ANNOTATING DOCUMENTS WITH RELEVANT WIKIPEDIA CONCEPTS

Janez Brank, Gregor Leban, Marko Grobelnik
Artificial Intelligence Laboratory
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773778; fax: +386 1 4251038
e-mail: {janez.brank,gregor.leban,marko.grobelnik}@ijs.si

ABSTRACT

We describe an efficient approach for annotating a document with relevant concepts from the Wikipedia. A pagerank-based method is used to identify a coherent set of relevant concepts considering the input document as a whole. The proposed approach is suitable for parallel processing and can support any language for which a sufficiently large Wikipedia is available.

1 INTRODUCTION

Recent years have seen a growth in the use of semantic technologies. However, in many contexts we still deal with largely unstructured textual documents that lack explicit semantic information such as might be required for further processing with semantic technologies. This leads to the problem of semantic annotation or semantic enrichment as an important preparatory step before further processing of a document. Given a document and an ontology covering the domain of interest, the challenge is to identify concepts from that ontology that are relevant to the document or that are referred to by it, as well as to identify specific passages in the document where the concepts in question are mentioned.

A specific type of semantic annotation, known as *wikification*, involves using the Wikipedia as a source of possible semantic annotations [1][2]. In this setting, the Wikipedia is treated as a large and fairly general-purpose ontology: each page is thought of as representing a concept, while the relations between concepts are represented by internal hyperlinks between different Wikipedia pages, as well as by Wikipedia's category memberships and and cross-language links.

The advantage of this approach is that the Wikipedia is a freely available source of information, it covers a wide range of topics, has a rich internal structure, and each concept is associated with a semi-structured textual document (i.e. the contents of the corresponding Wikipedia article) which can be used to aid in the process of semantic annotation. Furthermore, the Wikipedia is available in a number of languages, with cross-language links being available to identify pages that refer to the same concept in different languages, thus making it easier to support multilingual and cross-lingual annotation.

The remainder of this paper is structured as follows. In Section 2, we present the pagerank-based approach to wikification used in our wikifier. In Section 3, we describe our implementation and present some experimental evaluaton. Section 4 contains conclusions and a discussion of possible future work.

2 PAGERANK-BASED WIKIFICATION

The task of wikifying an input document can be broken down into several closely interrelated subtasks: (1) identify phrases (or words) in the input document that refer to a Wikipedia concept; (2) determine which concept exactly a phrase refers to; (3) determine which concepts are relevant enough to the document as a whole that they should be included in the output of the system (i.e. presented to the user).

We follow the approach described by Zhang and Rettinger [1]. This approach makes use of the rich internal structure of hyperlinks between Wikipedia pages. A hyperlink can be thought of as consisting of a source page, a target page, and the *link text* (also known as the *anchor text*). If a source page contains a link with the anchor text *a* and the target page *t*, this is an indication that the phrase *a* might be a reference to (or representation of) the concept that corresponds to page *t*. Thus, if the input document that we're trying to wikify contains the phrase *a*, it might be the case that this occurrence of *a* in the input document also constitutes a *mention* of the concept *t*, and the concept *t* is a *candidate annotation* for this particular phrase.

2.1 Disambiguation

In the Wikipedia, there may be many different links with the same anchor text *a*, and they might not all be pointing to the same target page. For example, in the English-language Wikipedia, there are links with *a* = "Tesla" that variously point to pages about the inventor, the car manufacturer, the unit in physics, a band, a film, and several other concepts.

Thus, when such a phrase *a* occurs in an input document, there are several concepts that can be regarded as candidate annotations for that particular mention, and we have to determine which of them is actually relevant. This is the problem of *disambiguation*, similar to that of word sense disambiguation in natural language processing.

There are broadly two approaches to disambiguation, local and global. In the local approach, each mention is disambiguated independently of the others, while the global approach aims to treat the document as a whole and disambiguate all the mentions in it as a group. The intuition behind the global approach is that the document that we're annotating is about some topic, and the concepts that we use as annotation should be about that topic as well. If the document contains many mentions that include, as some of their candidate annotations, some car-related concepts, this makes it more likely that we should treat the mention of "Tesla" as a reference to Tesla the car manufacturer as opposed to e.g. a reference to Nikola Tesla or to Tesla the

rock band.

2.2 The mention-concept graph

To implement the global disambiguation approach, our Wikifier begins by constructing a *mention-concept graph* for the input document. (Some authors, e.g. [2], refer to this as a *mention-entity* graph, but we prefer to use the term "mention-concept graph" as some of the Wikipedia pages do not necessarily correspond to concepts that we usually think of as entities, and our wikifier does not by default try to exclude them.) This can be thought of as a bipartite graph in which the left set of vertices corresponds to mentions and the right set of vertices corresponds to concepts. A directed edge $a \to c$ exists if and only if the concept c is one of the candidate annotations for the mention a (i.e. if there exists in the Wikipedia a hyperlink with the anchor text a and the target c). A transition probability is also assigned to each such edge, $P(a \to c)$, defined as the ratio [number of hyperlinks, in the Wikipedia, having the anchor text a and the target c] / [number of hyperlinks, in the Wikipedia, having the anchor text a].

This graph is then augmented by edges between concepts, the idea being that an edge $c \to c'$ should be used to indicate that the concepts c and c' are "semantically related", in the sense that if one of them is relevant to a given input document, the other one is also more likely to be relevant to that document. Following [1], the internal link structure of the Wikipedia is used to calculate a measure of semantic relatedness. Informally, the idea is that if c and c' are closely related, then other Wikipedia pages that point to c are likely to also point to c' and vice versa. Let L_c be the set of Wikipedia pages that contain a hyperlink to c, and let N be the total number of concepts in the Wikipedia; then the semantic relatedness of c and c' can be defined as

$$SR(c, c') = 1 - [\log(\max\{|L_c|, |L_{c'}|\}) - \log|L_c \cap L_{c'}|] / [\log N - \log(\min\{|L_c|, |L_{c'}|\})].$$

In the graph, we add an edge of the form $c \to c'$ wherever the semantic relatedness SR(c, c') is > 0. The transition probability of this edge is defined as proportional to the semantic relatedness: $P(c \to c') = SR(c, c') / \Sigma_{c''} SR(c, c'')$.

This graph is then used as the basis of calculating a vector of pagerank scores, one for each vertex. This is done using the usual iterative approach where in each iteration, each vertex distributes its pagerank score to its immediate successors in the graph, in proportion to the transition probabilities on its outgoing edges:

$$PR_{new}(u) = \tau\, PR_0(u) + (1 - \tau)\, \Sigma_v\, PR_{old}(v)\, P(v \to u).$$

The baseline distribution of pagerank, PR_0, is used both to help the process converge and also to counterbalance the fact that in our graph there are no edges pointing into the mention vertices. In our case, $PR_0(u)$ is defined as 0 if u is a concept vertex; if u is a mention vertex, we use $PR_0(u) = z \cdot$ [number of Wikipedia pages containing the phrase u as the anchor-text of a hyperlink] / [number of Wikipedia pages containing the phrase u], where z is a normalization constant to ensure that $\Sigma_u\, PR_0(u) = 1$. We used $\tau = 0.1$ as the stabilization parameter.

The intuition behind this approach is that in each iteration of the pagerank calculation process, the pagerank flows into a concept vertex c from mentions that are closely associated with the concept c and from other concepts that are semantically related to c. Thus after a few iterations, pagerank should tend to accumulate in a set of concepts that are closely semantically related to each other and that are strongly associated with words and phrases that appear in the input document, which is exactly what we want in the context of global disambiguation.

2.3 Using pagerank for disambiguation

Once the pagerank values of all the vertices in the graph have been calculated, we use the pagerank values of concepts to disambiguate the mentions. If there are edges from a mention a to several concepts c, we choose the concept with the highest pagerank as the one that is relevant to this particular mention a. We say that this concept is *supported* by the mention a. At the end of this process, concepts that are not supported by any mention are discarded as not being relevant to the input document.

The remaining concepts are then sorted in decreasing order of their pagerank. Let the i'th concept in this order be c_i and let its pagerank be PR_i, for $i = 1, \ldots, n$. Concepts with a very low pagerank value are less likely to be relevant, so it makes sense to apply a further filtering step at this point and discard concepts whose pagerank is below a user-specified threshold. However, where exactly this threshold should be depends on whether the user wants to prioritize precision or recall. Furthermore, the absolute values of pagerank can vary a lot from one document to another, e.g. depending on the length of the document, the number of mentions and candidate concepts, etc. Thus we apply the user-specified threshold in the following manner: given the user-specified threshold value $\theta \in [0, 1]$, we output the concepts $c_1, \ldots, c_m$, where m is the least integer such that $\Sigma_{i=1..m}\, PR_i^2 \geq \theta\ \ \Sigma_{i=1..n}\, PR_i^2$. In other words, we report as many top-ranking concepts as are needed to cover θ of the total sum of squared pageranks of all the concepts. We use $\theta = 0.8$ as a broadly reasonable default value, though the user can require a different threshold depending on their requirements.

For each reported concept, we also output a list of the mentions that support it.

2.4 Treatment of highly ambiguous mentions

Our wikifier supports various minor heuristics and refinements in an effort to improve the performance of the baseline approach described in the preceding sections.

As described above, anchor text of hyperlinks in the Wikipedia is used to identify mentions in an input document (i.e. words or phrases that may support an annotation). One downside of this approach is that some words or phrases occur as the anchor text of a very large number of hyperlinks in the Wikipedia and these links point to a large number of different Wikipedia pages. In other words, such a phrase is highly ambiguous; it is not only unlikely to be disambiguated correctly, but also introduces noise into the mention-concept graph by introducing a large number of concept vertices, the vast majority of which will be completely irrelevant to the input document. This also slows down the annotation process

by increasing the time to calculate the semantic relatedness between all pairs of candidate concepts.

We use several heuristics to deal with this problem. Suppose that a given mention a occurs, in the Wikipedia, as the anchor text of n hyperlinks pointing to k different target pages, and suppose that n_i of these links point to page c_i (for $i = 1, \ldots, k$). We can now define the entropy of the mention a as the amount of uncertainty regarding the link target given the fact that its anchor text is a: $H(a) = -\Sigma_{i=1..k} (n_i/n) \log(n_i/n)$. If this entropy is above a user-specified threshold (e.g. 3 bits), we completely ignore the mention as being too ambiguous to be of any use. For mentions that pass this heuristic, we sort the target pages in decreasing order of n_i and use only the top few of them (e.g. top 20) as candidates in our mention-concept graph. A third heuristic is to ignore candidates for which n_i itself is below a certain threshold (e.g. $n_i < 2$), the idea being that if such a phrase occurs only once as the anchor text of a link pointing to that candidate, this may well turn out to be noise and is best disregarded.

Optionally, the Wikifier can also be configured to ignore certain types of concepts based on their Wikidata class membership. This can be useful to exclude from consideration Wikipedia pages that do not really correspond to what is usually thought of as entities (e.g. "List of…" pages).

Another heuristic that we have found useful in reducing the noise in the output annotations is to ignore any mention that consists entirely of stopwords and/or very common words (top 200 most frequent words in the Wikipedia for that particular language). For this as well as for other purposes the text processing is done in a case-sensitive fashion, which e.g. allows us to ignore spurious links with the link text "the" while processing those that refer to the band "The The".

2.5. Miscellaneous heuristics

Semantic relatedness. As mentioned above, the definition of semantic relatedness of two concepts, $SR(c, c')$, is based on the overlap between the sets L_c, $L_{c'}$ of immediate predecessors of these two concepts in the Wikipedia link graph. Optionally, our Wikifier can compute semantic relatedness using immediate successors or immediate neighbours (i.e. both predecessors and successors) instead of immediate predecessors. However, our preliminary experiments indicated that these changes do not lead to improvements in performance, so they are disabled by default.

Extensions to disambiguation. Our Wikifier also supports some optional extensions of the disambiguation process. As described above, the default behavior when disambiguating a mention is to simply choose the candidate annotation with the highest pagerank value. Alternatively, after any heuristics from section 2.4 have been applied, the remaining candidate concepts can be re-ranked using a different scoring function that takes other criteria besides pagerank into account. This is an opportunity to combine the global disambiguation approach with some local techniques. In general, a scoring function of the following type is supported:

$$score(c|a) = w_1 f(P(c|a)) PR(c) + w_2 S(c, d) + w_3 LS(c, a)$$

Here, a is the mention that we're trying to disambiguate, and c is the candidate concept that we're evaluating. $P(c|a)$ is the probability that a hyperlink in the Wikipedia has c as its target conditioned on the fact that it has a as its anchor text. $f(x)$ can be either 1 (the default), x, or $\log(x)$. $PR(c)$ is the pagerank of c's vertex in the mention-concept graph. $S(c, d)$ is the cosine similarity between the text of the input document d and of the Wikipedia page for the concept c. $LS(c, a)$ is the cosine similarity between the context (e.g. previous and next 3 words) in which a appears in the input document d, and the contexts in which hyperlinks with the target c appear in the Wikipedia. Finally, w_1, w_2, w_3 are weight constants. However, our preliminary experiments haven't shown any improvements from the addition of these heuristics, so they are disabled by default ($f(x) = 1$, $w_2 = w_3 = 0$) to save computational time and memory (storing the link contexts needed for the efficient computation of LS has turned out to be particularly memory intensive).

3 IMPLEMENTATION AND EVALUATION

3.1. Implementation

Our implementation of the approach described in the preceding section is running as a web service and can be accessed at http://wikifier.org. The approach is suitable for parallel processing as annotating one document is independent of annotating other documents, and any shared data used by the annotation process (e.g. the Wikipedia link graph, and a trie-based data structure that indexes the anchor text of all the hyperlinks) need to be accessed only for reading and can thus easily be shared by an arbitrary number of worker threads. This allows for a highly efficient processing of a large number of documents.

Our implementation currently processes on average more than 500,000 requests per day (the total length of input documents averages about 1.2 GB per day), including all the documents from the JSI Newsfeed service [3]. The output is used among other things as a preprocessing step by the Event Registry system [4]. The wikifier currently supports all languages in which a Wikipedia with at least 1000 pages is available, amounting to a total of 134 languages. Admittedly, 1000 pages is much too small to achieve an adequate coverage; however, about 60 languages have a Wikipedia with at least 100,000 pages, which is already enough for many practical applications.

Annotations are returned in JSON format and can optionally include detailed information about support (which mentions support each annotation), alternative candidate annotations (concepts that were considered as candidates during the disambiguation process but were rejected in favour of some other more highly scored concept), and WikiData/DbPedia class membership of the proposed annotations. Thus, the caller can easily implement any desired class-based postprocessing.

3.2. Evaluation

One way to evaluate wikification is to compare the set of annotations with a manually annotated gold standard for the same document(s). Performance can then be measured using

metrics from information retrieval, such as precision, recall, and the F_1-measure, which is defined as the harmonic mean of precision and recall. We used a manually annotated set of 1393 news articles that was made available from the authors of the AIDA system and was originally used in their experiments [2]. This manually annotated dataset excludes, by design, any annotations that do not correspond to named entities. Since our wikifier does not by default distinguish between named entities and other Wikipedia concepts, we have explicitly excluded non-entity concepts (based on their class membership in the WikiData ontology) from the output of our Wikifier for the purposes of this experiment. In addition to our wikifier, we obtained annotations from the following systems: AIDA [2], Waikato Wikipedia Miner [6], Babelfy [7], Illinois [8], and DbPedia Spotlight [9].

	Gold	JSI	AIDA	Waikato	Babelfy	Illinois	Spotlight
Gold	1.000	0.593	0.723	0.372	0.323	0.476	0.279
JSI		1.000	0.625	0.527	0.431	0.489	0.363
AIDA			1.000	0.372	0.352	0.434	0.356
Waikato				1.000	0.481	0.564	0.474
Babelfy					1.000	0.434	0.356
Illinois						1.000	0.376
Spotlight							1.000

Table 1: F_1 measure of agreement between the various wikifiers and the gold standard.

Table 1 shows the agreement not only between each of the wikifiers and the gold standard, but also between each pair of wikifiers (the lower left triangle of the matrix is left empty as it would be just a copy of the upper right triangle, since the F_1-measure is symmetric). As this experiment indicates, our wikifier ("JSI" in the table) performs slightly worse than AIDA but significantly better than the other wikifiers. Furthermore, it turns out that there is relatively little agreement between the different wikifiers, which indicates that wikification itself is in some sense a vaguely defined task where different people can have very different ideas about whether a particular Wikipedia concept is relevant to a particular input document (and should therefore be included as an annotation) or not, which types of Wikipedia concepts can be considered as annotations (e.g. only named entities or all concepts), etc. Possibly the level of agreement could be improved by fine-tuning the settings of the various wikifiers; in the experiment described above, default settings were used.

4 CONCLUSIONS AND FUTURE WORK

We have presented a practical and efficient approach to Wikification that requires no external data except the Wikipedia itself, that can deal with documents in any language for which the Wikipedia is available, and that is suitable for a high-performance, parallelized implementation.

The approach presented here could be improved along several directions. One significant weakness of the current approach concerns the treatment of minority languages. When dealing with a document in a certain language, we need hyperlinks whose anchor text is in the same language if we are to identify mentions in this input document. Thus, if the document is in a language for which the Wikipedia is not available at all, it cannot be wikified using this approach; and similarly, if the Wikipedia is available in this language but is small, with a small amount of text, low number of pages, and generally poor coverage, the performance of wikification based on this will be low. One idea to alleviate this problem would be to optionally allow a second stage of processing, in which Wikipedias in languages other than the language of the input document would also be used to identify mentions and provide candidate annotations. This might improve coverage especially of concepts that are referred to by the same words or phrases across multiple languages, as is the case with some types of named entities. For the purposes of pagerank-based disambiguation in this second stage, a large common link-graph would have to be constructed by merging the link-graphs of the Wikipedias for different languages. This can be done by using the cross-language links which are available in the WikiData ontology, providing information about when different pages in different languages refer to the same concept.

Another interesting directon for further work would be to try incorporating local disambiguation techniques as a way to augment the current global disambiguation approach. When evaluating whether a mention *a* in the input document refers to a particular concept *c*, the local approach would focus on comparing the context of *a* to either the text of the Wikipedia page for *c*, or to the context in which hyperlinks to *c* occur within the Wikipedia. Preliminary steps taken in this direction in Sec. 2.5 did not lead to improvements in performance, but this subject is worth exploring further. Instead of the bag-of-words representation of contexts, other vector representations of words could be used, e.g. word2vec [5].

Acknowledgments

This work was supported by the Slovenian Research Agency as well as the euBusinessGraph (ICT-732003-IA) and EW-Shopp (ICT-732590-IA) projects.

References

[1] L. Zhang, A. Rettinger. *Final ontological word-sense-disambiguation prototype.* Deliverable D3.2.3, xLike Project, October 2014.

[2] J. Hoffart, M. A. Yosef, I. Bordino, *et al.* Robust disambiguation of named entities in text. *Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, 2011, pp. 782–792.

[3] M. Trampuš, B. Novak. Internals of an aggregated web news feed. *Proc. SiKDD 2012*.

[4] G. Leban, B. Fortuna, J. Brank, M. Grobelnik. Event registry: Learning about world events from news. *Proc. of the 23rd Int. Conf. on the World Wide Web* (WWW 2014), pp 107–110.

[5] T. Mikolov, K. Chen, G. Corrado, J. Dean. *Efficient estimation of word representations in vector space*. Arxiv.org, 2013.

[6] D. Milne, I. H. Witten. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194:222–239 (January 2013).

[7] A. Moro, A. Raganato, R. Navigli. Entity linking meets word sense disambiguation: A unified approach. *Trans. of the Assoc. for Comp. Linguistics*, 2:231–234 (2014).

[8] L. Ratinov, D. Roth, D. Downey, M. Anderson. Local and global algorithms for disambiguation to Wikipedia. *Proc. of the 49th Annual Meeting of the Assoc. for Comp Linguistics: Human Language Technologies* (2011), pp. 1375–84.

[9] J. Daiber, M. Jakob, C. Hokamp, P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. *Proc. of the 9th Int. Conf. on Semantic Systems*, 2013.