Semantically Driven Automatic Hyperlinking

Markus Pilzecker[1]

Abstract. This paper sketches some experiences and ideas, how a contemporary automatic hyperlinking system may be advantageously combined with sufficiently powerful methods that extract semantics from human language text[2].

1 Introduction

Automatic hyperlinking is when an automaton enriches a document with hyperlinks [5], [16].

1.1 Document Languages

As the *document language* we want to understand the language which makes up the document markup – clearly to be distinguished from the *human language*, which *text* content of the document is written in.

Even if, besides PDF and subtypes of SGML, document languages may also be something like LaTeX or FrameMaker.mif, we restrict ourselves here to document types for which we have browsers that make the documents, enhanced by hyperlinks, readable and navigable without a serious negative impact on document readability[3].

1.2 Scope

We want to focus our attention here to methods of automatic hyperlinking, which – at least to the larger part – base on an analysis of human language text. The pretty mechanistic understanding of automatic hyperlinking of the early days like e.g. table of contents generation or traversifying search engine results will only play a marginal role.

1.3 Hyperlinks

In most document languages, especially the subtypes of SGML, *hyperlinks* are understood as a navigable reference from a short piece of text in a source document to a *point* in a target document. Those two documents may happen to be the same.

An alternative understanding of the target end of such a hyperlink association may be a *text section*, which starts at the "physical" end of the link and ends at some well-defined location, e.g. the end of the document.

2 The Hyperlinker

An automatic *hyperlinker* generally acts as a filter, which takes a set of documents[4], performs some transformation on it and delivers the transformed set of documents, which may be fed to the browsers in question.

2.1 Whom Does It Serve?

The automatic hyperlinker may do its work at the side of the content provider[5] or at the side of the reader, who simply had to configure his browser to use the linker as proxy.

2.2 Automaton vs. Human

Since an automaton creates hyperlinks at a much lower cost than a human, one can cope with a much higher volume and complexity. Higher volume may [extensively] mean to process larger amounts of text or [intensively] to aim at a higher link density. Higher complexity may mean to serve more complex kinds of links[, e.g. one-to-many].

3 Document Worlds

A document world is called "closed", if the automatic hyperlinking engine knows the documents belonging to it; it is called "open", if not. Closed document worlds have some advantageous properties:

- it is possible to improve runtime efficiency by utilizing deterministic cache strategies, e.g. a link database
- it makes much more sense to ponder about topological properties of the link graph as a whole, i.e., the hyperlinker may implement a model for the paths via which your reader can traverse the document world
- it is in principle possible to achieve perfect link integrity

The latter point may be not too discriminative since even in an open document world you can achieve *approximately* perfect link integrity, if you construct your links much faster than the average document lifetime in your document world.

---

[1] DKFI, Saarbrücken, Germany, <mp@dfki.de>

[2] something, for which you also hear the term "text-mining" nowadays

[3] as opposed to, e.g., written hyperlinks like "see Charles Valentin Alkan: Le Festin d'Esope, variation no. 16 beat no. 7", where you have to start an odyssey to the next larger town's most profound music store to get your hands on the link target

[4] not necessarily in a form rendered ready-to-read. They may also be in a pre-publication status like a document object model.

[5] where it may also be part of a semi-automatic workflow, e.g. being used as an authoring tool

4 Linking Strategies

Since a hyperlink is a reference from a [source] piece of text to a [target] location in a document, a hyperlinker has to do both: determine the source end and determine the target end[s] of every hyperlink. And this assignment of targets to sources in its whole will be called *linking strategy* here.

Now, there is a huge variety of linking strategies out there, which, in case of the hyperlinker being a human, most often are only implicit. Anyway, the overwhelming part of them aims at implementing navigation paths along semantic relations between conceptual entities.

4.1 Examples

The variety of possible linking strategies poses hardly any limits to your imagination. Some examples may be:

- every term gets associated its corresponding entry in a dictionary for a certain foreign language
- every term gets associated its corresponding entry in the Encyclopaedia Britannica
- every company name gets associated the portal page of the company's web site
- every citation of a law gets associated the corresponding section of the law text itself
- every bibliographic reference gets associated the related document
- every nominalphrase gets associated a corresponding definition of this term
- every domain-specific noun phrase gets associated a handcrafted link target associated to a corresponding entry of a handcrafted thesaurus
- you build a [source-]document-driven semantic search engine, which interweaves your document into a web of hyperlinks to semantically related text sections of all-over the free, open web of http-accessible documents
- finally, you could also imagine a more statistical approach[6] of some future discipline like "corpus-hyperlinkuistics", where a well-trained learner puts hyperlinks according to what it has learned about hyperlinks from a human-hyperlinked corpus, it has examined earlier in its life

.

This variety of imaginable linking strategies makes it on the other hand difficult to get graspable, what[7] the very [conceptual] nature of a linking strategy is. Often it will turn out, that it's implicitly defined by the implementation of the algorithm along the lines of your customer's desires.

5 Ergonomy and GUI

5.1 Link Density

The denser[8] the hyperlinks lie in a document, the more important it gets, that your reader understands, according to which principle you constructed the link targets. This is, because every link is a path to escape the original flow of reading, and the more such escapes you give, the more reliable needs the path to be, you lead your reader to. Otherwise, he will soon find himself lost in some kind of link jungle and his only friend left is the back button of his browser.

5.2 Link Bundles

If one wants to make navigable more than about a handful of target locations, it proved to be quite helpful to offer the reader not only one target per link source, but a whole, possibly hierarchically[9] traversable, collection, a so-called *link bundle*. The advantage of link bundles over simple links is, that they drastically shorten the length of the path to the link targets in your world from $O(n)$ to $O(\log n)$, n being the number of targets to reach.

For SGML document types, link bundles may be implemented as ECMA[10]-scripted menues or as small intermediate pages containing the collection of target references. We found intermediate link pages to be more robust and more universally supported by basic browser features[, e.g. bookmarking].

Of course, the link [bundle] layout should reflect the results of the target-analysis and -ranking in some way. One of the means is to *type* [2] or classify target documents and to give every type a proper visualisation. You may also want to integrate your favourite automatic document classifier, [9], [11], [17] and construct the bundle layout based on its results.

If you feel that you need additional information per set of targets, it may also be possible to enrich every choice[11] with an automatically generated, short summary of target text sections of the respective cluster [13] or an automatically found topic [4].

6 A Prototypic Architecture Proposal

Most of the non-trivial linking strategies mentioned above end up in a task for which you can bring in your whole artillery of HLT.

A typical chain of tasks to perform may then be:

- Create an abstract syntax tree[12] of every source document
- find the locations of the source phrases. If your idea of what to select is semantically based, you have to map your ontological entities in mind to possible surface representations. Alternatively, you may stay near to the surface and decide to select potentially domain relevant phrases with something like tf∗idf, eventually improved by a word clustering engine, [8].
- for every such source phrase: map it to an ontological entity. This mapping essentially means "word sense disambiguation", which in turn requires some form of context analysis for the situation, the phrase occured at, [1].
- find a first selection of target documents, in which you ex-

[6] in contrast to the architectural one mentioned below
[7] beyond the definition above
[8] and automatically put hyperlinks tend to lie dense, because they are so cheap
[9] with the ergonomic maximum of 7 choices per selection
[10] a standardized subset of so-called JavaScript, see [3]
[11] representing that set of targets
[12] wrt the document language [, e.g. HTML]

pect to find *any*[13] serialisation[14] of entities[15], you want to refer to. A search engine with a good phrase search capability may do the trivial part.

- Create an abstract syntax tree[12] of every retrieved potential target document
- find the locations of possible serialisations of the desired target entities in the text sections[16] of the selected target documents
- perform a human language syntactical analysis [at least] of the text sections, you found the target serialisations at
- rank your target text sections based on their situation in the document's abstract syntax tree and on the results of the syntax analysis, [12]. E.g., you may only be interested in statements about generalization and mereonymical relations, like in [7], [15], and rank every verb phrase representing another relation to zero.

  Another alternative for this and the last item would be to base the ranking on the results of a term clustering algorithm operating on the text sections around your target serialisations.

  Another parameter that may further weight your ranking, is the authoritativeness of a document, which may be determined along the lines of [10].
- select those sections of documents, where your ranking algorithm delivers the highest density of valuable content.
- you may want to cluster your target text sections with your favourite document categorizer in order to get a more ergonomic link bundle layout.
- "annotate" your target documents at the beginning of the desired target text sections[17] in order to be able to address them.
- "annotate" your source document at the source end of the hyperlink association.

7 Some First Experiences

- – Assume, you are looking for targets, which give you some ontological information about, say, copper. You will find, that some of the best[18] target sections have a document structure, where the surface string "copper" appeares in some kind of header followed by something like a table, listing physical properties of this chemical element|material.

  Especially in documents of the exact sciences, we found the interesting terms themselves or [meta-]statements about the role of following text sections at prominent places in document markup. Such information is much easier to extract than finding something comparable in free-flowing [hl-]text.

  – HTML is an example of a document language, that has explicit markup for glossary entries. If it is not used for bookmarks, they very probably indicate, that a human author had the intention to give something like a definition for a term.
- document parsers are helpless when fed with something like the HTML output of, e.g., Microsoft Word, where almost all high level markup got lost, or with pages containing a substantial amount of text, which, for aesthetical reasons, has been rendered into images[19] . In such a case, you better rendered the document with a decent HTML-parser[20] and performed an OCR on its output.
- syntactical analysis by which you dig for semantics has to operate deep enough: e.g. you should be able to distinguish, if it was the Mobilcom, which took over the France Telecom or if it was the France Telecom, which took over Mobilcom. For this, you have at least to be able to distinguish subject and object[s], something that the simpler shallow HLT components cannot deliver.

  Anaphor resolution on the other hand seems not to be required, since the related phrase most often is near enough to play its own game.

8 Conclusion

We believe, that the various possible combinations of modern, semantics-based HLT methods with Automatic Hyperlinking can open up new fields of applications for the quite classic link-based web technology. Moreover the phrase "Semantic Web" gets a slightly new meaning, because the proposed combination of existing techniques can *discover* the semantics, which is hidden in form of serialized text. This could at least partially obsolve artificially enriching those documents with some kind of additional "exterior" semantic markup.

If one really bases hyperlink construction on the proposed semantic text analysis, hyperlinks may well be seen as a kind of semantic "annotation".

9 Acknowledgements

Many of the ideas, having been amalgamated here, originated and matured in common projects and discussions with Tillmann Wegst <wegst@dfki.de>, Thierry Declerck <declerck@dfki.de> and Hans Uszkoreit <uszkoreit.dfki.de>.

REFERENCES

[1] Varol Akman, 'On a proposal of strawson concerning context vs. 'what is said", (1999).

[2] J. Allan, 'Building hypertext using information retrieval', *Information Processing and Management*, **33**(2), 145..159, (1997).

[3] 'Standard ECMA-262 ECMAScript language specification', Standard ECMA-262, ECMA, 114, rue du Rhône, CH-1204 Geneva - Switzerland, (December 1999). http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM.

[4] Levent Ertöz, Michael Steinbach, and Vipin Kumar, 'Finding topics in collections of documents: A shared nearest neighbor approach', in *Text Mine'01, Workshop on Text Mining (1st SIAM International Conference on Data Mining)*, Midland Hotel, Chicago, USA, (April 2001). http://www-users.cs.umn.edu/~kumar/papers/snn14.pdf, http://citeseer.nj.nec.com/503235.html.

[13] meaning: "for any imaginable context situation"

[14] linguists usually would say "surface representation" here. The term "serialisation" is inspired by a CORBA-like communication model, where an object-oriented parser and its inverse counterpart transform a series of tokens into an [object-]ontology and vice versa.

[15] the idea of "query expansion"

[16] #PCDATA in case of SGML document languages

[17] if you are not content with referring to the begin of the document

[18] my human judgement

[19] widely to be found in web documents advertising the newest lifestyle products

[20] whose output is the "contractual" interface to user

[5] Eanass Fahmy and David T. Barnard. Adding hypertext links to an archive of documents, September 1990.
[6] Luc Goffinet and Monique Noirhomme-Fraiture, 'Automatic hypertext link generation based on similarity measures between documents', (1999).
[7] Marti A. Hearst, 'Automatic acquisition of hyponyms from large text corpora', Technical Report S2K-92-09, (1992).
[8] D. Hindle, 'Noun classification from predicate-argument structures', in *Proceedings of the 28st annual meeting of the Association for Computational Linguistics, ACL*, pp. 1268–1275, (1990).
[9] Makoto Iwayama and Takenobu Tokunaga, 'Cluster-based text categorization: a comparison of category search strategies', in *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, eds., Edward A. Fox, Peter Ingwersen, and Raya Fidel, p. 273..281, Seattle, US, (1995). ACM Press, New York, US.
[10] Jon M. Kleinberg, 'Authoritative sources in a hyperlinked environment', *Journal of the ACM*, **46**(5), 604–632, (1999).
[11] Daphne Koller and Mehran Sahami, 'Hierarchically classifying documents using very few words', in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)*, (February 1997). citeseer.nj.nec.com/koller97hierarchically.html.
[12] E. Morin. Automatic acquisition of semantic relations between terms from technical corpora, 1999.
[13] Annette Preissner, *Flexible Hybrid Summarization of Multilingual Markup Documents*, Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 2000. http://www.dfki.de/~noemi/mt.ps.gz.
[14] F. Rousselot, P. Frath, and R. Oueslati. Extracting concepts and relations from corpora, 1996.
[15] Gerda Ruge, 'Automatic detection of thesaurus relations for information retrieval applications', in *Foundations of Computer Science: Potential - Theory - Cognition*, pp. 499–506, (1997).
[16] Ross Wilkinson and Alan F. Smeaton, 'Automatic link generation', *ACM Computing Surveys*, **31**(4es), (1999).
[17] Lawrence W. Wright, Holly K. Grossetta Nardini, Alan R. Aronson, and Thomas C. Rindflesch, 'Hierarchical concept indexing of full-text documents in the UMLS Information Sources Map', *Journal of the American Society for Information Science*, **50**(6), 514..523, (1999). http://nls5.nlm.nih.gov/pubs/jasis98.pdf.