Application of Localized Similarity for Web Documents

Peter Reberšek
Zemanta
Celovška cesta 32
Ljubljana, Slovenia
peter.rebersek@zemanta.com

Mateja Verlič
Zemanta
Celovška cesta 32
Ljubljana, Slovenia
mateja.verlic@zemanta.com

Abstract

In this paper we present a novel approach to automatic creation of anchor texts for hyperlinks in a document pointing to similar documents. Methods used in this approach rank parts of a document based on the similarity to a presumably related document. Ranks are then used to automatically construct the best anchor text for a link inside original document to the compared document. A number of different methods from information retrieval and natural language processing are adapted for this task. Automatically constructed anchor texts are manually evaluated in terms of relatedness to linked documents and compared to baseline consisting of originally inserted anchor texts. Additionally we use crowdsourcing for evaluation of original anchors and automatically constructed anchors. Results show that our best adapted methods rival the precision of the baseline method.

1 Introduction

One of the features of hypertext documents are hyperlinks that point to other resources – pictures, videos, tweets, or other hypertext documents. A fairly familiar category of the latter is *related articles*; these usually appear at the end of a news article or a blog post with the title of the target document as anchor text. The target document is similar in content to original document; it may tell the story from another point of view, it may be a more detailed version of a part of the events in the original document, etc. Another category are the in-text links; these appear inside the main body of text and use some of the existing text as anchor. Ideally the anchor text is selected in such a way that it conveys some information about the target document; in reality sometimes just an adverb (e.g. here, there) is used, or even the destination URL may serve as anchor.

Our goal is to develop a system that automatically constructs in-text links, i.e. for a query document finds a target document and an appropriate part of the text of the query document that serves as the anchor text for the hyperlink. We want the target document to be similar in content to the query document and the anchor text to indicate that content.

There are many potential uses for such a system, especially for simplifying and streamlining document creation. This includes authors of blogs that may use the system for adding related content from other sources without exhausting manual search for such material. It may also be used when writing a scientific paper, automatically adding citations to other relevant papers inside the main body. This accelerates the writing, again reducing the time spent searching for possible existing research in the field. A citation can be considered an in-text link without a defined starting point.

We have addressed the problem in two steps, separately finding a similar document, and finding the anchor text for it. Since the retrieval of similar documents was a research focus for many years and is thus better researched, we have decided in this paper to focus on the placement of the anchor text for a link to a preselected document.

This paper is organized as follows: related work is discussed in Section 2, the methods, corpus, and evaluation are described in Section 3, followed by

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1399–1404,
Seattle, Washington, USA, 18-21 October 2013.

results and discussion in Section 4 and ending with conclusions in Section 5.

2 Related Work

Semantic similarity of textual documents offers a way to organize the increasing number of available documents. It can be used in many applications such as summarization, educational systems, finding duplicated bug reports in software testing (Lintean et al., 2010), plagiarism detection (Kasprzak and Brandejs, 2010), and research of a scientific field (Koberstein and Ng, 2006). Documents can vary in length from microblogs (Twitter) and sentences (Li et al., 2006; Koberstein and Ng, 2006) to paragraphs (Lintean et al., 2010) and larger documents (Budanitsky and Hirst, 2006).

There is also commercial software such as nRelate[1], Zemanta[2] and OpenCalais[3] with functionality that ranges from named entity recognition (NER) and event detection to related content. Publishers use in-house tools that offer automatic retrieval of in-house similar documents.

Most of the methods for comparing documents focus on the query document as a whole. The calculated score therefore belongs to the whole document and nothing can be said about more or less similar parts of the document. Our goal is to localize the similarity to a part of the query document, a paragraph, sentence, or even a part of the sentence that is most similar to another document. This part of the query document can then serve as anchor text for a hyperlink connection to the similar document.

Plagiarism detection methods (Alzahrani et al., 2012; Monostori et al., 2002) have a task of verifying the originality of the document. Extrinsic plagiarism detection methods compare two documents to determine if some of the material in one is plagiarised from the other. Methods range from simple exact substring matching to more advanced ones like semantic based methods that are able to recognize paraphrasing and refactoring (Alzahrani et al., 2012). These methods have localization of similarity already built-in as they are searching for parts of the text that seem to be plagiarised. We have focused on one such method, the winner of the PAN 2010 challenge (Kasprzak and Brandejs, 2010). This method uses shared n-grams from the two documents in order to determine if one of them is plagiarised.

Another similar research is automatic citation placement for scientific papers. Most of the work (Strohman et al., 2007; McNee et al., 2002) is concerned with putting citations at the end of the paper (non-localized), which is a task similar to inserting related articles for a news article at the of the text. There have been some attempts to place the citations in the main body of text (Tang and Zhang, 2009; He et al., 2011), typically used when referring to an idea or method.

Tang and Zang (2009) used a placeholder constraint: the query document must contain placeholders for citations, i.e. the places in text where citation might be inserted. Their method then just ranks all possible documents for a particular placeholder and chooses the best ranked document as a result. Documents are ranked on the basis of a learned topic model, obtained by a two-layer Restricted Boltzmann Machine.

He et al.(2011) made a step further towards generality of a citation location; they divide the text into overlapping windows and then decide which windows are viable citation context. The best method for deciding which citation context to use was a dependency feature model, an ensemble method using 17 different features and decision trees.

Named entity recognition (NER) also offers a useful insight into document similarity. If two documents share a named entity (NE), it is more likely they are similar. Detected NEs may also serve as anchor text for the link. NER is a fairly researched field (Finkel et al., 2005; Ratinov et al., 2011; Bunescu and Pasca, 2006; Kulkarni et al., 2009; Milne and Witten, 2008) and is also used in several commercial applications such as Zemanta, OpenCalais and AlchemyAPI[4], which are able to automatically insert links for a NE pointing to a knowledge base such as Wikipedia or IMDB. However, at this point they are unable to link to arbitrary documents, but may be useful in conjunction with other methods.

[1] nRelate: `http://www.nrelate.com/`

[2] Zemanta: `http://www.zemanta.com/`

[3] OpenCalais: `http://www.opencalais.com/`

[4] AlchemyAPI: `http://www.alchemyapi.com/`

3 Methodology

3.1 Corpus

We have chosen 100 web articles (posts) at random from the end of January 2012. We extracted the body and title of each document. All the present in-text links were also extracted and filtered. First, automatic filtering was applied to remove unwanted categories of links (videos, definition pages on wikipedia and imdb, etc.), and articles that were deemed too short for similarity comparison. The threshold was set at 200 words of automatically scraped body text of a linked document.

All the remaining links were manually checked to ensure the integrity of link targets. This way we collected 265 articles (hereinafter *related articles* - RA). A number of different methods were then used to calculate similarity rank and select the best part of the post text to be used as anchor text for a hyperlink pointing to the originally linked RA.

We have used CrowdFlower[5], a crowdsourcing platform, to evaluate how many of the 265 post–RA pairs were really related; the final corpus thus consisted of 236 pairs.

3.2 Evaluation

We have used each of the methods described in Subsection 3.3 to automatically construct anchor text for each of the 236 pairs of documents in the final corpus. If a method could not find a suitable anchor, no result was returned; on average there were 147 anchors per method. All the automatically created links were then manually scored by the authors with an in-house evaluation tool using scores and guidelines summarized in Table 1. To calculate precision and recall, we have counted scores 2 and 3 as positive result.

Additionally we crowdsourced the evaluation of results for some of the methods. For this task we prepared a special description of evaluation tasks and defined a set of questions for collecting results. We provided simplified guidelines for assigning scores to automatically created anchors and set a confidence threshold of 0.55 for an assignment to be considered valid. It is important to mention that the use of crowdsourcing for such tasks has to be carefully planned, because many issues related to monetary incentives, which are out of the scope of this paper, may arise.

[5]CrowdFlower: `http://crowdflower.com/`

Score	Description
0	Anchor does not signify anything about RA or gets it wrong
1	Some connection can be established (anchor is a shared Named Entity, Noun Phrase, Verb Phrase, etc.)
2	Anchor is a good estimation of RA topics, but not wholly (anchor is a non-main topic in RA)
3	RA topics can be directly inferred from the anchor

Table 1: Scores used for internal evaluation of automatically created anchors

3.3 Methods for constructing anchor texts

We have adapted a number of methods from a variety of sources to test how they perform for our exact purpose. Below is a short overview of the different methods used in this work.

3.3.1 Longest chunk

This method is based on natural language processing and extensively uses NLTK package (Bird et al., 2009); the text is first tokenized with the default NLTK tokenizer, and then POS tagged with one of the included POS taggers. After much testing, we have decided on a combination of Brill – Trigram – Bigram – Unigram – Affix – Regex backoff tagger with *noun* as default tag. The trainable parts of the tagger were trained on the included CoNLL 2000 tagged corpus.

Before chunking was applied, we also simplified some tags and removed some others to get a simpler structure of POS tags. We then used a regex chunker to find a sequence of a proper noun and a verb separated by zero or more other tokens. We have also tested a proper noun - verb - proper noun combination, but there were even fewer results, so this direction was abandoned.

3.3.2 Latent Semantic Indexing (LSI) based

A corpus is represented in LSI (Deerwester et al., 1990) as a large matrix of term occurrences

in individual documents. The rank of the matrix is then reduced using singular value decomposition that groups together terms that occur in similar context which should therefore account for synonyms.

We have used a tool called gensim (Řehůřek and Sojka, 2010) that enabled us to quickly train a LSI model using the whole corpus and index just the related articles. In order to localize the similarity and place an anchor, we split the source document into paragraphs and compute similarity scores between target document and each paragraph of the source document. We then split the paragraph with the highest score into sentences and again obtain scores for each. The sentence with the best score is then chosen as the result.

3.3.3 Sorted n-grams

Drawing on plagiarism detection, the winning method from the PAN 2010 (Kasprzak and Brandejs, 2010) seemed a viable choice. The basis of the method is comparing n-grams of the source and the destination documents. First, the text was again tokenized with NLTK, removed stopwords and tokens with two or less characters. Then overlapping n-grams were constructed. We have deviated from Kasprzak's merging policy and decided to merge two results if they are less than 20 tokens apart. We also required only one shared n-gram to consider the documents similar. Results were ranked based on the number of shared tokens within each.

3.3.4 Unigrams tf*idf

This method uses unigram tf*idf weighted scores. Since we had a closed system, we used corpus-wide frequencies; stopwords were also removed. We have scored tokens in the source document with tf*idf summary of the destination document; tokens not in summary are given a zero weight. We have experimentally determined that a summary of just top 150 tokens improves results. Sentences were ranked based on the sum of its tokens weights. We also included NEs from Zemanta API response for both source and destination document. Sentences containing shared NEs get their score multiplied by the sum of shared NE tf*idf weights. The result was then the sentence with the highest score.

	Manual		CrowdFlower	
	P	R	P	R
Original links	0.691	0.691	0.981	0.432
Sorted 5-grams	0.822	0.254		
Sorted 4-grams	0.741	0.352		
Sorted 3-grams	0.680	0.424	0.956	0.275
Longest Chunk	0.080	0.075	0.907	0.165
Unigrams tf*idf	0.626	0.242	0.882	0.127
LSI based	0.648	0.640		

Table 2: Precision and recall for manual and CrowdFlower evaluation

3.3.5 Baseline

Our baseline was a method that inserted links that were originally present in the source documents. This method was used to compare our automatic methods to what people are actually linking in the real world.

4 Evaluation Results and Discussion

Results are presented as precision and recall for different methods and both evaluations in Table 2. Empty cells in the table indicate that these methods were not evaluated using CrowdFlower. Recall is the fraction of relevant results out of all the possible results (236) and precision is the fraction of relevant results out of all the retrieved results.

The first thing we notice is the general disagreement between results from the authors and CrowdFlower workers; the latter tend to give higher scores, which leads to higher precision and recall. The reason for this might be in the authors' background knowledge and thus higher expectations.

As a contrast almost half of CrowdFlower workers stated they don't blog and of the rest, more than a third of them don't link out, i.e. do not use related articles. We also have only 74% median inter-annotator agreement leading us to believe that some of the annotators answered without being familiar with the question (monetary incentive issue).

Furthermore, CrowdFlower results for original links (our baseline) indicate that almost all of them were recognized as relevant, while our evaluators discarded 30% of them. Clearly seen in the results of different sorted n-grams methods is also the precision-recall trade-off.

5 Conclusion

Based on evaluation results and despite differences between the evaluators with background knowledge and the crowds, we can conclude that that our approach for automatic construction of in-text links rivals manual creation by professional writers and bloggers and is thus a promising direction for further research.

Acknowledgement

This work was partially funded by the Slovenian Ministry of Higher Education, Science and Technology, and the European Union – European Regional Development Fund.

References

S.M. Alzahrani, N. Salim, and A. Abraham. 2012. Understanding plagiarism linguistic patterns, textual features, and detection methods. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(2):133–149.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, March.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy.

Scott Deerwester, Susan T. Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C. Lee Giles. 2011. Citation recommendation without author supervision. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 755–764, New York, NY, USA. ACM.

Jan Kasprzak and Michal Brandejs. 2010. Improving the reliability of the plagiarism detection system lab report for pan at clef 2010.

Jonathan Koberstein and Yiu-Kai Ng. 2006. Using word clusters to detect similar web documents. In *Proceedings of the First international conference on Knowledge Science, Engineering and Management*, KSEM'06, pages 215–228, Berlin, Heidelberg. Springer-Verlag.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466, New York, NY, USA. ACM.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. on Knowl. and Data Eng.*, 18(8):1138–1150, August.

Mihai Lintean, Cristian Moldovan, Vasile Rus, and Danielle McNamara. 2010. The role of local and global weighting in assessing the semantic similarity of texts using latent semantic analysis. In *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference, Daytona Beach, FL.*

Sean M. McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, pages 116–125, New York, NY, USA. ACM.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA. ACM.

Krisztián Monostori, Raphael Finkel, Arkady Zaslavsky, Gábor Hodász, and Máté Pataki. 2002. Comparison of overlap detection techniques. In PeterM.A. Sloot, AlfonsG. Hoekstra, C.J.Kenneth Tan, and JackJ. Dongarra, editors, *Computational Science — ICCS 2002*, volume 2329 of *Lecture Notes in Computer Science*, pages 51–60. Springer Berlin Heidelberg.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Pro-*

ceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May. ELRA.

Trevor Strohman, W. Bruce Croft, and David Jensen. 2007. Recommending citations for academic papers. In *In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07*, pages 705–706.

Jie Tang and Jing Zhang. 2009. A discriminative approach to topic-based citation recommendation. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining*, volume 5476 of *Lecture Notes in Computer Science*, pages 572–579. Springer Berlin Heidelberg.