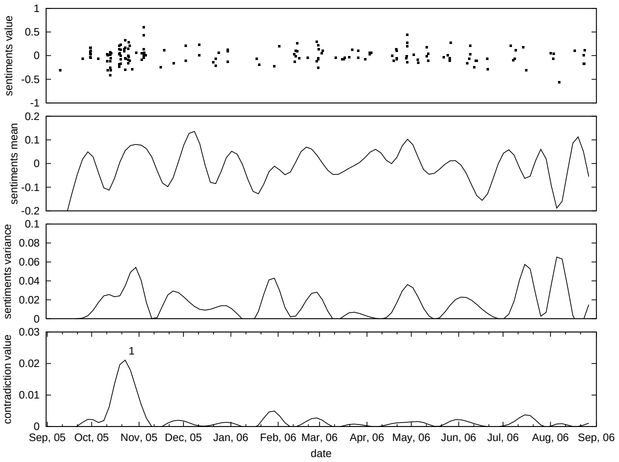
6.2 Evaluation of Contradictions

We now apply the introduced contradiction analysis approach to our datasets. In Figure 5, the top graph depicts the raw sentiment values for the topic "internet government control" taken from the Slashdot dataset, for the time interval September 2005 to September 2006. The following graphs show the aggregated sentiment and variance (two middle graphs), and contradiction values (bottom graph) for the above topic and time interval. Contradiction values have been calculated using a time window of ten days. Note that contradiction values are high for the time windows where topic sentiment is around zero and variance is high, which translates to a set of posts with highly diverse sentiments. These situations are not easy to identify either with a quick visual inspection of the raw sentiments, aggregated sentiments or sentiment variance.

The analysis shows that in this time interval there is one major contradiction (marked 1 in the bottom graph of Figure 5). This contradiction discusses the pros and cons of a law that would give the government more power in controlling the internet traffic, especially personal correspondence. Minor peaks in contradiction level here correspond to the discussion of a possible transfer of jurisdiction and control over top-level domains to United Nations. The table below shows extracts from several opposing posts that contributed to this contradiction. By taking a closer look at the corresponding weblog posts, we find out that the discussion is about restricted internet access and its advantages, while other contradictions contain a general discussion on the possibility of organizing the content by several top-level domains and restricting access to them.

Another example of contradicting posts may be observed in Figure 6, which illustrates conflicting opinions for the topic "Yaz"[6] for a selected time interval. In this case, there was an opinion disagreement on the effectiveness and possible side-effects of this drug.

Evidently, all the discovered contradictions correspond to discussions expressing different points of view on the same topic, and having an automated way of identifying them can be very useful.



PRO: It would be helpful for restricting the flow of information, which is a double edged sword.
PRO: I suppose we better wrap a firewall around our country and not let those damn foreigners access to our internet.
CONS: And what exactly does a neutral Internet do? It takes away the right of anyone who lays down the wires or installs the access points to control what goes through their network. My point: don't complain about taking rights away when you advocate to take rights away.
CONS: While it sounds like a decent idea, I'm really all for the whole uncensored and unregulated internet. I really like my internet the way it is.
CONS: Sure, they can ruin Internet inside USA, but the rest of the world couldn't care less.
CONS: We don't need the FCC regulating the Internet. Not for "neutrality" or any other excuse someone can think of.

Figure 5: Mean, variance and contradiction values of sentiments for the topic "Internet government control".

6.3 Evaluation of Usefulness

In the following paragraphs we describe a user study which we conducted in order to evaluate the effectiveness and usefulness of our approach for the task of contradiction discovery.

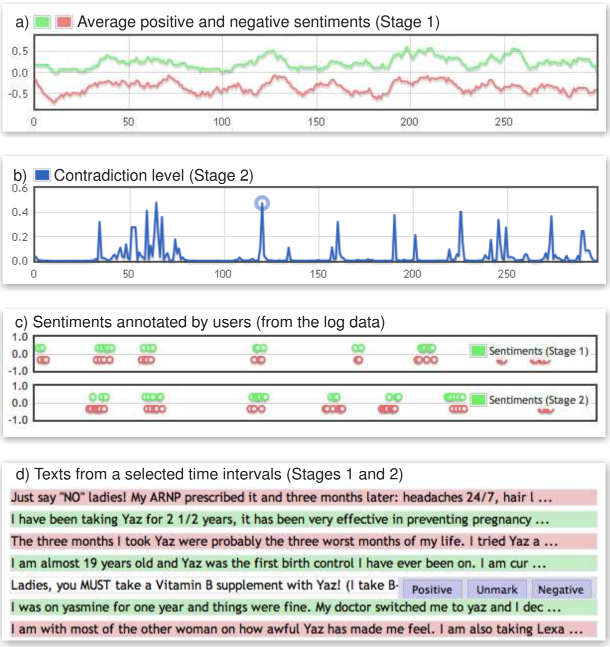
In our usefulness evaluation, we used four datasets corresponding to opinionated posts for four topics extracted from three diverse real datasets (refer to Table 1). For each topic, we selected a varying number of posts, spanning in time from one to almost three years. The shortest list contained 60 posts, and the largest about 480. Moreover, the quality of posts for topics also differed a lot. The drug review datasets contained primarily brief and concise opinions about drugs; Slashdot topics featured large and detailed comments, with an average size of several paragraphs; YouTube comments were, on the contrary, short and often off-topic.

The group of users consisted of eight persons (PhD students at the University of Trento), and the experiment was conducted as follows. Users were asked to detect groups of contradicting posts for each of the topics in the above datasets (and label the positive and negative posts). We provided users with a web application that featured two approaches to help them identify time-intervals with potentially contradicting posts (see Figure 6): The first approach (marked as "stage 1" in the figure), based on the visualization method proposed by Chen et al. [2], displays to users the intensity over time of the positive and negative sentiments expressed in the posts (Figure 6(a)). The second approach (marked as "stage 2" in the figure) is based on the method proposed in this study, and displays to users a graph that marks the time points at which contradictions were automatically detected (Figure 6(b)). Using our tool, the users could see the time intervals that our tool had identified as contradictory, and could therefore, focus their exploration in these

[4] http://drugratingz.com

[5] http://caw2.barcelonamedia.org/

[6] *Yaz* is a drug for contraception



Just say "NO" ladies! My ARNP prescribed it and three months later: headaches 24/7, hair l ...

I have been taking Yaz for 2 1/2 years, it has been very effective in preventing pregnancy ...

The three months I took Yaz were probably the three worst months of my life. I tried Yaz a ...

I am almost 19 years old and Yaz was the first birth control I have ever been on. I am cur ...

Ladies, you MUST take a Vitamin B supplement with Yaz! (I take B-

I was on yasmine for one year and things were fine. My doctor switched me to yaz and I dec ...

I am with most of the other woman on how awful Yaz has made me feel. I am also taking Lexa ...

Figure 6: Annotation page for the dataset "Yaz" demonstrating opposite opinions.

regions. Figure 6(d) shows some posts in a time-interval, which have been marked with positive (green) and negative (red) sentiments. These sentiments values are also illustrated in the overall time-line, depicted in Figure 6(c). In order not to favor any of the two approaches, in our experiments we alternated the approach required to be completed first.

For both approaches, we measured the average time, T_1 and T_2, and the average number of time-intervals examined by the users during the search, N_1 and N_2, needed to identify a single contradiction. Additionally, we asked users to rate the overall difficulty, D_1 and D_2, of completing the task when using each one of the two approaches, according to the following scale: 1- very difficult; 2 - somewhat difficult; 3 - normal; 4 - somewhat easy; 5 - very easy.

The aggregated results (averaged over all the users) of our evaluation are reported in Table 1. We report the improvements[7] we measured when our approach was used (stage 2), compared to the alternative approach (stage 1), computed as follows: $\Delta D = D_2/D_1$, $\Delta T = T_2/T_1$, and $\Delta N = N_2/N_1$.

We observe that when users employed our approach in order to detect contradictions, they were able to identify contradictions faster, requiring 23% less time on average (ranging between 7% and 40%). The biggest improvement was for the topic "Ambien"[8] ($\Delta T = 0.60$), which had a few contradicting posts visible using our approach, but otherwise hard to discover. Our approach also led to a reduction by 28% of the time-intervals examined in order to identify contradictions (ranging between 12% and 42%). The largest reductions were observed for the topics "Zune HD" and "Internet Control" ($\Delta N = 0.62$ and 0.58, respectively), which contained several posts that did not take a position, or were off topic. The average difficulty ratings were also favorable for our approach, which was consistently being marked as more helpful. This difference was most pronounced for the "Zune HD" topic ($\Delta D = 2.07$), which involved many posts. In this case, going through the posts was not easy, and our approach allowed users to focus their search and identify the contradicting posts.

Dataset	Topic name	Size	ΔD	ΔT	ΔN	P_1	P_2	ΔP
Drug Ratingz	Ambien	60	1.50	0.60	0.88	0.70	0.81	1.20
	Yaz	300	1.58	0.93	0.78	0.75	0.95	1.32
Slashdot	Int. control	159	1.17	0.89	0.58	0.37	0.63	2.14
YouTube	Zune HD	472	2.07	0.68	0.62	0.36	0.61	2.09
Average			**1.58**	**0.77**	**0.72**	**0.55**	**0.75**	**1.69**

Table 1: Evaluation results for different topics.

Finally, in Table 1 we report an additional measure of usefulness: since both approaches aim at guiding the users to the time-intervals that are most promising for containing contradictions, we computed the percentage, P_1 and P_2, of the examined time-intervals that led to the identification of a contradiction, as well as the improvement of our approach when compared to the alternative, $\Delta P = P_2/P_1$. Even though the approach by Chen et al. [2] (stage 1) was not designed with this measure in mind, in the case of our approach, this measure is indicative of its precision since it measures how many of the automatically identified contradictions were real ones (i.e., verified by the users). The results show that our approach was always more successful in suggesting to users time-intervals that contained contradictions, with an overall average success rate of 75%, and as high as 95% (topic "Yaz").

The above results demonstrate that our approach can successfully identify contradictions in an automated way, and quickly guide users to the relevant parts of the data.

6.4 Evaluation of Scalability

We evaluate the scalability of the TimeTree for solving Problems 1 and 2, using a relational database implementation, where information is stored in a single table that contains contradiction values for each topic with respect to time intervals of different granularities. This implementation leads to simple and efficient SQL queries for detecting interesting contradictions. Remember that in the topic contradiction problem (Problem 1) we want to identify the contradictions and corresponding time windows of a single topic within some time interval, while in the all topic contradictions problem (Problem 2) we are interested in doing the same for all topics.

During this study, parameters of the contradiction formula were at their default values as described in Section 4. Changing formula's parameters will enlarge or reduce the number of contradictions being detected, but the computational efficiency will be the same. Performance of our approach does not depend on the value of threshold because we are not storing pre-computed contradiction values, and so the database is unable to apply indices or filtering on this parameter. Fixed and adaptive threshold approaches, however, return slightly different sets of contradictions. The first one returns largest contradictions themselves, and the second returns contradictions that are greater than p-times values of their respective parent intervals. The value of p was empirically set at 0.6 to return a result set with an average size equal to the one when using a fixed threshold. This allows us to compare the relative performance of both methods.

To test the performance of our solutions, we generated sets of 25 single-topic and all-topics queries (corresponding to the Topic and Time Interval Contradictions problems, respectively), using granularities and topic ids drawn uniformly at random. In these experiments, we used 1,000 topics. We measured the time needed to execute these queries against the database as a function of the time

[7] We omit presenting the detailed results for all parameters measured and each approach due to lack of space.

[8] *Ambien* is a drug for treating insomnia

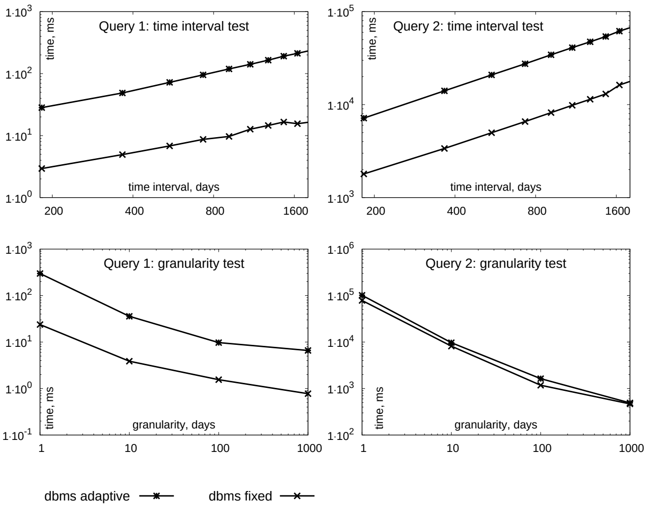


Figure 7: Scalability of single-topic and all-topics queries.

interval, τ, and the granularity of the time windows (Figure 7). We report results for both the fixed and the adaptive thresholds.

The adaptive threshold queries require in all cases more time since the threshold in this case has to be computed based on the contradiction value of the parent time window, which incurs more computation. This difference is pronounced for the database implementation, because it involves an extra join for obtaining the parent time window.

We observe that both single-topic and all-topics queries (see Figures 7(a-b)) scale linearly with the size of τ. This confirms our analytic results, and is explained by the fact that the queries have to return contradictions for all time windows (of a specific granularity) that are contained in τ. For single-topic queries with fixed threshold, the database is able to use all its indices (i.e., on topic id, time windows, and granularity) to answer the queries, therefore, achieving very fast response times.

Figures 7(c-d) depict the time results when we vary the granularity of the time windows specified by the queries. Increasing the granularity translates to larger time windows (i.e., moving up in the time hierarchy) and a smaller number of time windows for the same time interval. Thus, response times get lower.

7. DISCUSSION

The problem considered in this paper is new, in the sense that it considers contradictions on the large scale, while taking time into account (i.e., we consider the timestamps of the texts, as opposed to treating the text collections as sets). An approach that relies upon sentiment information and that exploits data engineering methods to detect such contradictions in texts at a large scale has been introduced and evaluated.

The evaluation of our approach on various datasets proved its ability of discriminating highly contradicting regions provided with a sequence of sentiments on some topic. Being scalable and computationally efficient, it can serve as a preliminary step for more sophisticated contradiction analysis, identifying the most interesting points for further processing.

An important feature of our contradiction detection method is its ability to operate on data with neutral sentiments. The contradiction formula we propose shows almost the same performance with or without neutral sentiments, allowing it to incorporate sentiment detection algorithms of different types.

As was mentioned previously, to build the contradiction formula we used such values as mean and variance. We believe that the effectiveness of our approach increases with the growing scale, relying on the fact that representativeness of statistical metrics also increases when larger number of samples is involved in computation. Moreover, tests on the synthetic data proved our formula's stable behavior in the presence of noise.

Finally, we note that we are aware that the evaluation of our (and related) approach to contradiction detection is still limited with respect to the precision and recall measures. The main reason for this is the absence of a benchmark dataset, and the difficulty in creating one. We are currently working toward such a dataset, suitable for testing different algorithms in this area.

8. CONCLUSIONS

In this paper, we proposed an approach to detect contradictions in documents, which is the first general and systematic solution to the problem. The experimental evaluation, with synthetic data and three diverse real-world datasets, as well we the user-study, demonstrate the applicability and usefulness of the proposed solution.

We are currently working on extending our approach so that it can work in an online mode. This will enable us to continuously monitor opinions in real-time.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *JMLR*, 3, 2003.

[2] C. Chen, F. Ibekwe-SanJuan, E. SanJuan, and C. Weaver. Visual analysis of conflicting opinions. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66, 2006.

[3] M. C. de Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. In *ACL-08: HLT*, pages 1039–1047, 2008.

[4] K. Denecke and M. Brosowski. Topic detection in noisy data source. In *ICDIM*, pages 50–55, 2010.

[5] R. Ennals, B. Trushkowsky, and J. M. Agosta. Highlighting disputed claims on the web. In *WWW*, pages 341–350, 2010.

[6] S. Harabagiu, A. Hickl, and F. Lacatusu. Negation, contrast and contradiction in text processing. In *AAAI*, pages 755–762, 2006.

[7] R. Johansson and A. Moschitti. Reranking models in fine-grained opinion analysis. In *COLING*, pages 519–527. ACL, 2010.

[8] K. Lerman, S. Blair-Goldensohn, and R. Mcdonald. Sentiment summarization: Evaluating and learning user preferences. In *EACL*, pages 514–522, 2009.

[9] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW*, pages 342–351. ACM, 2005.

[10] J. Liu, L. Birnbaum, and B. Pardo. Spectrum: Retrieving different points of view from the blogosphere. In *ICWSM*, pages 114–121, 2009.

[11] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *KDD*, pages 341–349, 2002.

[12] S. Pado, M.-C. de Marneffe, B. MacCartney, A. N. Rafferty, E. Yeh, and C. D. Manning. Deciding entailment and contradiction with stochastic and edit distance-based alignment. In *TAC*, 2008.

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

[14] E. Riloff, J. Wiebe, and W. Phillips. Exploiting subjectivity classification to improve information extraction. In *AAAI*, pages 1106–1111, 2005.

[15] S. Siersdorfer, S. Chelaru, W. Nejdl, and J. San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *WWW*, pages 891–900. ACM, 2010.

[16] M. Tsytsarau, T. Palpanas, and K. Denecke. Scalable discovery of contradictions on the web. In *WWW*, pages 1195–1196, 2010.

[17] I. Varlamis, V. Vassalos, and A. Palaios. Monitoring the evolution of interests in the blogosphere. In *ICDE Workshops*, pages 513–518, 2008.

Faceted Approach To Diverse Query Processing

Alessandro Agostini
Department of Computer Science
Prince Mohammad Bin Fahd University
Al-Khobar, Saudi Arabia
aagostini@pmu.edu.sa

Devika P. Madalli, A.R.D. Prasad
Documentation Research and Training Centre
Indian Statistical Institute
Bangalore, India
{devika,ard}@drtc.isibang.ac.in

ABSTRACT

This paper presents a formal framework for implementing a query refinement method. The method uses general principles of facet analysis. Two key notions are advanced and discussed: diversity and focus. Diversity refers to the information needs of a querying user; it is captured by the notion of 'facet'. A focus refers to how diversity is captured from the documents as organized by the user; it provides a kind of context to the user query. The method is situated within the formal framework of the smallest propositionally closed description logic $\mathcal{ALC}$, thereby betting that $\mathcal{ALC}$ provides us with a suitable SAT solver to implement a facet engine, which is the main component of our method.

Categories and Subject Descriptors

H.3.7 [**Information Systems**]: Digital Libraries; I.2.4 [**Computing Methodologies**]: Knowledge Representation Formalisms and Methods

General Terms

Design, Human Factors, Algorithms.

Keywords

Query refinement, facet-based search, text-based search, context-based search, user issues, description logic.

1. INTRODUCTION

Classical libraries had systems that processed subjects or domains and built representations such as subject indices. Among these system, the Colon Classification System (CCS) first proposed by S.R. Ranganathan [20] is currently widely used by almost all Indian libraries. The CCS had enough contextual information in the method of facetisation and synthesis so that it formed a semantic formalisation of the domain scope of the library collections.

In order to digitize CCS and similar facet-based systems, Prasad and Guha [18] demonstrate the applicability of faceted schema in describing resources in web directories and annotating resources in digital libraries using SKOS/RDF representation to express DEPA strings, according to faceted theory by Ranganathan [20] and DEPA facet analysis [7]. On the other hand, current keyword-based querying methods does not use DEPA strings to represent web directories and annotating resources in digital libraries, so they seem inadequate to search over digital repositories organized according to CCS and similar faced-based classification systems.

For answers to be relevant, a user must ask the appropriate query in order to retrieve the desired information and fulfill the information need (IN). For keyword-based search this means that a high number of keywords is necessary to the user to narrow down the search according to her information need. This is due the semantic ambiguity of querying languages, often built upon natural language, as it is the case of keyword-based querying. Unfortunately, the query length of keyword-based search on average is reported to be short, with 90% of the queries being less than four keywords [12]. As a consequence, the ambiguity of the query is somewhat mirrored in the relative relevance of search results [32, 3]; diversity in search results arises [15] and query refinement by the user is often the only solution. To resolve such ambiguity some authors advanced the notion of 'context' in web search, see for instance [14, 10] and references cited therein. However, in contect-based solutions the user is often assumed to know how data and information are organized in the search domain. This is often hard to happen in real-world, distributed scenarios like the Web, due to large amounts of heterogeneous data organized in an unknow structure.

In this paper we present a formal framework wherein we define a method for the extraction of DEPA facets from a user query. The facets are then used to refine the original query for search and retrieval purposes. The method is aimed to suggest the user a list of facets that the user would hardly be aware of by simply typing a keyword-based query into a search engine, without any query context. These automatically suggested new facets can be used by the user, for instance by clicking on one of the new facets, to narrow down the search space by expanding the original user query with the suggested facet.

This paper is organized as follows. In Section 2 we define basic concepts related to facet analysis. In Section 3 we discuss the first step of our method. In Section 4 we build a formal faceted ontology to formalize the focused terms and contexts that we successively process, in Section 5, to produce new facets to be shown to the user for query refinement. After building the faceted ontology and defining the facet engine, in Section 6 we present the three different yet related querying methods we offer to the user; these are keyword-based, by focus, and on subject. In Section 7 we discuss related work. In Section 8 we conclude the paper.

2. FACETS ANALYSIS

Facet analysis is essentially a conceptual analysis of the subject matter, or the topical content of a concept into distinct divisions that together constitute a semantic description of the concept. In order to build the facet repository available to a user to refine a query, in this section we present some elements of facet analysis.

Our facets repository is organized around two main notions of the DEPA paradigm for facet analysis [6, 7]: subjects and facets. A *subject of a concept* is the topical content of the concept, that is, the concept's overall semantics, as defined by the combination of extensional and intensional semantics of the concept term. The definition can be extended to a query, which in its simplest form can be thought of as a finite sequence of concept terms; see subsections 6.1 and 6.3. A *facet* consists of a "group of terms derived by taking each term and defining it, per genus et differentiam, with respect for its parent class." [31, p. 12]. According to Ranganathan [20], each domain is made of distinct divisions or facets that are groups of mutually exclusive concepts and many such facets together constitute a domain. The notion of such facetization has been extended by Bhattacharyya [7] to subject indexing by representing content as a string of fundamental categories DEPA (Discipline, Entity, Property and Action) that are conceptually equivalent to 'facets'. To illustrate, we rely on the following two examples.

EXAMPLE 1. *Consider a document titled 'Improving EU labour market access for Rome'. DEPA facet analysis of the title leads to facets such as: Labour Market (Entity), Access (Action), Rome (Space - from commonly applicable facet schedules across domains). The facet 'Discipline' is extrapolated from faceted document representation, and it is 'Economics'.*

Note that in case a concept would be classified within more than one discipline, as a homonymous or synonymous concept, then all such different combinations of facets are taken into account and presented to the user for further refinement.

EXAMPLE 2. *Consider a document titled 'Treating Apple trees for bacterial disease in Trentino'.*[1] *DEPA facet analysis provides a classification of the document into the following facets: Agriculture (D), Apple Trees (E), Treating (A), Disease (P), and Bacterial (as 'Modifier' to P, cf. [6]).*

We are now ready to define the facet repository for a given context. A *facet repository for a context* $\mathcal{C}$ is the set

$$FR(\mathcal{C}) = \{\langle C : d, e, p, a\rangle \mid C \text{ has DEPA facets } d, e, p, a\},$$

where C is a concept description in description logic $\mathcal{ALC}$ (see subsection 4.2) of a concept or subject of interest in context $\mathcal{C}$, and d, e, p, a are, respectively, a Discipline, Entity, Property and Action in DEPA classification system.

EXAMPLE 3. *Consider the previous two examples. We can assume that 'Improving EU labour market access for Rome' is represented by a concept description C_1, and 'Treating Apple trees for bacterial disease in Trentino' is represented by a concept description C_2 in a context $\mathcal{C}$. The facet repository $FR(\mathcal{C})$ contains $\langle C_1 : Economics, LabourMarket, p, Access\rangle$ for p is unspecified, and $\langle C_2 : Agriculture, AppleTrees, Disease, Treating\rangle$.*

3. FOCUSED TERMS FROM TEXT

In the present work, we apply facetization as a technique to combine extensional and intensional semantics of concepts viz. queries, or equivalently to disclose the subject of concepts and queries to the querying user, for the purpose of query refinement and search assistance. We implement facetization in two related steps: 1. we produce certain "focused terms" from documents organized in a polyhierarchy, and 2. from focused terms we produc new facets to be shown to the user for the purpose of query refinement. We present step 1 in subsections 3.1 and 3.2 in this section, and step 2 in sections 4 and 5.

3.1 Organization of documents

Although our method can be adopted as integral part of digital libraries systems, both for describing the documents collection and for faceted querying over the collection or the web, in this paper we assumed the method assists a *querying user* in query refinement. As the method in this specific application uses a textual collection of documents stored in the user's querying machine, we stipulate the following convention.

CONVENTION 1. *We denote the set of available documents to a querying user by $\mathcal{D}$. All available documents are textual, that is, they can be processed by text information retrieval techniques as the variant of a standard technique discussed in Section 3.*

Intuitively, the domain $\mathcal{D}$ of documents can be thought of as the set of all documents the querying user has classified and stored in the querying machine.

CONVENTION 2. *We assume that the querying user organizes documents in $\mathcal{D}$ by using a 'polyhierarchical classification', or polyhierarchy.*

A *polyhierarchical classification* is a hierarchical classification permitting some concept terms to be listed in multiple categories of a taxonomy, or branches of a hierarchy [16]. An example of polyhierarchy can be found in Figure 1. Note that what makes the hierarchical classification in Figure 1 be polyhierarchical is the concept term 'Apple'.

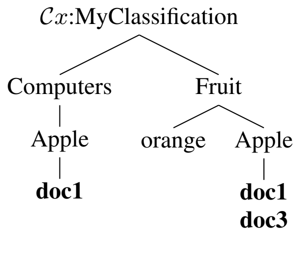


Figure 1: A polyhierarchy, or polyhierarchical context $\mathcal{C}x$.

A subset of documents is organized in 'contexts', each context be organized into related sets of documents. A *context* is a polyhierarchical classification composed by sets of documents, i.e., 'nodes' of the polyhierarchy, called *clusters*, and a relation over the nodes as defined by the polyhierarchy. Typical relations are the binary relations of subsumption, part-of, is-a, among others relations. Each cluster in a given context has a name composed by a finite sequence

[1] Trentino is a Province of the Italian North-east known for the Dolomites and for its quality production of red and yellow apples.

of words from a representation language, often a natural langiage thereby betting that clusters are named by a human—the querying user, who naturally applies her native language for clusters naming. A cluster's name in such representation language is referred to as *concept term*. A *concept* is a concept term provided with a semantics. Two kinds of semantics are provided to a concept term: an extensional semantics, defined over the documents in the cluster named by the concept term; and an intensional semantics, defined by the unique position of the concept term in a given 'focus'.

Contexts provide a way to define finite, ordered sequences of concept terms, each sequence called a *focus*. A *focus* consists of an ordered set of related concept terms, each concept term naming a cluster built upon the collection of documents in $\mathcal{D}$. Intuitively, a focus is a path of concept terms corresponding to a path in a given context. Figure 2 provides an example of both a context (left-hand side) and a focus (right-hand side). With reference to Figure 2, we write $\mathcal{C}x$:Fruit>Trentino>Apple to denote the focus named 'Apple' in the context $\mathcal{C}x$. In boldface are written two documents in the cluster 'Apple': **docRdoc** and **docGtxt**.

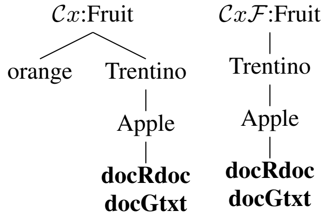


Figure 2: An example of context (left) and focus (right).

3.2 Concept terms grounded in documents

In this section, our goal is to automatically assign a 'label' to every cluster of a given context. Each cluster's label produced by Algorithm 1 below is a finite, simple concatenation of terms with maximum 'weight', extracted by using $\mathsf{Text}\,(\cdot)$. Formally, we proceed as follows.

Let $\mathsf{Text}\,(\cdot)$ be a text extraction function. In this paper, we refer to $\mathsf{Text}\,(\cdot)$ as a standard keywords extraction function, for instance see [25, Sec. 4]. Given a document d, $\mathsf{Text}\,(d)$ listes all the keywords in d, precisely, the most frequent 'tokens'. Applied to a document d, $\mathsf{Text}\,(\cdot)$ produces a set $\mathsf{Text}\,(d)$ of *terms* (or 'keywords'). Let d be any document in $\mathcal{D}$. As terms are defined from documents, from now on we write $k \in \mathsf{Text}\,(d)$ to denote a generic term retrieved by using $\mathsf{Text}\,(\cdot)$ d. Given a document d, we rank a term $k \in \mathsf{Text}\,(d)$ by adapting IR standard TF/IDF ("Term Frequency / Inverse Document Frequency") method [22, 23] to deal with contexts and unique concept terms' *position*, i.e., focus, within a context. Observe that in the following, for a given context $\mathcal{C}$ we write 'C in $\mathcal{C}$' in place of 'C in $\mathcal{C}$' set of clusters' for every cluster C.

Let querying user **u** organizing a context $\mathcal{C}$, cluster C in $\mathcal{C}$, and term $k \in \mathsf{Text}\,(d)$ for a document $d \in \mathcal{D}$ be given. We define the *weight of k in C* as follows:

$$\mathsf{W}_{\mathbf{u}}[k,C] = (\sum_{d\in C}\mathsf{TF}[k,d])\cdot log\frac{Card(FC)}{\mathsf{doCK}_{\mathbf{u}}[k]}, \qquad (1)$$

where $\mathsf{TF}[k,d]$ is the total number of occurrences of k in d, so that $\sum_{d\in C}\mathsf{TF}[k,d]$ is the total number of occurrences of k in C; $Card(FC)$ is the number of focuses in $\mathcal{C}$ with leaf C, and $\mathsf{doCK}_{\mathbf{u}}[k]$ is the total number of clusters in the set

$$\mathcal{C}\setminus\{C' \mid C' \neq C \text{ is a cluster in a focus in } \mathcal{C} \text{ with leaf } C\} \qquad (2)$$

which contain k. Intuitively, (1) says that, in order to represent the extensional semantics of a focus, the importance of a retrieved term for a cluster, i.e., the value of $\mathsf{W}_{\mathbf{u}}[k,C]$, is inversely proportional to the number of different focuses with C as leaf which contain the term.

The label of a cluster C is the most representative term or sequence of terms for the cluster. Now we want compute the label of all clusters of a given context. For doing this, we process all documents stored in each cluster by considering the position of each cluster in the context. To define the process formally, we rely on the following technical definition. Let context $\mathcal{C}$ organize (a subset of) documents in $\mathcal{D}$ and cluster C in $\mathcal{C}$ be given. We define

$$IR(\mathcal{D},\mathcal{C},C) = \{k \in \mathsf{Text}\,(d) \mid d \in C, C \text{ in } \mathcal{C}\}. \qquad (3)$$

We expect that the label of cluster C in (3) is the most representative term or sequence of terms in $IR(\mathcal{D},\mathcal{C},C)$. The most representative term among terms in $IR(\mathcal{D},\mathcal{C},C)$ is the term with the highest weight among all terms in $IR(\mathcal{D},\mathcal{C},C)$ according to weighting measure 1. Formally, a term k in $IR(\mathcal{D},\mathcal{C},C)$ is the most representative for the cluster C in $\mathcal{C}$, and we say that k is a *label of* C, if $\mathsf{W}_{\mathbf{u}}[k',C] \leq \mathsf{W}_{\mathbf{u}}[k,C]$ for all terms k' in $IR(\mathcal{D},\mathcal{C},C)$. A sequence $k_1, k_2, ...k_n$ of terms in $IR(\mathcal{D},\mathcal{C},C)$ is a label of C if (a) $\mathsf{W}_{\mathbf{u}}[k_i,C] = \mathsf{W}_{\mathbf{u}}[k,C]$ for $i = 1,2,...n$, and (b) k is a label of C.

LEMMA 1. *Every cluster C organized by a querying user* **u** *in a context $\mathcal{C}$ has a label if and only if C contains a document d such that* $\mathsf{Text}\,(d)$ *is nonempty.*

To compute a label of every nonempty cluster C of a given context $\mathcal{C}$, we exhibit an algorithm that produces the label l_C of C; see Algorithm 1. Set $IR = IR(\mathcal{D},\mathcal{C},C)$.

Algorithm 1 Context-based cluster labeling.

```
Input: $\mathcal{C}, \mathcal{D} \neq \emptyset$
foreach $C$ in $\mathcal{C}$ with $C \neq \emptyset$ do
  foreach $k \in IR(\mathcal{D},\mathcal{C},C)$ do
    compute $\mathsf{W}_{\mathbf{u}}[k,C]$ according to formula (1) od;
  compute $M = \{k \in IR \mid \forall k' \in IR, \mathsf{W}_{\mathbf{u}}[k',C] \leq \mathsf{W}_{\mathbf{u}}[k,C]\}$;
  Let $n$ be the cardinality of $M$;
  Let $\{k_1, k_2, \ldots, k_n\}$ be the lexicographical ordering of $M$;
  Set $l_0 = \emptyset$;                              /* empty sequence */
  for $i = 1$ to $i = n$ do
    Pick $k_i \in M$;
    Set $l_i = l_{i-1}k_i$ od od;                /* simple concatenation */
Define $l_C = l_n$
Return : set of labels $\{l_C \mid C$ in $\mathcal{C},\ C \neq \emptyset\}$.
```

Observe: 1. If $C \neq \emptyset$ then $IR \neq \emptyset$. 2. The label l_C computed by Algorithm 1 in not unique. In fact, M in Algorithm 1 is assumed to be ordered according to lexicographical ordering. Other orderings of the elements in M are possible and, as a consequence, a different label can be generated from each ordering.

EXAMPLE 4. *To illustrate how Algorithm 1 works, consider the context $\mathcal{C}x$ in Figure 2. The result of applying Algorithm 1 to*

$\mathcal{C}x$, limited to focus $\mathcal{C}x\mathcal{F}$ in Figure 2 is depicted in Figure 3. Each label in the three, e.g., l_{Apple}, is a simple concatenation $k_1...k_n$ of terms extracted by Algorithm 1.

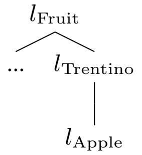


Figure 3: A focus as labeled by Algorithm 1.

We are now ready to define "focused terms." Let a focus $\mathcal{F}$ with concept term C as leaf be given. A *focused term for $\mathcal{F}$* is any term that appears in a label l_C of a cluster C in $\mathcal{F}$. In symbols, the set of focus terms for $\mathcal{F}$ is

$$FT(\mathcal{F}) = \{k \mid k \text{ appears in } l_C, C \in \mathcal{F}\}.$$

A focused term for C is any term that appears in l_C. A focus term for a concept term plays the role of a synonymous, or alias names, of the concept term. As we will see in Section 6, alias names are important to improve keyword-based querying.

4. FACETED ONTOLOGY BUILDING

The result of extracting terms from documents and "facetizing" the concepts of a polyhierarchical classification by using them produces a basic kind of faceted taxonomy, provided that (1) the extracted terms or, often, a proper subset of these [9], are matched with a predefined set of facets, and (2) the clusters in a focus are related to each other by a subsumption relation. For a *faceted taxonomy* consists of: (a) a set of facets, where each facet consists of a predefined set of terms; and (b) a subsumption relation among the terms. In this section we provide the formal framework we need to formalize the focused terms and labeled contexts we have produced by Algorithm 1 by shallowly assuming (2)[2].

4.1 Description Logics

Description Logics (DLs) [5] are a family of logic-based knowledge representation formalisms designed to represent and reason about the knowledge of an application domain in a structured and well-understood way. In this paper, we use a basic description logic, called $\mathcal{ALC}$, thereby betting that $\mathcal{ALC}$ provides us with an efficient SAT solver to implement our facet engine (Section 5). $\mathcal{ALC}$ is the smallest propositionally closed DL, and provides the concept constructors

$$\neg C, C \sqcap D, C \sqcup D, \exists R.C, \forall R.C,$$

as well as concept inclusion (or subsumption) $C \sqsubseteq D$ and concept equality $C \equiv D$, where C, D are concept descriptions and R is a named role. A DL knowledge base (KB) consists of concept axioms (such as concept inclusion and concept equality axioms), role axioms (such as functional role axioms) and assertions of the form $C(a)$, $R(a, b)$ where a and b are named individuals. For the goal of this paper, we use a limited part of $\mathcal{ALC}$'s expressive power; in particular we do not use role axioms and assertions. Moreover, we write concept descriptions in lower case, as concept description from now on are terms extracted by Algorithm 1 from documents as explained. Due to the limitation of space, we do not provide a detailed introduction of Description Logics (DLs), but rather point the reader to [5, 4] and offer the reader an example.

EXAMPLE 5. *Consider the labeled focus in Example 4. We can represent it within $\mathcal{ALC}$ by a set of equality axioms, that we present as labels of the labeled focus in Figure 4. The concept descriptions k_i^j that appear in the tree refer to the focused terms extracted by Algorithm 1 for each concept in the focus; $hasK$ is a named role, which is intuitively interpreted as 'has keyword'. For example, $\exists hasK.k_1^3$ intuitively means that concept term 'Fruit' in focus $\mathcal{F}$:Fruit>Trentino>*Apple *is extended with focused term (keyword) k_1^3. Each equality axiom that appears along the tree defines in $\mathcal{ALC}$ a concept term in $\mathcal{F}$; the focus itself is formalized by the equality axiom: FocusApple $\equiv \text{Apple} \sqcap \exists\text{R}.(\text{Trentino} \sqcap \exists\text{R}.\text{Fruit})$. An $\mathcal{ALC}$ KB for this example is the set of the three equality axioms depicted along the tree plus the equality axiom that defines 'FocusApple' as the 'focus Apple', i.e., the focus $\mathcal{F}$.*

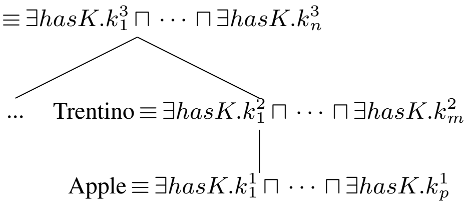

Fruit $\equiv \exists hasK.k_1^3 \sqcap \cdots \sqcap \exists hasK.k_n^3$
...
Trentino $\equiv \exists hasK.k_1^2 \sqcap \cdots \sqcap \exists hasK.k_m^2$
Apple $\equiv \exists hasK.k_1^1 \sqcap \cdots \sqcap \exists hasK.k_p^1$

Figure 4: A labeled focus in $\mathcal{ALC}$.

4.2 Formal Faceted Classifications

Now we generalize the example. Algorithm 2 below provides a way to build an $\mathcal{ALC}$ faceted knowledge base, or faceted ontology, for a given context. The algorithm works in two main steps.

First, it builds a knowledge base by adding $\mathcal{ALC}$ equality axioms that formally define the concept terms of an input context by using focused terms computed by Algorithm 1 over the same context. For maching purposes that we will see in Section 5, if strictly more or strictly less (but at least one) focused terms were computed for a concept term, then the algorithm adds to the knowledge base all the equality axioms defined over all possible combinations of four focused terms picked up, possibly with repetitions, from the computed terms.

Second, the algorithm adds to the knowledge base so obtained all $\mathcal{ALC}$ equality axioms that formally define DEPA facets of every concept as stored in the facet repository (see Section 2). These axioms have the form

$$C \equiv \exists FacetD.d \sqcap \exists FacetE.e \sqcap \exists FacetP.p \sqcap \exists FacetA.a, \quad (4)$$

where C represents a concept c available in the facet repository, $FacetD$, $FacetE$, $FacetP$, and $FacetA$ are named roles rapresenting the property of c in terms of DEPA facet analysis paradigm.[3] The intended interpretation of these named roles relates to the facet repository. For example, $\exists FacetD.f$ means that there is a concept in the facet repository with facet 'Discipline' be f. By extension, equality axiom (4) means that there is a concept in the facet repository with facet 'Discipline' d, 'Entity' e, 'Property' p, and 'Action'

[2] That in our approach clusters in a focus are related to each other by a subsumption relation follows from Convention 2 by observing that polyhierarchical classifications are often subsumption hierarchies. However, we do not need to strictly assume (2) in this paper.

[3] To shorten notation, in algorithms we use D, E, P, A instead.

a, and that concept has name C. Hence, as per second step, Algorithm 2 adds to the knowledge base all axioms of form as in (4) if and only if there is a concept (or a subject) with DEPA facets d, e, p, a in the facet repository. We make the system insensitive to case and punctuation in the facets d, e, p, a by adding additional axioms where variants of d, e, p, a with the same meaning are used. We call the ontology produced by Algorithm 2 a *formal faceted classification* (FFC).

Algorithm 2 Building a $\mathcal{ALC}$ faceted ontology $\mathcal{O}$.

```
Input: $\mathcal{C}, \mathcal{D} \neq \emptyset, FR(\mathcal{C})$
Set $\mathcal{O} = \emptyset$;                           /* $\mathcal{ALC}$ ontology to be built */
foreach $C$ in $\mathcal{C}$ with $C \neq \emptyset$ do
  $l_C := \langle k_1 k_2 \cdots k_n \rangle$;            /* $l_C$ computed by Algorithm 1 */
  for $i = 1$ to $i = \binom{n}{4}$ do
      $\mathcal{O} := \mathcal{O} \cup \{C \equiv \exists hasK.k_{i1} \sqcap \cdots \sqcap \exists hasK.k_{i4}\}$ od;
  if $\langle C : d, e, p, a \rangle \in FR(\mathcal{C})$
                        /* facets $d, e, p, a$ for $C$ in facet repository */
    then
        $\mathcal{O} := \mathcal{O} \cup \{\mathrm{C} \equiv \exists \mathrm{D}.d \sqcap \exists \mathrm{E}.e \sqcap \exists \mathrm{P}.p \sqcap \exists \mathrm{A}.a\}$;
                                         /* axiom of form in (4) added */
  fi od;
Return : $\mathcal{O}$.
```

5. FACET ENGINE

Now we design within our framework a facet engine that computes the matching between the focused terms of a input context and the predefined set of facets stored in the facet repository for a number of concepts. Intuitively, the facet engine looks at all keywords generated for each concept name in a focus for all focuses of the hierarchy, and browse through the focus from the root to the leaf to identify what keywords are DEPA facets stored in facet repository. The facet engine's main component is Algorithm 3. The basic steps of the algorithm are the following:

Step 1. Input a concept description C that represents a user's query; the different possible queries that can be represented this way are presented in Section 6.

Step 2. Find and retrieve from the ontology built by Algorithm 2 all equality axioms that define C in the ontology either by focused terms or DEPA facets. If no axioms do exist, that is, C is not defined according to the knowledge stored in the ontology, the algorithm ends with no help to the user. This state means that the search engine cannot provide the user with help for query refinement by facets.

Step 3. For all retrieved axioms and for each axiom of the form $C \equiv \exists hasK.k_1 \sqcap \cdots \sqcap \exists hasK.k_n$, where $l_C = k_1 ... k_n$ is the label computed by Algorithm 1, the algorithm runs the $\mathcal{ALC}$ SAT solver in order to match (focused) terms k_i in the axiom to all DEPA facets for C possibly stored in the facet repository. Note that the performance of our method mainly dependents on this step, namely, the number and complexity of the matchings. Preliminary results suggested that the algorithm satisfies the requirements of a query refinement system in terms of real time performance. A complete study of the complexity of this step is in progress.

Step 4. For all successful matchings computed in Step 3, the retrieved DEPA facets are output and shown to the user.

Algorithm 3 Query expansion with facets from focused terms.

```
proc QueryExpansion
  Input: $C$, $\mathcal{O}$, $FR(\mathcal{C})$    /* $C$ is meant to represent user query */
  Define $\Omega_K$ be the set of axioms in $\mathcal{O}$ of the form
  $C \equiv \exists hasK.k_1 \sqcap \cdots \sqcap \exists hasK.k_n$;          /* $k_1 ... k_n = l_C$ */
  Define $\Omega_F$ be the set of axioms in $\mathcal{O}$ of the form
  $C \equiv \exists D.d \sqcap \exists E.e \sqcap \exists P.p \sqcap \exists A.a$;
                                    /* $\langle C : d, e, p, a \rangle$ is in $FR(\mathcal{C})$ */
  if $\Omega_K \vee \Omega_F = \emptyset$
    then exit                           /* no query exspansion provided */
     else
        $s := Card(\Omega_K)$;                 /* $\Omega_K$ cardinality is $s \geq 1$ */
        $t := Card(\Omega_F)$;                 /* $\Omega_F$ cardinality is $t \geq 1$ */
        $FacetSet(C) := \emptyset$;     /* set of facets retrieved for $C$ */
        for $j = 1$ to $j = s$ do
            $F_{00} := \emptyset$;        /* different facets strings retrieved */
                                   /* by using a single axiom in $\Omega_K$ */
            for $l = 1$ to $l = t$ do
                for $i = 1$ to $i = \binom{n}{4}$ do
                    if $\mathcal{O} \models \exists hasK.k_{i1} \sqcap \cdots \sqcap \exists hasK.k_{i4}\} \equiv$
                      $\exists D.d \sqcap \exists E.e \sqcap \exists P.p \sqcap \exists A.a$
                      /* focused terms and DEPA facets match */
                      then
                          $F_{li} := F_{li-1} \cup \{\langle C : d, e, p, a \rangle\}$
                                        /* $\langle C : d, e, p, a \rangle$ retrieved */
                                        /* depending on $k_{i1}$,...,$k_{i4}$ */
                    fi od
            od;
            $FacetSet(C)_j := FacetSet(C)_{j-1} \cup F_{li}$
                 /* all DEPA strings for $C$ in $FR(\mathcal{C})$ retrieved */
        od fi;
  Return : $FacetSet(C)_j$.
```

6. QUERY PROCESSING

After building the faceted ontology and defining the facet engine we are ready to use them to provide new facets to the user for query refinement. We allow the user to make three kind of query: keyword-based, by focus, and on subject. We discuss each querying method in turn.

6.1 Keyword-based querying

The user types one or more keywords in the search box. This method is the simplest one and it is often the only method available when the user does not know anything about the subject to search, or the user's knowledge on the query subject is not based on documents locally stored in the user querying machine, so that we can not use the ontology and facet engine we have advanced. This is also a tyipical case of keyword-based querying by common search engines, where the keywords used in the query are listed without a specific ordering on the only basis of the user's information need.

We deal with this method of querying as follows. Each keyword is mapped to zero or more concept terms in the context $\mathcal{C}$. We do that using an exact string match of the keyword to the concept term or one of its alias names, namely, its focused terms.

If no concept term and its alias names match any keyword, no concept description is available to the facet engine, and as a consequence no facets for query refinement are shown to the user.

If one concept term or its alias names match some keywords, then the concept description C of the concept term is generated and processed by Algorithm 3 for query expansion. The facets that occur in the query expansion are shown to the user. When selecting one of the new facets, the user will narrow down the search by expanding the original query with the suggested facet.

If multiple concept terms match some keywords, then the concept description of each term is generated and processed by Algorithm 3 for query expansion. The facets that occur in the query expansion of every concept description are shown to the user. Alternatively, the user is given the option to refine their query to indicate which concept term, namely, keyword they meant the most.

6.2 Querying-By-Focus

Now suppose that the user knows at least something about the subject to search, and the user's knowledge comes from documents stored and polyhierarchically organized in the user's document collection. In this case, it would always be desiderable for the user to get better and better understanding of the hidden content of the query, as it is automatically generated by a suitable method, so as to discover new facets of the original query that the user was not aware of before. For example, suppose the query is 'apple' as contextualized in Figure 5. The user clicks on a concept term in a context $\mathcal{C}$. In doing that, the user selects a focus in $\mathcal{C}$. Alternatively, the user types some keywords as in keyword-based querying, but in a specific order to mean a focus in $\mathcal{C}$. For example, the user may click on (an appropriate graphic-version of) 'Apple' in context or either type keywords 'fruit', 'trentino', 'apple' *in this order*, as to mean $\mathcal{C}x$:Fruit>Trentino>Apple. In the example, by selecting the facet 'Fruit' the user would narrow down the search space by excluding all subjects about Apple Computers and related subjects as search results (see Figure 1). Similarly, by selecting facet 'Trentino' the user would be able to narrow down the search space by excluding all subjects about fruits that are not related to Trentino's production of apples. It follows that the keyword-based method and querying by focus are not equivalent for at least one reason, that is, in keyword-based querying the order of keywords does not matter, in querying by focus does. The other main difference between these two querying methods arises looking at query processing. The difference is that concept terms in a focus are not 'pure' keywords; a concept term is represented by a *string of similar keywords* as generated by Algorithm 1. Concept terms relate to documents in the user's repository, while keywords are usually unrelated to the user's documents.

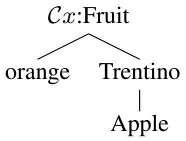


Figure 5: A focus for query 'Apple'.

A query-by-focus is similar to a query by example, yet it is more specific. In querying by example, a sample document (*the example*) is selected by the user to refine the query. On the other hand, in querying by focus the *position* of the sample document is also considered, that is, the place the document is stored within the user's documentary repository. To illustrate, suppose that a user stores his documents according two different structures, see Figure 6. Now suppose the user selects the document named **doc1** as the sample document. In classical querying by example, a relevant answer to the user would be any document about 'apple', as meant as either a fruit or a computer. In contrast, using querying by focus the only relevant answers to the user would be documents from one of the two focus Fruit>Apple and Computers>Apple.

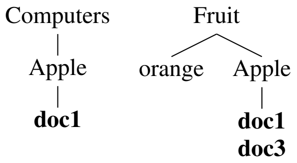


Figure 6: Position of sample document doc1 matters.

We deal with querying by focus as follow. First, a concept description C of the concept term that is the leaf of the focus is generated and processed by Algorithm 3 for query expansion. The facets that occur in the query expansion are shown to the user. When selecting one of the new facets, the user will narrow down the search by expanding the original query with the suggested facet.

Note that the case where query by focus applies in practical situations is not as uncommon as it may seem, because almost all users start a search from a device storing text and text-annotated documents, and these are often organized by the user according to a polyhierarchical classification. More importantly, the fact that a user searches the Web does not mean that documents *from the Web* will be used for the purpose of querying by focus. The documents used for querying by focus are all and only the documents locally stored in the user's querying device, whatever the search objective is either to retrieve documents stored in the user's device or in the Web. As a consequence, querying by focus clearly scales to the size of the web. To understand a bit further, recall that our method is about query refinement, it is not a query search method. We use standard methods and search engines to search; the difference is that the keywords we let the search engines to use are automatically generated by our facetization technique.

6.3 Querying-On-Subject

Subject-based querying is the most common approach by specialized users, where 'subject' refers to the topical intent of a query (cf. Section 2). In our faceted approach to representation of documents in collection $\mathcal{D}$, 'subjects' are broken down into distinct divisions, the facets of subject. A typical 'query-on-subject' is deemed to relate to a specific subject of a preexisting faceted classification. For example, a subject-based query is: 'What are the documents on the effects of nitrogen fertilizers on rice plants?' The subject of the concept subsumed by this query is one of possibly many focuses, for example the following:

$$\mathcal{C}x\text{:rice plants} > \text{nitrogen fertilizers} > \text{effects.} \tag{5}$$

This is a partial focus, in the sense that the discipline subsumed by the query as provided by the DEPA facet analysis is

$$\mathcal{C}x\text{:Agriculture} > \text{rice plants} > \text{nitrogen fertilizers} > \text{effects.} \tag{6}$$

Another possible focus for the subject of query's concept is the following:

$$\mathcal{C}'x\text{:Agriculture} > \text{ effects of nitrogen} > \text{ fertilizers} > \text{rice plants.} \tag{7}$$

A number of different but equivalent focuses could exists for a given subject-based query. Note the the existance of a focus for this query as well as the focus form depend only upon the querying user's classification of documents. The take-away point is that by merging a subject to one or more focuses, by automatically transforming a query-on-subject to a query-by-focus, the method provides the user with assistance in query refinement. In fact, we compute the focuses generated from the query on subject, and for each focus we consider the concept description that represents the focus in $\mathcal{ALC}$ ontology computed by Algorithm 2. Then we proceed as in the case of querying by focus and compute the query expansion of the focus according to knowledge stored in the ontology. Finally, the retrieved facets are shown to the user. If multiple focuses are computed from the query's subject, the user is given the option to refine the original query to indicate which focus they meant for the searched subject.

7. RELATED WORK

There has been extensive work on automated facet construction motivated by query refinement, browsing and navigation over document collections, see for instance [29], [8, 9], [10], .[24], [30, 13]. The notion of context in these related works differ from the notion of focus; in [10] context is a piece of text, from a document the user is presented to, surrounding the query, which is marked by the user on the document. The structural nature of a focus contrasts with the plain, linguistic nature of query context as meant in [10]. The navigation trees discussed in [28] are similar to the focuses discussed in this paper. The formal approach of [28], moreover, as well as the use of faceted taxonomies is close in spirit, if not in the formal development to our work presented here. As far as we know, none of the foregoing approaches uses a DEPA facet schema.

Our method is a focused retrieval method, in the sense that focused retrieval addresses ways to provide a querying user a more direct access to relevant information [26]. Focused retrieval aims to identify not only documents relevant to a user information need, but also where within the document the relevant information is located. Our approach of querying-by-focus is similar to querying by focus on hierarchical classifications proposed by [1, 2].

In the Indian Context, faceted library systems, especially the Colon Classification System (CCS), has been adopted by majority of the academic libraries for organizing collections in semantic arrangement. However, there is a wide scope for use of the faceted theory behind systems such as CCS to other knowledge modeling efforts. Prasad and Guha [18] intoduced a facet-based method to formulate the descriptive domain metadata that could be used to annotate digital library resources. Prasad and Madalli [19] propose a generic model for building semantic infrastructure for digital libraries based on facets as used in traditional library classification systems.

Faceted taxonomies are extensively studied, see for instance [21, 27, 28] and references therein. Facet techniques include that studied by Tvarožek and Bieliková [27], who have proposed faceted navigation and its personalization in digital libraries. They follow a method of faceted browser adaptation based on an automatically acquired user model with support for dynamic facet generation J. Polowinski [17] argues for use of Faceted Browsing as a visual selection mechanism to browse data collections as it is deemed as being particularly suitable for structured, but heterogeneous data with explicit semantics. Normalized Formal Classifications (NFC) used in [11] does this by taking into account both the label of the node and its position using natural language processing techniques (see [11, sec 4]). On the other hand, we have used an information retrieval technique to find out the keywords that will successively represented in concept descriptions by using role names of the form $hasK.k$. This is an important difference with [11]. A focus is called "concept at a node" in [11, p. 70], although we believe that the two notions are not totally equivalent (to be investigated). The notion of Formal Faceted Classification (FFC) extends the notion of "lightweight ontology" of [11] to facets. A main difference with lightweight ontologies by [11] is that FFC's descriptive language is not propositional as the language used in [11]. Yet, it allows us to automate, through DL reasoning services (SAT), query refinement, as we did in this paper. Moreover, by our query language we allow a user to specify a query by selecting a sample document, to be interpreted of as the "information need" of documents similar to the sample selected. As a consequence, we provide a user with a mechanism of "querying by example" as a special case. On the other hand, in [11] it seems not easy to formalize querying by example, as the propositional language used does not allow to represent instances.

8. CONCLUSION

This paper presented a formal framework for a querying refinement method that enables the extraction of the diversity aspects, or facets, of a user query. The method uses the general principles of facet analysis in the DEPA paradigm of facetization and the notion of 'focus', which is used to infer new facets from the user query. The method provides a user with additional and essential contextual information, in form of new facets. When selecting one of the new facets, the user can narrow down the search by expanding the original query with the suggested facets. The proposed method of query refinement is based on diversity in querying and a multi-dimensionality of information. Three methods of querying weree discussed: keyword-based, by focus, and on subject. For each method, textual and structural dimensions were used to assist the user in query refining. The textual dimension allowed us to generate the top-k most relevant terms for each concept of a given polyhierarchy of text and text-annotated documents. The structural dimension of the polyhierarchy was used to match DEPA facets with the user query. We have situated our framework within the smallest propositionally closed description logic $\mathcal{ALC}$, and we have used $\mathcal{ALC}$'s solver to implement the facet engine as the main component of the method.

9. REFERENCES

[1] A. Agostini and P. Avesani. On the discovery of the semantic context of queries by game-playing. In H. Christiansen, M.-S. Hacid, T. Andreasen, and H. Larsen, editors, *Proceedings of the Sixth International Conference On Flexible Query Answering Systems (FQAS-04)*, pages 203–216, Berlin Heidelberg, 2004. Springer-Verlag LNAI 3055.

[2] A. Agostini and G. Moro. Identification of communities of peers by trust and reputation. In D. F. C. Bussler, editor, *Proceedings of the Eleventh International Conference on Artificial Intelligence: Methodology, Systems, Applications - Semantic Web Challenges (AIMSA-04)*, pages 85–95, Berlin Heidelberg, 2004. Springer-Verlag LNAI 3192.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data*

Mining (WSDM-00), pages 5–14, New York, NY, 2009. ACM Press.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *Handbook of Description Logics*, Cambridge, UK, 2002. Cambridge University Press.

[5] F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. M. Guinness, and P. P.-S. D. Nardi, editors, *Handbook of Description Logics*, pages 47–100. Cambridge University Press, Cambridge, UK, 2002.

[6] G. Bhattacharyya. POPSI: its fundamentals and procedure based on a general theory of subject indexing languages. *Library Science with a Slant to Documentation*, 16(1):1–34, 1976.

[7] G. Bhattacharyya. Subject indexing language: its theory and practice. In *Proceedings of the DRTC Refresher Seminar–13, New Developments in LIS in India*, Bangalore, India, 1981. DRTC, ISI Bangalore Centre.

[8] W. Dakka, R. Dayal, and P. Ipeirotis. Automatic discovery of useful facet terms. In *Proceedings of the ACM SIGIR 2006 Workshop on Faceted Search*, New York, NY, 2006. ACM Press.

[9] W. Dakka and P. Ipeirotis. Automatic extraction of useful facet hierarchies from text databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE-08)*, pages 466–475, Washington, DC, USA, 2008. IEEE Computer Society.

[10] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revised. In *Proceedings of the Tenth International World Wide Web Conference (WWW-2001)*, pages 406–414, New York, NY, 2001. ACM Press.

[11] F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Encoding classifications into lightweight ontologies. In S. Spaccapietra and et. al., editors, *Journal on Data Semantics VIII*, pages 57–81. Springer-Verlag LNCS 4380, Berlin Heidelberg, 2007.

[12] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227, 2000.

[13] K. Latha, K. R. Veni, and R. Rajaram. AFGF: An automatic facet generation framework for document retrieval. In *Proceedings of the 2010 International Conference on Advances in Computer Engineering (ACE-2010)*, pages 110–114, Washington, DC, USA, 2010. IEEE Computer Society.

[14] S. Lawrence. Context in Web Search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.

[15] V. Maltese, F. Giunchiglia, K. Denecke, P. Lewis, C. Wallner, A. Baldry, and D. Madalli. On the interdisciplinary foundations of diversity. In G. Boato and C. Niederee, editors, *Proceedings of the First International Workshop on Living Web at ISWC-09, Washington D.C., USA, October 26, 2009*. CEUR-WS, 2009.

[16] P. Morville and L. Rosenfeld. *Information architecture for the World Wide Web, 3rd edition*. O'Reilly Media, Inc., Sebastopol, CAe, 2006.

[17] J. Polowinski. Human interface and the management of information. Designing information environments. In M. J. Smith and G. Salvendy, editors, *Proceedings of the Symposium on Human Interface 2009, held as Part of HCI International 2009 (HCII-09), San Diego, CA, USA, July 19-24, 2009*, pages 601–610, Berlin Heidelberg, 2009. Springer-Verlag LNCS 5617.

[18] A. Prasad and N. Guha. Expressing faceted subject indexing in SKOS/RDF. In *Proceedings of the First International Conference of Semantic Web and Digital Libraries, Bangalore 21-23 February (ICSWDL-07)*, 2007.

[19] A. Prasad and D. Madalli. Semantic digital faceted infrastructure for semantic digital libraries. *Library Review*, 57(3):225–234, 2008.

[20] S. R. Ranganathan. *Prolegomena to Library Classification*. Asia Publishing House, London, 1967.

[21] G. Sacco and Y. Tzitzikas, editors. *Dynamic Taxonomies and Faceted Search*, The Information Retrieval Series, v. 25, Berlin Heidelberg, 2009. Springer-Verlag.

[22] G. Salton, editor. *The SMART Retrieval System—Experiments in Automatic Document Retrieval*, Englewood Cliffs, NJ, 1971. Prentice-Hall Inc.

[23] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.

[24] E. Stoica, M. A. Hearst, and M. Richardson. Automating creation of hierarchical faceted metadata structures. In *Proceedings of the Human Language Technology Conference (NAACL HLT)*, pages 244–251, Rochester, NY, USA, 2007. Association for Computational Linguistics.

[25] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Using keyword extraction for web site clustering. In K. Wong, editor, *Proceedings of the Fifth International Workshop on Web Site Evolution (WSE-03)*, pages 41–48, Amsterdam, The Netherlands, 2003. IEEE Computer Society.

[26] A. Trotman, S. Geva, J. Kamps, M. Lalmas, and V. Murdock. Current research in focused retrieval and result aggregation. *Special Issue in the Journal of Information Retrieval*, 13(5):407–411, 2010.

[27] M. Tvarôzek and M. Bieliková. Personalized faceted browsing for digital libraries. In L.ács, N. Fuhr, and C. Meghini, editors, *Research and Advanced Technology for Digital Libraries. Proceedings of the 11th European Conference on Digital Libraries (ECDL-07), Budapest, Hungary, September 16-21, 2007*, pages 485–488, Berlin Heidelberg, 2007. Springer-Verlag LNCS 4675.

[28] Y. Tzitzikas, N. Spyratos, P. Constantopoulos, and A. Analyti. Extended faceted taxonomies for web catalogs. In *Proceedings of the Third International Conference on Web Information Systems Engineering (WISE-02)*, pages 192–204, 2002.

[29] R. van Zwol and B. Sigurbjörnsson. Faceted exploration of image search results. In *Proceedings of the Nineteenth International World Wide Web Conference (WWW-10)*, pages 961–970, New York, NY, 2010. ACM Press.

[30] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE-08)*, pages 228–236, Washington, DC, USA, 2008. IEEE Computer Society.

[31] B. Vickery. *Faceted classification: A guide to construction and use of special schemes*. Aslib - Asia Publishing House, London, 1960.

[32] K. Weinberger, M. Slaney, and R. van Zwol. Resolving tag ambiguity. In *Proceedings of the 16th International ACM Conference on Multimedia (MM 2008)*, New York, NY, 2010. ACM Press.

Approximate subgraph matching for detection of topic variations

Mitja Trampuš
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
mitja.trampus@ijs.si

Dunja Mladenić
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

The paper presents an approach to detection of topic variations based on approximate graph matching. Text items are represented as semantic graphs and approximately matched based on a taxonomy of node and edge labels. Best-matching subgraphs are used as a template against which to align and compare the articles. The proposed approach is applied on news stories using WordNet as the predefined taxonomy. Illustrative experiments on real-world data show that the approach is promising.

Categories and Subject Descriptors

I.2.8 [**Artificial intelligence**]: Search—*Graph and tree search strategies*; I.5.4 [**Computing methodologies**]: Applications—*Text processing*

1. INTRODUCTION

One of the classic goals of text mining is to structure natural language text – for obvious reasons: the amount of information we can extract from the data using shallow approaches like bag-of-words is limited. By enhancing text with structure, we can start to observe information that is encoded in more than one word or sentence. Also, structure enables us to bring the additional power of semantic methods and background knowledge into the play.

While reasonably reliable methods have been developed for structuring text by annotating and identifying a specific subset of information, mostly named entities, little work has been done on semantically capturing the macro-level aspects of the text. In this article, we present some early work on constructing *domain templates*, a generic "summary" that fits many pieces of text on a specific theme (e.g. news stories about bombings) at the same time.

The genericness of the template provides for data exploration in two ways:

1. By automatically mapping specific facts and entities in an article to the more general ones in a template, we are providing structure to the articles as entities from potentially many articles get mapped to a single semantic "slot" in a single template.
2. By (possibly statistically) inspecting all the articles that were mapped to a chosen template, we can observe the diversity of articles *in a specific aspect*, exploiting the fact that they are semantically aligned to an extent. For example, if the template contains the statement `happen_at..location`, no further processing is required to find the specific locations which the template-mapped articles describe.

To determine entities and relations that subsume those from individual news articles and thus construct a template, we make use of a general-purpose ontology, in our case WordNet. To represent the templates as well as individual news stories, we use *semantic graphs*, i.e. graphs in which entities represented as nodes and the (binary) relationships connecting them are represented as edges.

A sample of the patterns we obtain can be seen in Figure 2. For example, analyzing a collection of articles describing various bombing attacks, the pattern in the first line emerges: a *person* was killed on a *weekday*; that same *person* was killed in an attack which took place. The concrete instantiations of *person* and *weekday* vary across articles from which the pattern was derived.

Domain specifics.

Note that media companies have a considerable interest in semantically annotating text, particularly news items. For this reason, and because of easy availability of datasets, we focus on the domain of newswire in this paper. Despite this, there is in principle nothing specific in this domain that would limit the applicability of our method to it. In general, the required input data is a collection of text items which are assumed to discuss roughly the same aspects of a single topic. Examples of such collections are "news articles about earthquakes", "Wikipedia articles on football players" or "microwave product reviews".

2. RELATED WORK

Because it aligns articles to a common template, our method has much in common with other information extraction mechanisms. Automatic construction of information extraction templates is already relatively well-researched. Most methods aim for *attribute extraction*, where the goal is to extract a single predefined type of information, e.g. the title of a book. Each separate type of information requires a separate

DiversiWeb 2011. Hyderabad, India.
.

classifier and training data. Examples of such approaches are [1, 2].

More recently, a generalized problem of *relation extraction* has received considerable attention. The goal is to find *pairs* of items related by a predefined relation. As an example, Probst et al. [7] mine product descriptions to simultaneously identify product attributes and their values. Relation extraction is particularly popular in biomedicine where pairs of proteins in a certain relation (e.g. one inhibits the other) are often of interest.

The task in this article is more generalized still; we attempt to decide both what information is interesting to extract as well as perform the extraction. This is known as *domain template extraction.* To our knowledge, little work has been done in the area so far. The most closely related work is by Filatova et al. [4], who find templates by mining frequent parse subtrees. Also closely related is work by Li et al. [6]; similarly to Filatova, they mine frequent parse subtrees but then cluster them into "aspects" with a novel graphical model. Both approaches produce syntax-level patterns. Unlike ours, neither of the two approaches exploits background knowledge. Also belonging to this group is our previous work [10] which mostly shares the goal and data representation ideas with this article, but uses different methods apart from preprocessing.

Graph-based templates are also used in [9] in a context similar to ours, though the semantics are shallower. Also, the authors focus on information extraction and do not attempt to generalize the templates.

Templates somewhat similar to those we aim to construct automatically and with no knowledge of the domain have already been created manually by domain experts. FrameNet [?] is a collection of templates for the events like "disclosing a secret", "speaking", "killing", "arresting" etc. They focus mostly on low-level events, of which typically many can be found in a single document, be it a news article or not. The project does not concern itself with the creation of the templates, other than from the methodological point of view. There is little support for automatic annotation of natural language with the FrameNet frames.

3. METHOD OVERVIEW

This section describes the various stages in our data processing pipeline. The assumed input data is, as discussed above, a collection of text items on the same topic. The goal is to identify a pattern which semantically matches a substantial number of the input texts.

The key idea is rather simple: we first represent our input data as semantic graphs, i.e. graphs of ontology-aligned entities and relations. A pattern is then defined as a (smaller) graph such that, by specializing some of its entities, a subgraph of at least θ input graphs (θ being a parameter). We seek to identify all such patterns.

We proceed to describe our approach to the construction of semantic graphs and to the mining of approximate subgraphs.

3.1 Data Preprocessing

Starting with plain text, we first annotate it with some basic semantic and linguistic information. Using the ANNIE tool from the GATE framework, we first detect named entities and tag them as person, location or organization. Following that, we use the Stanford parser [5] to extract subject-verb-object triplets. We then use the web service by Rusu [8] to perform coreference and pronoun resolutions ("Mr. Obama", "President Barack Obama" and "he" might all refer to the same entity within an article).

We acknowledge that the triplets acquired in this way do not necessarily provide a proper semantic description of the article data. The discrepancies go both ways:

- We include some triplets which do not make sense semantically, e.g. "`people..kill..Monday`" coming from "93 people were killed on Monday".

- We fail to create triplets for information not encoded with (lexicogrammatically) transitive verbs. For example, "President's visit to China ..." will not spawn "`president..visit..China`". In our experiments, this shortcoming is alleviated by using redundant information - each story, e.g. president's visit to China, is described by several articles which increases the probability that at least one will convey this information in a form we can detect. However, the problem is not completely overcome this way – some information e.g. the "93" in "93 people were killed on Monday" will never appear as the object of a transitive verb.

As a last step, we align all triplets to WordNet; that is, for each subject, verb and object appearing in any of the triplets, we try to find the corresponding concept (or "synset", as they are called) in WordNet. We first remove inflection from the words using python NLTK (Natural Language Toolkit), then align it to the corresponding synset. If more than one synset matches, we choose the most common sense which is a well-tried and surprisingly good strategy. For words not found in WordNet, we create a new synset on the fly. If the new word (e.g. "Obama") was previously tagged by ANNIE (with e.g. "person"), the new synset's hypernym is set accordingly.

3.2 Semantic Graph Construction

From a collection of triplets, we proceed to construct the semantic graph. Here, we rely rather heavily on the fact that news articles tend to be focused in scope: we do not disambiguate entities other than by name (not necessarily a proper name; e.g. "book" is also a name). As an example, if an article mentions two buildings, one of which burns down and the second of which has a green roof, our method detects a single "building" and assigns both properties to it. In the newswire domain, we have not found this to be a significant issue: entities which do need to be disambiguated are presented with more unique names ("France" instead of "country" etc.). This rationale would have to be revised if one wanted to apply the approach to longer texts.

This assumption greatly simplifies the construction of the semantic graph: we start by treating each triplet as a 2-node component of a single very fragmented graph and then collapse the nodes with the same labels.

Dataset specifics.

In our experiments, each input "document" in the sense described here was in fact a concatenation of actual documents, all of which were reporting on the exact same news event. Section 4 contains the details and rationale.

3.3 Approximate Pattern Detection

Given a collection of labeled graphs, we now wish to identify frequent "approximate subgraphs", i.e. patterns as described at the beginning of Section 3.

Formal task definition: Given a set of labeled graphs $S = \{G_1, \ldots, G_n\}$, a transitive antisymmetric relation on graph labels $genl(\cdot, \cdot)$ (with $genl(A, B)$ interpreted as "label A is a generalization of label B") and a number θ, we wish to construct all maximal graphs H that are *approximate subgraphs* of at least θ graphs from S. A graph H is said to be an approximate subgraph of G iff there is a mapping f of $V(H)$ onto a subset of $V(G)$ such that $genl(v, f(v))$ holds for all $v \in V(H)$.

This is not an easy task. Mining frequent subgraphs is in itself computationally demanding because of isomorphisms; satisfactorily fast algorithms for this seemingly basic problem are relatively recent [11]. By further requiring that the frequent subgraph only match the input graphs in a *soft* way implied by a taxonomy (here WordNet hyperymy), the complexity becomes insurmountable. We compensate by introducing two assumptions.

1. The hierarchy imposed by *genl* has a tree-like form, it is not a general DAG. This is true of WordNet: every synset has at most one hypernym defined.

2. Very generic patterns are not interesting and can (or even should) be skipped. This too is a safe assumption in our scenario: a pattern in which every node is labeled with the most generic label `entity` is hardly informative regardless of its graph structure.

We can now employ a simple but effective three-stage search. The stages are illustrated in 1 with the minimal example of two two-node graphs.

1. Generalize all the labels of input graphs to the maximum extent permissible. Under the first assumption, "generalizing a label" is a well-defined operation. The exact meaning of "maximum extent permissible" is governed by the second assumption; no label should be generalized so much as to fall in the uninteresting category. In our experience with WordNet, the following simple rule worked very well: generalize verbs as much as possible and generalize nouns to two levels below the hierarchy root. See steps 1 to 2 in Fig. 1.

2. Mine θ-frequent maximal subgraphs with support of the generalized input graphs. This step cannot be shown in Fig. 1 as the graphs are too small.

3. Formally, the resulting subgraphs already satisfy our demands. However, to make them as descriptive as possible, we try to specialize the pattern's labels, taking care not to perform a specialization that would reduce the pattern's support below θ. Specialization, unlike generalization, is not a uniquely defined operation (a synset can have many hyponyms), but with some we can afford to recursively explore the whole space of possible specializations. We use the sum of labels' depth in the WordNet hierarchy as a measure of pattern descriptiveness that we optimize. See steps 2 to 3 in Fig. 1.

For frequent subgraph mining, we developed our own algorithm, inspired by the current state-of-art[11, 3]. We included some improvements pertaining to maximality of output graphs and to scalability – all existing open-source software crashed on our full input data.

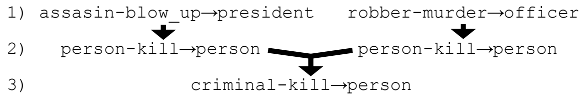


Figure 1: Generalization of input graphs and re-specialization of the pattern.

4. PRELIMINARY EXPERIMENTS AND RESULTS

As a preliminary, let us define some terminology suitable for our experiment domain. An *article* is a single web page which is assumed to report on a single *story*. A story is an event that is covered by one or more articles. Each story may fit some *domain template* (also *event template* or simply *template*) describing a certain type of event.

We obtained a month's worth of articles from Google News by crawling. Each article was cleaned of all HTML markup, advertisements, navigation and similar. Articles were grouped into stories according to Google News.

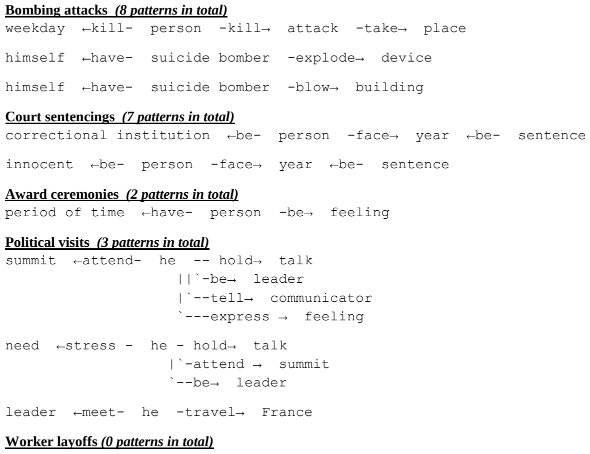
For each *story*, a semantic graph was constructed. The reason to use an aggregate story graph rather than individual article graphs was twofold. First, by representing each story as a single graph, all stories were represented equivalently (as opposed to the case where each article contributed a graph, resulting in stories being weighted proportionally to the number of their articles). Second, the method for extracting triplets has relatively low precision and recall; it therefore makes sense to employ the redundancy inherent in the collection of articles reporting on the same event. To construct the aggregate story graph, we simply concatenated the plain text of individual articles; aggregation at this early stage has the added benefit of providing cross-article entity resolution. Finally, the collection of semantic graphs from stories on a single topic was input to the pattern mining algorithm.

We defined five topics on which to observe the behavior of the method: bomb attacks, award ceremonies, worker layoffs, political visits and court sentencings. For each, we identified about 10 stories of interest. Note that each story further comprises about 100 articles, clustering courtesy of Google News; in total, about 5000 articles were therefore processed.

As semantic graphs were constructed on the level of stories rather than articles, their structure was relatively rich. They had about 1000 nodes each and an average node degree of roughly 2.5. The 20% most connected nodes, which are also the ones likely to appear in the patterns, had an average degree of about 20.

For each topic, graphs of all its stories were input to the algorithm. The minimal pattern support was set at 30% for all the topics. The algorithm output several patterns for each topic; the sizes of the outputs along with the interesting patterns are presented in Figure 2.

For instance, the last person in the "visits" domain shows that in at least 30% of the stories, there was a male person ("he", e.g. Obama) who traveled to France (a coincidence), and that same person met a "leader" (a president in some of the stories, a minister in other).



Bombing attacks (8 patterns in total)

```
weekday  ←kill-  person  -kill→  attack  -take→  place

himself  ←have-  suicide bomber  -explode→  device

himself  ←have-  suicide bomber  -blow→  building
```

Court sentencings (7 patterns in total)

```
correctional institution  ←be-  person  -face→  year  ←be-  sentence

innocent  ←be-  person  -face→  year  ←be-  sentence
```

Award ceremonies (2 patterns in total)

```
period of time  ←have-  person  -be→  feeling
```

Political visits (3 patterns in total)

```
summit  ←attend-  he  -- hold→  talk
                   ||`-be→  leader
                   |`--tell→  communicator
                   `---express →  feeling

need  ←stress -  he - hold→  talk
                  |`-attend →  summit
                  `--be→  leader

leader  ←meet-  he  -travel→  France
```

Worker layoffs (0 patterns in total)

Figure 2: Manually selected best patterns for each domain.

5. DISCUSSION AND FUTURE WORK

The preliminary results seem sound. The mappings of individual stories onto the patterns (not given here) also provide a semantically correct alignment. We can observe how each story fits the template with slightly different entities. Sometimes, the variations are almost imperceptible – "correctional facility" from the "court" domain, for example, appears as either "jail" or "prison", which for some reason are two distinct concepts in WordNet.

In other cases, the distinctions are significant and express the subtopical diversity we were looking for. For example, the groundings for "leader" in the "visits" domain varied even in our small dataset over president, minister, instigator or simply leader. In the same domain, "feeling" was either sorrow, disappointment or satisfaction. The "building" in the "bombings" domain was generalized from mosque, restaurant, hotel and building. It might be interesting to investigate this further and use the amount of variation between pattern groundings as a measure of pattern interestingness.

Unexpectedly, diversity can occasionally be found in the natural clustering that the patterns provide. Observe the two patterns in the "court" domain: in both, the defendant is facing a sentence of (one or more) years, but is found innocent in one cluster and sent to(?) the jail in the other.

While the current experiments are too small to draw any conclusive evidence, we can make some speculations about precision and recall. While the first is low but usable (a data analyst should not mind going through e.g. 5 patterns to identify a useful one), the latter seems a bigger issue. We hope to improve the results significantly by developing a better triplet extractor[1]; the previously discussed deficiencies of current triplets appear to hit performance most.

The tests also indicate that the method is not equally suitable for all domains. The "layoffs" domain, for example, had no single pattern which would occur in 30% of the stories. (A threshold of 25% produces a single rather nonsensical pattern "it—cut⟶job⟵lose—people"). The "awards;; domain does not fare much better. Most probably, these two topics are too broad, causing stories to have only little overlap.

Note that in current implementation, all final patterns with less than three nodes (e.g. `worker..lose..job` for the "layoffs" topic) were discarded. Partly this is because we are, in perspective, interested in (dis)proving that structured patterns can provide more information than sentence-level patterns found in related work[2]. Partly, however, it is also because including two-node patterns would introduce additional noise in the output. Even now, the precision is relatively low; it would therefore be interesting to investigate measures of interestingness of patterns other than raw frequency.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under PASCAL2 (IST-NoE-216886), ACTIVE (IST-2008-215040) and RENDER (FP7-257790).

7. REFERENCES

[1] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. pages 337–348, 2003.

[2] S. Brin. Extracting patterns and relations from the world wide web. *Lecture Notes in Computer Science*, pages 172–183, 1999.

[3] Y. Chi, S. Nijssen, R. Muntz, and J. Kok. Frequent subtree mining-an overview. *Fundamenta Informaticae*, 66(1):161–198, 2005.

[4] E. Filatova, V. Hatzivassiloglou, and K. McKeown. Automatic creation of domain templates. In *Proceedings of COLING/ACL 2006*, pages 207–214, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[5] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430. Association for Computational Linguistics, 2003.

[6] P. Li, J. Jiang, and Y. Wang. Generating templates of entity summaries with an entity-aspect model and pattern mining. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 640–649, 2010.

[7] K. Probst, R. Ghani, M. Krema, A. Fano, and Y. Liu. Semi-supervised learning of attribute-value pairs from product descriptions. pages 2838–2843, 2007.

[8] D. Rusu, B. Fortuna, M. Grobelnik, and D. Mladenić. Semantic Graphs Derived From Triplets With Application In Document Summarization. *Informatica Journal*, 2009.

[9] H. Tanev and B. Magnini. Weakly supervised approaches for ontology population. 2006.

[10] M. Trampuš and D. Mladenić. Learning event templated from news articles. In *Proceedings of SiKDD09*, 2009.

[11] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. page 721, 2002.

[1] But this is a new project in itself.

[2] The "visits" domain is a nice indication that this may be true.

Mining Diverse *Views* from Related Articles

Ravali Pochampally
Center for Data Engineering
IIIT Hyderabad
Hyderabad, India
ravali@research.iiit.ac.in

Kamalakar Karlapalem
Center for Data Engineering
IIIT Hyderabad
Hyderabad, India
kamal@iiit.ac.in

ABSTRACT

The world wide web allows for diverse articles to be available on a news event, product or any topic. It is not impossible to find a few hundred articles that discuss a specific topic thus making it difficult for a user to quickly process the information. Summarization condenses huge volume of information related to a topic but does not provide a delineation of the issues pertaining to it. We want to extract the diverse issues pertaining to a topic by mining views from a collection of articles related to it. A view is a set of sentences, related in content, that address an issue relevant to a topic. We present a framework for extraction and ranking of views and have conducted experiments to evaluate the framework.

Categories and Subject Descriptors

H.5 [**Information Systems**]: Information Interfaces and Presentation

General Terms

Human Factors, Experimentation

Keywords

text mining, views, diversity, information retrieval

1. INTRODUCTION

The world wide web is a storehouse of information. Users who want to comprehend the content of a particular topic (e.g. FIFA 2010) are often overwhelmed by the volume of text available on the web. Websites which organize information based on content (google news[1]) and/or user ratings (amazon[2], imdb[3]) also output several pages of text in response to a query. It is difficult for an end-user to process all the text presented.

Multi-Document Summarization [2] is a prominent Information Retrieval (IR) technique to deal with this problem of *information overload*. But summaries typically lack the semantic grouping to present the multiple views addressed by a group of articles. Providing diverse views and allowing users to browse through them will faciliate the goal of information exploration by providing the user a definite and detailed snapshot of their topic of interest. Articles which pertain to a common topic (e.g swine-flu in India) are termed as 'related'. By isolating views we aim to organize content in a detailed manner than that of summarization. We define a *view* as

A sentence or a set of sentences which broadly relate to an issue addressed by a collection of related articles and aid in elaborating the different aspects of that issue

1.1 Motivating Example

Here is a pair of views obtained by our framework. Both the views are mined from Dataset 1. The number in the curly brackets indicates the ID of the article from which the sentence is extracted. Description of datasets is given in Table 1.

Example Views

1. The irresponsibility of the financial elite and US administrations has led the US economy to the brink of collapse. {18} On Friday, the Dow was down a mere 0.3% on the week - but to get there, the Fed and the Treasury had to pump hundreds of billions into the global financial system. {14} The collapse of the oldest investment bank in the country could strongly undermine the whole US financial system and increase the credit crisis. {3} After a week that saw the collapse of Lehman Brothers, the bailout of the insurer AIG and the fire sale of Merrill Lynch and the British bank HBOS, policy makers hit back, orchestrating a huge plan to sustain credit markets and banning short sales of stock. {48} It was a dramatic reversal from the first half of the week, when credit markets virtually seized up and stocks around the globe plunged amid mounting fears for the health of the financial system. {18}

2. The Swiss National Bank is to pump USD 27 billion into markets and the Bank of Japan (BOJ) valued its part in the currency swap with the Federal Reserve at 60 billion. {35} The Bank of Canada was also involved, and The Bank of England said it would flood 40 billion into the markets. {26} And, despite the agreements that Barclays Capital and Bank of America will sign with executives at Lehman Brothers or Merrill Lynch, it is the hunting season in the banking world for the crème de la crème. {14}

[1] http://news.google.com/
[2] http://www.amazon.com/
[3] http://www.imdb.com/

The first view details the breakdown of the US economy along with a few signs of damage control. The second view reports the actions of various banks during the financial turmoil in 2008. These views capture a glimpse of the specific issues pertaining to the topic of 'financial meltdown'. A list of such diverse views would organize the content of a collection of related articles and provide a perspective into that collection.

The problem statement is

> *Given a corpus of related articles A, identify the set V of views pertaining to A, rank V and detect the most relevant view (MRV) along with the set of outlier views (OV)*

1.2 Related Work

Allison et. al [1] [8] proposed that providing multiple viewpoints of a document collection and allowing to move among these view-points will facilitate the location of useful documents. Representations, processes and frameworks required for developing multiple view-points were put forth.

Tombros et al. [10] proposed the clustering of Top-Ranking Sentences (TRS) for efficient information access. Clustering and summarization were combined in a novel way to generate a personalized information space. Clusters of TRS were generated by a hierarchical clustering algorithm using the group-average-link method. It was argued that TRS clustering presents better information access than routine document clustering.

TextTiling [5] is a technique for subdividing text into multi-paragraph units that represent passages or subtopics. It makes use of patterns of lexical co-occurence and distribution. The algorithm has three parts: tokenization into sentence-sized units, determination of a score for each unit and detection of sub-topic boundaries. Sub-topic boundaries are assumed to occur at the largest valleys in the graph that result from plotting sentence-units against scores.

1.2.1 Views vs. Summary

Summary and views generated for Dataset 5 are here - (https://sites.google.com/site/diverseviews/comparison) The summary is generated by update summarization 'baseline algorithm' [6]. It is conspicuous by the lack of organization. Though successful in covering the salient features of the review dataset, it groups several conflicting sentences together (observe the last two sentences of the summary). The views generated by our framework present an organized representation by generating clusters of semantically related sentences. As is evident, the first view is discussing the positive attributes of hotel taj krishna in hyderabad while the second view is negative in tone. The third and fourth views discuss specific aspects of the hotel such as the food and the facilities available. Presenting multiple views for a topic allows us to model the diversity in its content. Our representation is concise as the average number of sentences per view was found to be 3.9. In our framework, we address two drawbacks of summarization - lack of organization and verbosity (due to user-specified parameters).

ID	Source	Search Term	# Articles
1	google news	financial meltdown	49
2	google news	swine flu india	100
3	google news	israel attacks gaza	24
4	amazon.com	the lost symbol	25
5	tripadvisor.com	hotel taj krishna	20
6	tripadvisor.com	hotel marriott	16
7	google news	fifa vuvuzela	39
8	google news	gulf oil spill	26

Table 1: Datasets

1.3 Contributions

The main contributions of this work are

1. Defining the concept of a *view* over a corpus of related articles

2. Presenting a framework for mining diverse views

3. Ranking the views based on a quality parameter (*cohesion*) defined by us and

4. Presenting results to validate the framework

1.4 Organization

In section 2, we elaborate on the framework for the extraction of views. MRV, OV and the ranking mechanism are explained in detail in section 2.5. Section 3 is for experimental evaluation and discussion. In section 4, we sum up our contributions and outline the future work.

2. EXTRACTION OF VIEWS

In this section, we detail the steps involved in the extraction of views and define a quality parameter for ranking the views according to their relevance. Figure 1 presents an overview of the framework by depicting the steps involved in the algorithm. Input and output are specified for each step of the algorithm.

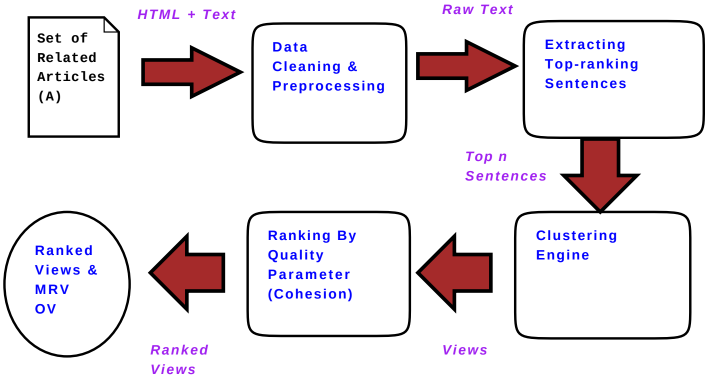

Set of Related Articles (A)
HTML + Text
Data Cleaning & Preprocessing
Raw Text
Extracting Top-ranking Sentences
Top n Sentences
Clustering Engine
Views
Ranking By Quality Parameter (Cohesion)
Ranked Views
Ranked Views & MRV OV

Figure 1: Framework

2.1 Datasets

Articles which make relevant points about a common topic but score low on pairwise cosine similarity can be included in our datasets because we aim to present multiple views from a set of related articles, rather than group them based on overall content similarity. We used data from news aggregator and review web sites as they group articles discussing a common topic, inspite of the low semantic similarity between them. We crawled articles published between a range of dates when the activity pertaining to a relevant topic peaked. For example, we crawled articles published on 'gulf oil spill' between 15 April 2010 and 15 July 2010 when the news activity pertaining to that topic was maximum. We crawled websites which provided rss feeds or had a static html format that could be parsed. Table 1 provides the description of datasets. For instance, Dataset 1 is collected from google news using the search term 'financial meltdown' and contains 49 articles. Datasets can be found here - (https://sites.google.com/site/diverseviews/datasets)

2.2 Data Cleaning and Preprocessing

Web data was collected using Jobo[4], a java crawler. The data was given as an input to the data cleaning and preprocessing stage. Data Cleaning is important as it parses the html data and removes duplicates from the articles. We define a 'duplicate' as an article having the exact syntactic terms and sequences, with or without the formatting differences. Hence, by our definition, duplicates have a cosine similarity value of one.

Text data devoid of html tags is given as an input to the data preprocessing stage. Stemming and stopword removal are performed in the preprocessing stage. Stemming is the process of reducing inflected (or derived) words to their stem or root form. (example: running to run, parks to park etc.) In most cases, these morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. Stopwords are the highly frequent words in english language (example: a, an, the, etc.). Owing to their high frequency, and usage as conjunctions and prepositions, they do not add any significant meaning to the content. Hence, their removal is essential to remove superfluous content and retain the essence of the article. In order to capture the user notion, the review datasets were not checked for typographical and grammatical errors and were retained verbatim. Python modules HTMLParser[5] and nltk.wordnet[6] were used to parse the html data and perform stemming respectively. IR metrics such as word frequency and TF-IDF[7] were extracted for future analysis.

2.3 Extraction of Top-Ranking Sentences

A dataset consisting of many articles and having content spanning various issues needs an pruning mechanism to extract sentences from which the views can be generated. We prune a dataset by scoring each sentence in it and extracting the top-ranked ones. A list of notations used in our discussion is given in Table 2

$T_{i,j}$	:	$tf - idf_{i,j}$
$tf - idf_{i,j}$	:	TF-IDF of term t_i in article d_j
$tf - idf_{i,j}$	:	$tf_{i,j} * idf_i$
$tf_{i,j}$	:	$\frac{n_{i,j}}{\sum_k n_{k,j}}$
$n_{i,j}$	:	Number of occurences of t_i in article d_j
$\sum_k n_{k,j}$	:	$\sum$ of occurences $\forall$ t_k in article d_j
idf_i	:	$\log \frac{\|D\|}{\|d : t_i \in d\|}$
$\|D\|$	:	Total number of articles in the corpus
$\|d : t_i \in d\|$	:	Number of articles which have the term t_i

Table 2: Notations

Let $< S_1, S_2, S_3 ... S_n >$ be the set of sentences in an article collection. $tf - idf_{i,j}$ (TF-IDF) of a term t_i in article d_j is obtained by multiplying its weighted term frequency $t_{i,j}$ and inverse document frequency idf_i. A high value of $tf - idf_{i,j}$ ($T_{i,j}$) is attained by a term t_i which has a high frequency in a given article d_j and low occurence rate among the spectrum of articles present in that collection. Appearance of some words in an article is more indicative of the issues addressed by it than others. $T_{i,j}$ is a re-weighting of word importance, though it increases proportionally by the number of times a word appears in an article, it is offset by the frequency of the word in the corpus. We consider a product of the $T_{i,j}$ of constituent words in a sentence to be a good indicator of its significance. A product can be biased by the number of words in a sentence hence, we normalize the product by dividing it with the length of the sentence. Given the notation above, we thus define the importance I_k, of a sentence S_k, belonging to an article d_j and having r constituent words as

$$I_k = \frac{\prod_{i=1}^{r} T_{i,j}}{r}$$

$T_{i,j} = tf - idf$ of word $w_i \in S_k \wedge d_j$
$I_k =$ product of $tf - idf$ of $\forall w_i$ normalized according to sentence (S_k) length, r

Logarithm normalization was not used as the σ value for r was 2.2 and variance in its value was not exponential. Sentences are arranged in the non-increasing order of their *importance* (I) scores. We choose the top n sentences for our analysis. Experiments are conducted to correlate the range of n with the corresponding score obtained by our ranking parameter.

2.4 Mining Diverse Views

A measure of similarity between two sentences is required to extract semantically related views from them. Semantic similarity calculates the correlation score between sentences based on the likeness of their meaning. Mihalcea et al. [7] proposed that the *specificity* of a word can be determined using its inverse document frequency (idf). Using a metric for word to word similarity and specificity, the semantic similarity of two text sentences S_i and S_j , where w represents a word in a sentence, is defined by them as

[4] http://java-source.net/open-source/crawlers/jobo
[5] http://docs.python.org/library/htmlparser.html
[6] http://www.opendocs.net/nltk/0.9.5/api/nltk.wordnet-module.html
[7] http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

$$sim(S_i, S_j) = \frac{1}{2}\left(\frac{\sum_{w \in \{S_i\}}(maxSim(w, S_j) * idf(w))}{\sum_{w \in \{S_i\}} idf(w)} + \frac{\sum_{w \in \{S_j\}}(maxSim(w, S_i) * idf(w))}{\sum_{w \in \{S_j\}} idf(w)}\right)$$

This metric is used for our analysis as it combines the semantic similarities of each text segment with respect to the other. For each word w in the segment S_i, we identify the word in segment S_j that has the highest semantic similarity, i.e. $maxSim(w, S_j)$, according to some pre-defined word-to-word similarity measures. Next, the same process is applied to determine the most similar word in S_j with respect to the words in S_i. The word similarities are then weighed with corresponding word specificities, summed up and normalized according to the length of each sentence.

Wordnet based similarity measures score well in recognizing semantic relatedness [7]. Pucher [9] has carried out the performance evaluation of all the wordnet based semantic similarity measures and found that *wup* [4] is one of the top performers in capturing semantic relatedness. We also chose *wup* because it is based on the path length between synsets of words and its performance is consistent across various parts-of-speech (POS). We used Python nltk.corpus[8] to implement wup. Pairwise semantic similarity $sim(S_i, S_j)$ or $s_{i,j}$ is a symmetric relation. Thus, we used the upper traingle of the similarity matrix (X) to reduce computational overhead.

$$\forall s_{i,j} \in X \implies \{s_{i,j} = s_{j,i}\}$$

We used clustering to proceed from a set of sentences to views containing similar content. The similarity-matrix (X) was given as an input to Python scipy-cluster[9] which uses Hierarchical Agglomerative Clustering (HAC). HAC was used because we can terminate the clustering when the values of the scoring parameter converge without explicitly specifying the number of clusters to output.

Each cluster comprises of sentences grouped according to the similarity measure $(s_{i,j})$ discussed above. Hence, it is logical to treat them as views discussing a specific issue. In the next section, we propose a quality parameter for the ranking and evaluation of views.

2.5 Ranking of Views

Qualitative parameter for ranking the views focuses on average pairwise similarity between constituent sentences of a view V in order to define its cohesion (C). We define cohesion as

$$C = \frac{\sum_{i,j \in V} s_{i,j}}{len(V)}$$

$$s_{i,j} = sim(T_i, T_j)$$

$V =$ set of sentences (T_i) comprising the view
$len(V) =$ number of sentences in the view.

As per our definition, higher the value of cohesion, greater is the content similarity between the sentences of a view. Our framework wanted to ascribe importance to views with maximum pairwise semantic similarity. Thus, we defined Most Relevant View (MRV) as the view with maximum value of cohesion, i.e., maximum content overlap amongst its constituent sentences. Outlier views (OV) represent the set of views containing a single sentence. They are termed as outliers because their semantic similarity with others is too low to have any meaningful grouping. We rank all the views in the non-increasing order of their cohesion. As their corresponding pair-wise similarity is zero, outlier views have a cohesion value of zero. Hence, we order outlier views according to their importance (I) scores.

2.6 Framework for Extracting Views

Algorithm 1 provides the steps involved in mining diverse views from a set of related articles. The articles are cleaned by parsing the html and removing duplicates. IR metrics such as TF-IDF are collected before calculating the importance (I) of each sentence. The sentences are ranked in the non-increasing order of their importance to pick the top n sentences. We calculate the pair-wise semantic similarity between the chosen sentences to cluster them. Clustering is used to generate semantically related views from a set of disparate sentences. We rank the views according to the quality parameter proposed by us.

3. EXPERIMENTAL EVALUATION

Extraction of Top-Ranking sentences requires the number of constituent sentences (n) as an input. The ideal range of values for an input parameter is the one which can maximize the cohesion of views and determining it is a critical part of our framework. Hence, we analysed the result data to find the relevant range for n.

An input parameter producing views where the median cohesion is greater than (or equal to) the mean is preferred. As the mean is influenced by the outliers in a dataset, the median being at least as high as the mean indicates consistency across the values of cohesion. If all the values of mean cohesion are greater than that of median, the input parameter yielding views with the maximum mean cohesion is preferred.

We collected statistics about the cohesion (mean, median), number of views, outliers etc. for values of n equal to 20, 25, 30, 35, 40, and 50. The results are presented in Table 5. ID indicates the dataset-ID (as per Table 1), TRS stands for the number of Top-Ranking sentences, V and O stand for the number of views and outliers respectively.

Figures 2 to 9 plot the variation in the mean and median cohesion in relation to the number of TRS (n). The value of n is plotted on the horizontal axis and the value of cohesion is plotted on the vertical axis. We can deduce from the graphs that the mean and median cohesion are peaking for $20 \leq n \leq 35$. The exact breakup of the value of n yielding the best cohesion for all the datasets is provided in Table 3.

As evident from our results, choosing more top-ranking sentences need not necessarily lead to views with better cohesion. To extract views with best cohesion one can start with

[8] http://nltk.googlecode.com/svn/trunk/doc/api/nltk.corpus-module.html
[9] http://code.google.com/p/scipy-cluster

a lower bound (e.g. 20) of top-ranking sentences and incrementally add x sentences until one reaches an upper bound (e.g. 35). Incremental clustering [3] can be used to obtain views. The cohesion values can be compared to present the set of views which yield the best cohesion. Below we present three views mined by our framework. The value of n for each view is the one which yields best cohesion for that dataset (as presented in Table 3)

Algorithm 1 Mining Diverse Views

```
Require: Related Articles $A$
Ensure: Ranked Views $V$ with $MRV$ and $OV$
for all $a$ in A do
    aClean ← ParseHTML(a)

    if aClean is not duplicate then
        ACLEAN ← ACLEAN + aClean
    else
        discard aClean
    end if
end for
for all $a$ in ACLEAN do
    a ← removeStopwords(a) //ranks.NL stopwords
    ASTEM ← ASTEM + stem(a) //nltk stemmer
end for
for all $a$ in ASTEM do

    for all word in $a$ do
        computeTFIDF(word)
    end for
end for
for all sentence in ASTEM do
    rankedSentences ← calculateImportance(sentence) //section 2.3
end for
topN ← pickTOPsentences(rankedSentences,$n$) //as per importance (I)
for all sentence1 as $s1$ in topN do
    for all sentence2 as $s2$ in topN do
        if ($s1$,$s2$) not in simMatrix then
            simMatrix ← simMatrix + calculateSimilarity($s1$,$s2$)
        end if
    end for
end for
rawViews←clusteringEngine(simMatrix)//scipy-cluster
for all view in rawViews do
    views←views+calculateCohesion(view)//section 2.5
end for
rankedViews ← rankByCohesion(views)
$MRV$ ← chooseMaxCohesion(rankedViews)
$OV$ ← chooseZeroCohesion(rankedViews)
```

Example 1 | fifa vuvuzela (7) | n: 35 | cohesion: 40.71 (MRV)
The true origin of the vuvuzela is disputed, but Mkhondo and others say the tradition dates back centuries - ”to our forefathers” - and involves the kudu.{5} The plastic trumpets, which can produce noise levels in excess of 140 decibels, have become the defining symbol of the 2010 World Cup. {12} For this reason, there is no doubt that the vuvuzela will become one of the legacies that Africa will hand over to the world after the world cup tournament, since the Europeans, Americans and Asians could not resist the temptation of using it and are seen holding it to watch their matches. {3} Have you ever found yourself in bed in a dark room with just a single mosquito for company? The buzzing sound of the vibrations made by the mosquito’s wings. {10} On the other hand, its ban will affect the mood of the host nation and, of course, other African countries at the world cup, because of the deep rooted emotions attached to it by fans. {3} This has sparked another controversy in the course of the tournament and has become the single item for discussion in the media since the LOC made that controversial statement on Sunday evening. {3}

Example 2 | swine flu india (2) | n: 25 | cohesion: 4.52 (Rank 4)
Patnaik, who created the image with the help of his students, on the golden beach has depicted the pig wearing a mask with the message ‘Beware of swine flu’. The sculpture was put on display late Thursday on the beach in Puri, 56 km from the state capital Bhubaneswar. {18} Of the six cases reported in Pune, three are students who contracted the virus in the school. {91}

Example 3 | the lost symbol (4) | n: 20 | cohesion: 40.02 (MRV)
I read the book as fast as I could. Of course as a Dan Brown classic, it was very interesting, exciting and made me wanting to read as fast as I could. {13} Every symbol, every ritual, every society, all of it, even the corridors and tunnels below Washington, DC, it’s all real. {3} I feel more connected to the message of this book (the reach and the power of the human mind) than I did to possibility that Jesus had a child. {12} Malakh is after secret knowledge guarded by the Masons and he’ll stop at nothing to get it. To that end he’s kidnapped Peter Solomon, the head of the Masonic order in Washington, DC. {1} Malakh is about to reach the highest level in the Masonic order, but even so, he knows he will not be admitted to the most secret secrets. Secrets that he’s sworn to protect. He is not what he seems to his fellow Masons. He’s lied to them. He has his own agenda. {2} [sic]

The first and third examples were ranked first (MRV) by our framework and the second one was ranked fourth. If we examine the first example, a user who does not know the term ‘vuvuzela’ can immediately glean that it is a plastic trumpet which caused quite a stir in the fifa world cup 2010. There are also some sentences which insinuate toward a likely ban and surrounding controversy. In an ideal scenario, we would like to group sentences about the ban and the controversy in another view, but as it stands now, our view describes the instrument and the impact of vuvuzela on the world cup and serves as a good introduction to a novice or as a concise issue capsule to a user who is already familiar with the topic.

Similarly, the second example which was ranked fourth by our framework talks about the repercussions of the disease swine flu on pune and puri (cities in India). The third example, ranked first, contains some positive opinions about the book ‘The Lost Symbol’ and also a sneak peek into the intentions of the character Malakh. *Additional example views are provided in the appendix.*

The average number of sentences across all the views was found to be 3.9 and the average number of views across all the datasets was found to be 4.88. Table 4 presents the breakup for each dataset. Mean (S) indicates the average number of sentences across all the views, and Mean (N) indicates the average number of views. The implementation of the framework as described in Algorithm 1 took an upper-bound of 4.2 seconds to run, with computeTFIDF and calculateImportance being the time consuming steps at 2.6 seconds.

The main difference between summarization and our framework is that we provide multiple diverse views as opposed to summarization which lacks such an organization. We also rank these views thereby allowing a user to just look at the Most Relevant View (MRV) or the top x views as per his convienience. As we provide the IDs of the source articles in each view, a user can also browse through them to know more about that view.

4. CONCLUSION

Users who want to browse the content of a topic on the world wide web (www) have to wade through diverse articles available on it. Though summarization is successful in condensing huge volume of information, it groups several issues pertaining to a topic together and lacks an organized representation of the underlying issues representing it. In this paper, we propose a framework to mine the multiple views addressed by a collection of articles. These views are easily navigable and provide the user a detailed snapshot of their topic of interest. Our framework extends the concept of clustering to the sentence or phrase level (as opposed to document clustering) and groups semantically related sentences together to organize content in a way that is different from text summarization.

In future, we want to determine the polarity of a view (positive/negative/neutral) by examining the adjectives in it. We also want to incorporate user feedback by means of clicks, time spent on a page (implicit) and ratings, numerical scores (explicit) to evaluate the performance of our framework and if possible, re-rank the views.

Dataset	:	Number of TRS
financial meltdown	:	25
swine flu india	:	25
israel attacks gaza	:	30
the lost symbol	:	20
hotel taj krishna	:	20
hotel marriott	:	25
fifa vuvuzela	:	35
gulf oil spill	:	30

Table 3: Breakup of n

Dataset	Mean (S)	Mean (N)
financial meltdown	3.91	3.17
swine flu india	4.26	5.67
israel attacks gaza	3.74	4.17
the lost symbol	4.16	5.33
hotel taj krishna	3.67	4.17
hotel marriott	3.82	5.83
fifa vuvuzela	3.56	5.33
gulf oil spill	4.21	5.00

Table 4: Mean values

5. REFERENCES

[1] J. C. F. Allison L. Powell. Using multiple views of a document collection in information exploration. In *CHI'98: Information Exploration Workshop*, 1998.

[2] R. K. Ando, B. K. Boguraev, R. J. Byrd, and M. S. Neff. Multi-document summarization by visualizing topical content. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 79–98, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[3] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 626–635, New York, NY, USA, 1997. ACM.

[4] Z. W. Department and Z. Wu. Verb semantics and lexical selection. In *In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, 1994.

[5] M. A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23:33–64, March 1997.

[6] R. Katragadda, P. Pingali, and V. Varma. Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm. In *CLIAWS3 '09: Proceedings of the Third International Workshop on Cross Lingual Information Access*, pages 46–52, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[7] R. Mihalcea and C. Corley. Corpus-based and knowledge-based measures of text semantic similarity. In *In AAAI '06*, pages 775–780, 2006.

[8] A. L. Powell and J. C. French. The potential to improve retrieval effectiveness with multiple viewpoints. Technical report, VA, USA, 1998.

[9] M. Pucher. Performance evaluation of wordnet-based semantic relatedness measures for word prediction in coversational speech. In *IWCS 6:Sixth International Workshop on Computational Semantics Tilburg, Netherlands*, 2005.

[10] A. Tombros, J. M. Jose, and I. Ruthven. Clustering top-ranking sentences for information access. In *in Proceedings of the 7 th ECDL Conference*, pages 523–528, 2003.

APPENDIX

Example 4 | gulf oil spill (8) | n: 35 | cohesion: 16.64 (Rank 2) BP and the Coast Guard are also using chemicals to disperse the oil, which for the most part is spread in a thin sheen. But the area of the sheen has expanded to more than 150 miles long and about 30 miles wide. {1} The Coast Guard confirmed that the leading edge of the oil slick in the Gulf of Mexico is three miles from Pass-A-Loutre Wildlife Management Area, the Reuters news agency reported. The area is at the mouth of the Mississippi River. {1} ”They’re going to be focusing on the root cause, how the oil and gas were able to enter the [well] that should’ve been secured,” he said. ”That will be the primary focus, how the influx got in to the [well].” {1}

Example 5 | hotel marriott (6)| n: 30 | cohesion: 15.23 (Rank 3) Well located hotel offering good view of the lake. The rooms are clean and comfortable and have all amenities and facilities of a 5 star hotel. The hotel is not overtly luxurious but meets all expectations of a business traveller. The Indian restaurant is uper and a must-try. {6} The food is excellent and like I said, if it were not for the smell and so-so servie, I would stay here. {14} The rooms are great. Well lit, loaded with amenities and the trademark big glass windows to look out.. The bathroom is trendy and looks fabulous with rain shower and a bathtub. {12} [sic]

Example 6 | swine flu india (2) | n: 25 | cohesion: 15.98 (Rank 3) Three new cases of swine flu were confirmed in the city on Sunday, taking the total number of those infected to 12 in the State. {5} ”Currently, it isn’t the flu season in India, but if the cases keep coming in even after the rains, it will clash with our flu season (post-monsoon and winter period) which could be a problem”, he said. {55} In Delhi, out of the four cases, three people, including two children aged 12, contracted the virus from a person who had the flu. {12}

Example 7 | financial meltdown (1) | n: 35 | cohesion: 0 (Outlier View) It has to be said: The model of the credit rating agencies has collapsed. Whether because of their unprofessionalism or inherent conflicts of interest, the fact that the agencies receive their pay from the companies they cover has bankrupted the system. {11}

Example 8 | israel attacks gaza (3) | n: 40 | cohesion: 0 (Outlier View) ”I heard the explosions when I was standing in the hall for protection. Suddenly, in a few seconds, all of the police and firemen were in the building,” said resident Rachel Mor, 25. {21}

ID	TRS	Mean (C)	Median (C)	V	O
1	20	17.58	17.6	3	10
	25	32.52	36.87	3	13
	30	11.23	11.23	2	15
	35	10.78	10.78	2	18
	40	12.5	16.74	3	20
	50	4.43	4.69	6	25
2	20	18.86	15.63	4	10
	25	15.6	15.6	4	13
	30	10.98	4.97	5	15
	35	14.16	5.12	7	18
	40	10.03	5.12	7	20
	50	11.42	4.79	7	25
3	20	13.32	4.52	3	12
	25	17.34	14.82	4	15
	30	19.38	21.56	4	18
	35	18.53	15.07	5	21
	40	7.11	5.1	4	24
	50	11.44	4.75	5	30
4	20	23.32	24.04	4	10
	25	16.55	16.3	6	13
	30	20.37	16.86	5	15
	35	7.18	5.07	5	18
	40	13.13	11.25	6	20
	50	8.46	4.54	6	25
5	20	10.61	10.61	2	10
	25	7.34	5.58	3	13
	30	5.48	5.58	3	15
	35	17.25	5.58	7	18
	40	12.02	6.59	4	20
	50	10.64	5.11	6	25
6	20	10.51	5.18	3	10
	25	19.83	15.23	5	13
	30	14.94	10.21	6	15
	35	14.55	10.37	6	18
	40	14	10.33	8	20
	50	7.87	4.47	7	25
7	20	11.52	5.09	4	10
	25	13.55	4.72	4	14
	30	11.9	4.73	5	16
	35	14.74	4.73	5	20
	40	8.35	4.59	6	26
	50	7	4.5	8	32
8	20	10.52	10.72	4	10
	25	13.95	10.55	4	14
	30	14.54	16.3	5	16
	35	12.94	10.61	6	19
	40	10.92	4.58	5	27
	50	11.81	4.72	6	34

Table 5: Results

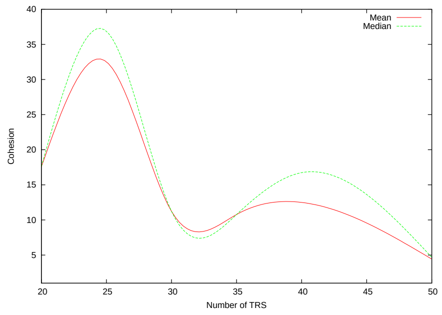


Figure 2: financial meltdown

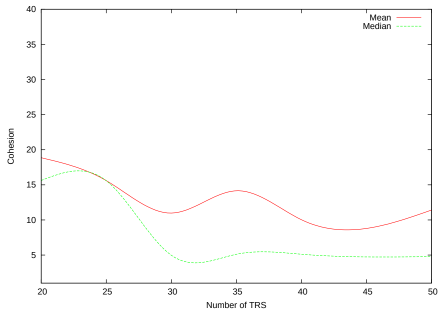


Figure 3: swine flu india

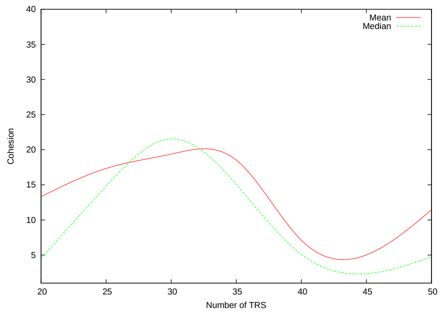


Figure 4: israel attacks gaza

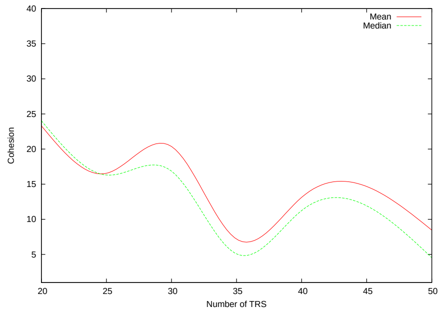


Figure 5: the lost symbol

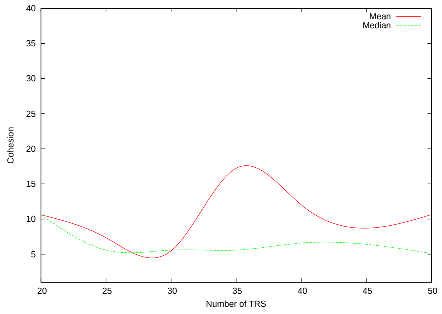


Figure 6: hotel taj krishna

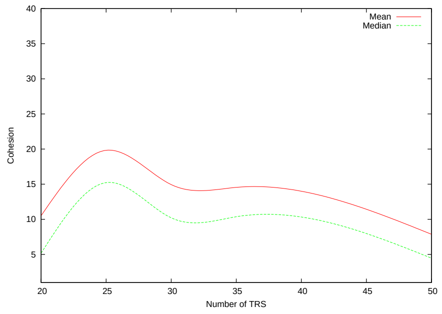


Figure 7: hotel marriott

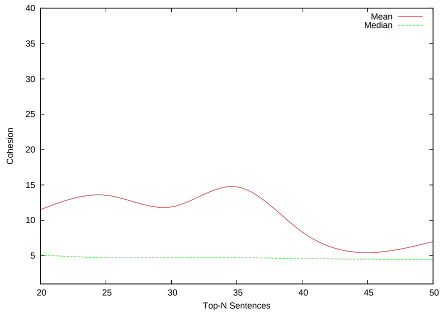


Figure 8: fifa vuvuzela

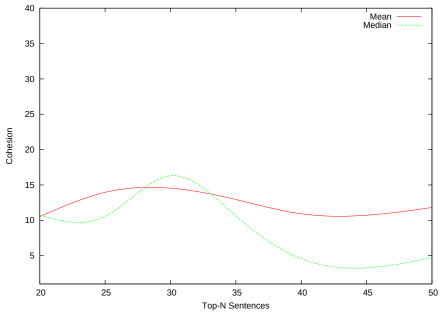


Figure 9: gulf oil spill