Fact Discovery in Wikipedia

Sisay Fissaha Adafre
School of Computing, Dublin City University
sadafre@computing.dcu.ie

Valentin Jijkoun Maarten de Rijke
ISLA, University of Amsterdam
jijkoun,mdr@science.uva.nl

Abstract

We address the task of extracting focused salient information items, relevant and important for a given topic, from a large encyclopedic resource. Specifically, for a given topic (a Wikipedia article) we identify snippets from other articles in Wikipedia that contain important information for the topic of the original article, without duplicates. We compare several methods for addressing the task, and find that a mixture of content-based, link-based, and layout-based features outperforms other methods, especially in combination with the use of so-called reference corpora that capture the key properties of entities of a common type.

1. Introduction

Traditionally, to get answers to informational questions, people have turned to reference works. With the growth of the web, reference works have to some degree been replaced by general web search engines. However, Wikipedia,[1] the free online editable encyclopedia, has become one of the largest reference works ever, making it a natural target for informational queries. Wikipedia is organized around articles, one per topic, that gather the most important aspects of the topic. Nevertheless, important information for the topic of a given article may also be found in *other* Wikipedia articles, either in the same or other languages, reflecting national interests and/or expertise levels. Such information, "spread" throughout Wikipedia, might not have been incorporated in the topic article, e.g., because of the decentralized nature of Wikipedia's authoring process.

We describe search aids to harvest material on a given topic t that is not contained in the Wikipedia article on t, but is available from other Wikipedia articles in the same language and/or from articles in other languages. Such a system provides useful functionality to several types of Wikipedia users: to readers and to editors, that are interested in getting a full picture of a topic in the entire encyclopedia. This information access task has been evaluated at the Question Answering Using Wikipedia (WiQA) pilot that ran at CLEF 2006 [17]. In our approach to this task, we try to use the properties of Wikipedia itself: namely, the article layout structur, the hyperlinks between the articles and the manual categorization of the articles provided by the Wikipedia editors. We use this structural information to determine which other articles discuss the topic we are targeting, and to estimate which information snippets are likely to be *important* for topics of a certain category.

Our main contributions in this paper are the following. We describe a system that operates in four stages: (1) identify additional sentences relevant for t; (2) single out sentences that are important for t; (3) remove sentences that contribute information that is already contained in the article devoted to t, and (4) make sure that no two sentences in the resulting list contain the same information.

We present and compare several algorithms for the subtasks, combining methods based on generative language modeling for identifying relevant sentences with methods that exploit Wikipedia's unique features (link, category, and layout structure) as part of a graph-based algorithm to estimate the importance of sentences. We report on the evaluation of the proposed algorithms within the WiQA pilot.

The remainder of this paper is organized as follows. In §2 we provide a description of the task we address. We describe related work in §3 and our architecture and main algorithms in §4. A detailed evaluation of our algorithms is presented in §5. We conclude in §6.

2. The Task

We follow [17] in defining the task of an automatic system. Specifically, the task is to locate information snippets in Wikipedia that are: (1) outside the given source article; (2) in one of the specified target languages; and (3) substantially new w.r.t. the information contained in the source article, and important for the topic of the source article, in other words, worth including in the content of (future editions of) the article. We focus on the English monolingual version of this task, where the source and target languages are both English. Our monolingual system is, nevertheless,

[1] URL: `http://www.wikipedia.org`

language independent.

We use the terms *relevant* and *important* to characterize sentences at different stages of processing, and with different degrees of informational quality; *important* sentences are assumed to be *relevant* but not conversely. E.g., the article on *Vertigo Records* mentions that another label (*Philips Records*) used this name in the sixties—this information is missing in the Wikipedia article on *Philips Records*, though it is clearly important for the topic. On the other hand, the information that *Harvest Records was founded... to compete with Philips Records, among others* (provided in the article on *Harvest Records*), may well be relevant but not important for the topic *Philips Records*.[2]

3. Related Work

Related work comes in several areas: *access to Wikipedia* and *information retrieval*, specifically, in *question answering* and *summarization*. Quite diverse methods for acquiring important information for a given topic have been proposed in the literature. It is beyond the scope of this paper to provide a detailed comparison with our methods. Instead, we abstract away from the implementation details and focus on the core techniques typically applied by some of these methods. We apply these methods to our data and report the results. All our experiments are done using the resources made available as part of WiQA 2006.

Access to Wikipedia According to [29], the current content of Wikipedia is not amenable to automatic interpretation or reasoning since the bulk of the content is unstructured textual data with limited semantic annotations. In order to achieve focused access to Wikipedia, they propose an extention of MediaWiki that allows users to add semantic annotation to the existing content of Wikipedia, implying additional work on the part of the user. This may be unwanted: ease of editing or changing articles contributes to the success of Wikipedia [7]. Also, the success of such a proposal depends on the acceptance of the suggested extensions by the community. Although [29]'s goals overlap with ours, i.e., improving access to Wikipedia, we adopt a different, *bottom up* and *data-driven*, approach. Our methods assist users in their search for information without the need to change established reading and authoring practices. A number of tools and techniques have been developed in the areas of natural language processing, information extraction, and machine learning for mining useful semantic information from plain text. Today, these can already be used to automate the task of harvesting information about a topic that supplements a source page.

Information retrieval Wikipedia has become a fruitful source for research in information extraction; see, e.g., [24]. While our algorithms share several features with some of this work, it is more closely related to research in question answering, summarization, and novelty checking.

Question answering (QA) QA has attracted a great deal of attention, especially since the launch of the QA track at TREC in 1999. While significant progress has been made in technology for answering general factoids (e.g., *How fast does a cheetah run?*), there is a need to go beyond such factoids [28]. At the TREC QA track this has been recognized through the introduction of definition questions and of so-called "other" questions. Similar scenarios were examined at the Question Answering Using Wikipedia (WiQA) pilot at CLEF 2006 [17]. One of the tasks at the 2004 edition of the Document Understanding Conference [10] was to provide a short biographical summary in response to "Who is X?" There are various strategies for answering such definition questions as well as the earlier "other" questions. Some systems implement pattern matching techniques for identifying potential answers, based on either surface or linguistic structures [8, 15, 25]. Others rely on knowledge bases built through offline mining of corpora, again based on surface patterns [14], or deeper linguistic analyses for extracting facts from a corpus [18]. One of our importance estimation methods in §4 ("Word Overlap") is similar to a method introduced in [1], who used Wikipedia as "an importance model" in answering "other" questions. More recently, machine learning approaches have been explored [3, 4].

Summarization The graph-based part of one of our importance estimation methods (see §4) is related to ideas from extractive summarization, particularly graph-based summarization techniques and feature-based biographical information extraction [12]. LexRank [11] and TextRank [22] incorporate graph-based methods for ranking sentences based on their relevance to document summaries. Although the methods as originally proposed are not appropriate for our task of identifying important descriptions, a modification, combined with our idea of reference corpora, can be effectively applied to our task. This usage of Wikipedia is similar to ideas discussed in [12], but instead of explicitly generating highly specific occupation related lexical features, we use Wikipedia in a data-driven way to generate reference corpora on any topic for which a Wikipedia category exists, and leave the topic-related knowledge implicit.

Novelty checking Within the setting of the TREC Novelty track [26], a number of methods for identifying relevant and novel sentences for a topic have been used, many of them based on inter-sentence similarity measures [2]. Novelty detection largely depends on the quality of relevant sentence identification step, thus differentiating our task from the more restricted novelty checking task.

Finally, Wikipedia is now used as the document collection for several retrieval evaluation efforts at CLEF and

[2] See `http://en.wikipedia.org/wiki/`*title of the article*.

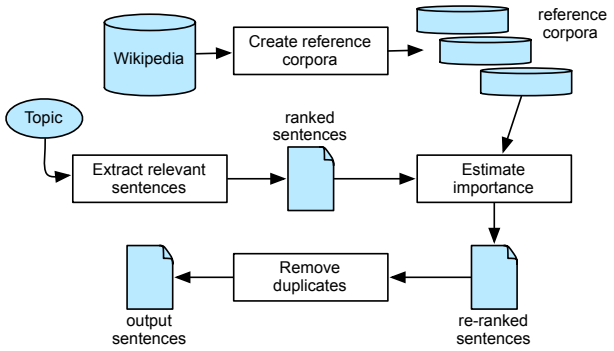

Figure 1. System description.

INEX. The WiQA task at CLEF assesses the task we address [17]. At INEX 2006, the ad hoc task used an XML-ified version of (the English) Wikipedia; in response to a query, systems could return arbitrary XML elements from the collection [16]. Unlike the INEX task, we deal with a fixed unit of retrieval, that need to meet additional requirements over and above the ones considered at INEX.

4. Modeling the Tasks

The fact discovery task may be factored into subtasks, each representing a phase in the processing pipeline: (1) identifying relevant sentences; (2) estimating sentence importance, and (3) removing redundant sentences.

Fig. 4 gives an overview of our approach to the task, with modules corresponding to the subtasks. First, before the topics are known, the system uses the category structure of Wikipedia to create a reference corpus for each category (see below). At query time, given a topic t, the system extracts sentences s relevant to the topic and assigns relevance scores $score_R(s,t) \in [0,1]$. Then, we estimate the importance of the relevant sentences—$score_I(s,t) \in [0,1]$—using the reference corpora, and rerank them by the combined relevance and importance score:

$$score_{RI}(s,t) = score_R(s,t) + score_I(s,t) \qquad (1)$$

Finally, given a list of sentences s_1, s_2, ..., s_n ordered by $score_{RI}(s_i,t)$, we determine the final score of s_i by subtracting s_i's redundancy score from its combined score:

$$score(s_i,t) = score_{RI}(s_i,t) - redundancy(s_i,t), \qquad (2)$$

$$redundancy(s_i,t) = \max_{s' \in T \cup \{s_1,\ldots,s_{i-1}\}} sim(s,s'), \qquad (3)$$

where t is the topic for which we are seeking additional information, T is the set of sentences in t, and $sim(s,s')$ is the similarity between s and s' calculated as the Jaccard coeffient [5], i.e., the number of common terms in s and s' divided by the total number of terms in s and s'. Our final scoring method is similar to, but different from, the maximal marginal relevance (MMR) method [6]; see also [20].

We will describe the calculation of the scoring function $score$ in three stages—the relevance part $score_R$ is detailed in §4.1 below, the importance part $score_I$ in §4.2, and our method for computing redundancy is detailed in §4.3.

For each stage, we present different approaches for accomplishing the corresponding subtask. Our emphasis will be on estimating sentence importance: as pointed out before, the results of [2] indicate that the effectiveness of methods for redundancy removal largely depends on the accuracy of earlier steps in the pipeline—in our case, finding important relevant sentences.

4.1 Identifying relevant sentences

The first step of the processing pipeline is to identify an initial set of candidate sentences and score them with respect to their relevance. We consider two strategies: *link-based retrieval* and *vector space retrieval*.[3] In both cases, we first identify articles that are relevant to a given topic t, and then extract sentences from these articles.

Link-based retrieval The link structure, more particularly, the structure of incoming links (similar to Wikipedia's "What links here" feature), provides a simple mechanism for identifying relevant articles. If an article contains a reference (a hyperlink) to a given topic t then it is likely to contain relevant information about t. Since hyperlinks are created by humans, this approach tends to produce little noise. However, due to inconsistencies in manual processing and editing requirements, not all mentions of a topic may be hyperlinked which may hurt recall. E.g., if an article mentions a particular topic with its own dedicated article, Wikipedia's guidelines recommend that only the first mention uses an explicit hyperlink and subsequent mentions are given in plain text. Often, these subsequent mentions also use a different form of reference to the topic: e.g., John Lennon may be mentioned with a hyperlink to the *John Lennon* article, and later using *Lennon* or just *he*. Moreover, Wikipedia contains a number of "redirect" pages providing alternative ways of referring to an entity: e.g., the page *John Winston Lennon* redirects to the article *John Lennon* which contains the actual information about the musician.

State-of-the-art coreference resolution techniques that use deep linguistic analysis do not scale up to corpora of the size of Wikipedia [27]. Instead, we devised a simple coreference resolution method for a particular frequent class of coreferences: references by last name (for persons). This method checks whether a person name appears at least once in the article as the anchor text of a hyperlink. If so, the last

[3] Based on results of [2], further explorations of additional sentence retrieval models does not seem promising.

token of the name is taken as the person's last name. All occurences of the last name in the article are then replaced by an explicit hyperlink. Similarly, we replace all links to redirect pages with links to actual content articles.

Given a topic t, we identify in Wikipedia all articles containing at least one hyperlink to t. For each such article a we perform coreference resolution as described above and extract all sentences containing a hyperlink to the topic. Each extracted sentence is then assigned a relevance score: the probability of the title of t generated by the unigram language model of the article a, using the maximum likelihood estimate. The resulting relevance scores of sentences are subsequently normalized between 0 and 1.

Vector space retrieval The second method for retrieving relevant sentences is based on the standard vector space model as implemented in Lucene [21]. We index the content of Wikipedia articles and retrieve relevant articles using the title of the topic t as the query. For the retrieved articles we perform coreference resolution as in the link-based method above. We split the articles into sentences and only retain sentences containing occurrences of the title of t. Unlike the link-based retrieval approach outlined above, the vector space method allows us to retrieve sentences from articles which do not contain explicit hyperlinks to the topic, but still contain mentions of the topic title. As a result, the vector space method is more recall-oriented, identifying more relevant sentences than the link-based method. The relaxed criterion also means that the method is likely to pick up more noise, especially in case of ambiguous names such as *Lennon* which occurs in the titles of 13 different Wikipedia articles. The retrieval status value produced by Lucene is normalized to a score between 0 and 1.

For efficiency purposes we retain only the 200 highest scoring sentences per topic for further processing.

4.2 Estimating sentence importance

Given a list of sentences relevant for topic t, the next step is to estimate how important these sentences are for t. The importance score ($score_I$) is then combined with the relevance score ($score_R$) to produce a re-ranked list of sentences for the next step of the pipeline: removing duplicates.

Position-based evidence The first source of evidence of importance that we consider comes from Wikipedia's authoring guidelines and conventions: the earlier a sentence appears in an article, the more important it is assumed to be for the topic of the article. The position score for a sentence s is defined similarly to [23] as $pos(s) = (N_s - POS_s + 1) \cdot N_s^{-1}$, where N_s is the number of sentences in the article containing s, and POS_s is the position of s in the article.

Evidence from reference corpora This type of evidence for sentence importance is based on the assumption that there is a high degree of similarity among sentences describing similar entities in Wikipedia, where "similar" is defined as "belonging to the same Wikipedia category." First, even before the system is provided with actual topics, for every Wikipedia category c, we create a *reference corpus* C_c: this corpus will be used to estimate importance of sentences relevant for topics that belong to the category c.[4] In order to create a reference corpus for a category c, we take a random sample of 20 articles labeled with that category. The combination of these articles constitutes C_c. Given a topic t, we build a reference corpus C_t as the union of all reference corpora C_c such that the article t belongs to category c.

To define the importance score $score_I$ we will estimate an importance factor $\mu(s,t)$ (of sentence s for topic t) on the basis of a reference corpus. There are several ways of estimating this importance factor; below, we examine three methods of computing similarity of a sentence with the reference corpus: methods using word overlap and using language modeling, and a graph-based method.

Word overlap In this estimation method we assume that sentence s is important for topic t if it resembles a sentence from the reference corpus C_t. We compute the "resemblance" using word overlap. Given a topic t and a sentence s, the importance factor $\mu(s,t)$ is the maximum of the Jaccard coefficients between s and sentences in C_t.

Language modeling Similarly to the Word Overlap method, the factor $\mu(s,t)$ is determined by "resemblance" of s to the reference corpus, but now we use a language modeling approach. Given a topic t with reference corpus C_t, we take $\mu(s,t)$ to be the likelihood of the target sentence given the reference corpus. That is, $\mu(s,t) = p(s|C_t)$, where the probability is calculated using the maximum likelihood estimates for the words $w_1, \ldots, w_n$ in s:

$$p(s|C_t) = P(w_1, \ldots, w_n|C_t) = \prod_{i=1}^{n} p(w_i|C_t). \quad (4)$$

Eq. 4 depends on the length of the target sentence. To factor out the length, [19] proposed a length normalized generative probability estimate for ranking documents for the task of topic tracking, which in our case becomes:

$$\mu(s,t) = \sum_{w \in s} \big(\log p(w|s) \log p(w|C_t) - \log p(w|s) \log p(w|W) \big), \quad (5)$$

where $p(\cdot|s_i)$ is the model based on the target sentence s_i, $p(\cdot|C_c)$ is the model based on the relevance corpus, and $p(\cdot|W)$ is the model based on the background corpus: in our case, a language model based on the Wikipedia.

[4] Sentences in the reference corpus C_c will be called *reference* sentences, and sentences whose importance for a given topic we want to estimate will be referred to as *target* sentences.

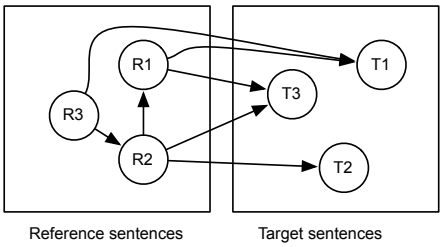

Figure 2. A graph for importance estimation.

In the models described above, the probabilities $p(w|C_t)$ and $p(w|s)$ are smoothed with a background model (the entire Wikipedia corpus), using

$$p(w|X) = \frac{n(w,X) + \beta p(w|W)}{|X| + \beta}, \quad (6)$$

where β is a smoothing parameter, X is C_t or s and $p(\cdot|W)$ is the background model. During experiments with development topics, we found that the length-normalized version (5) performs better than the standard language model. Therefore, we selected (5) for further experiments.

Graph-based importance estimation The use of a graph-based method for estimating the importance of a sentence is motivated by the intuition that importance is a *global* notion, and this is exactly what graph-based methods allow us to achieve. We create a graph with target and reference sentences as nodes and weighted edges indicating similarity between pairs of sentences. In this model, sentences receive evidence of their importance from other sentences and, in turn, they "pass on" their importance, in a recursive manner. Our graph-based ranking method extends a method proposed in [11, 22]: we bring in "weighted support" from the reference corpus. Specifically, we construct a graph structure of the type shown in Fig. 2. Reference sentences are sources of weighted directed links to target sentences; the weight attached to a link is the Jaccard similarity between the two sentences.

Once we have the graph configuration described above, the computation of the importance factor $\mu(s,t)$ of target sentences s is based on the weighted PageRank algorithm:

$$\mu(s,t) = PR(s,t) = \frac{d}{N} + (1-d) \sum_{u \in adj(s)} \frac{JC(s,u)}{\sum_{v \in adj(u)} JC(v,u)} PR(u,t), \quad (7)$$

where d is the PageRank 'damping factor' which is set to 0.85, N is the total number of nodes in the graph, u and v represent nodes in the graph, $JC(u,v)$ is the similarity (Jaccard coefficient) between the sentences u and v [5], and $adj(s)$ is the set of start-points of all incoming links of s.

Combining the sources of evidence In order to obtain a final estimation of the importance of a sentence s, we combine the position-based score and the score derived from the reference corpus: $score_I(s,t) = \mu(s,t) \cdot (pos(s) + 1)$, In the experiments described below we considered all three ways of estimating the importance factor $\mu(s,t)$: based on word overlap, language modeling and sentence graphs.

4.3 Removing redundant sentences

The final step of the processing pipeline (Fig. 4) consists of removing redundant sentences from the list of (at most) 200 sentences ranked by the combined relevance and importance $score_{RI}(s,t)$ (as defined in (1)). A target sentence may be redundant for two reasons: because it contains facts already mentioned in the topic article t, or because there are other target sentences that express the same fact.

In order to identify redundant sentences we sort the sentences by decreasing combined relevance and importance score. We compare each candidate sentence with the sentences in the topic article t, and with the retrieved sentences that appear higher in the ranked list. Again, we use the Jaccard coefficient to estimate similarity between pairs of sentences and define the redundancy score (see Eq. 3) of a candidate sentence as the maximum similarity. Finally, the redundancy score is subtracted from the combined sentence score and the list of sentences is sorted again in decreasing order of the resulting scores. The top 10 sentences comprise the output for topic t.

5. Evaluation

5.1. Experimental setup

The main goal of our experiments was to compare methods for estimating relevance and importance of sentences as part of the fact discovery task: which factors improve this estimation? We based our experiments on the infrastructure created for the CLEF 2006 WiQA pilot task [17].

Data We used the XML version of the English Wikipedia corpora made available by [9]. It contains 659,388 articles and has 2.28 categories per article, on average. The corpus contains annotations for common structural elements in Wikipedia such as article title, sections, paragraphs, sentences, hyperlinks, and templates. Additionally, the corpus annotated the semantic classes of the articles: PERSON, LOCATION, ORGANIZATION or OTHER. This rich annotation substantially simplifies the processing of the corpus. As a test set, the WiQA 2006 task organizers provided 65 topics (Wikipedia articles). We use this test set for reporting the results below.

Assessment Systems had to return a ranked list of sentences as a response to a topic. The top 10 sentences are then submitted for manual assessment. Each sentence in the list also provides the title of Wikipedia article from which it originates. The sentences are assessed using the criteria defined in the WiQA pilot: *support*, *relevance and importance*, *novelty*, and *non-repetition*. I.e., sentences should be relevant to the topic, contain new information with respect to the content of the topic article, be unique (non-repetitive), and come from Wikipedia articles other than the one containing the topic. The evaluation measures used are:

- Average yield per topic: the total number of supported, important, novel, and non-repeated snippets for all topics among the top 10 snippets, divided by the number of topics (*Avg. Yield*);
- MRR: Mean Reciprocal Rank of the first supported important novel non-repeated snippet among top 10, according to the system's ranking (*MRR*);
- Precision: the number of supported important novel non-repeated snippets among the top 10 snippets per topic, divided by the total number of top 10 snippets per topic (*Precision*).

We evaluated 7 variants of the system: *baseline* (vector space-based relevance and no importance estimation); *word overlap-based sentence importance* (either with link- or vector space-based relevance); *language modeling-based sentence importance* (either with link- or vector space-based relevance); and *graph-based sentence importance* (either with link- and vector space-based relevance).

5.2. Results

Table 1 shows the results of the comparisons of the 7 versions of our system. The two sentence retrieval approaches are indicated by *Ret* (retrieval only approach) and *Link* (link-based approach). The graph-based methods outperform the corresponding link-based methods. The baseline method which uses only the retrieval scores for ranking snippets performs better than the word overlap and language modeling approaches. Hence, the graph-based method makes effective use of the reference corpora—while the word overlap and language modelling approaches do not. The addition of the reference corpus yields a minor improvement in the MRR score for the word overlap method.

Method	Retrieval	Avg. yield	MRR	Precision
Baseline	Ret	2.046	0.391	0.226
Word over.	Ret	1.875	0.404	0.207
	Link	1.861	0.401	0.197
Lang mod.	Ret	2.031	0.399	0.224
	Link	1.769	0.286	0.187
Graph	Ret	2.938	0.523	0.329
	Link	**3.385**	**0.579**	**0.358**

Table 1. Comparing approaches.

The vector space-based retrieval method (Ret) results in better performance than link-based retrieval (Link) when it is used in combination with language modeling and word overlap methods. However, the combination of link-based retrieval and graph-based method achieves the best performance overall. The language model based version outperforms the word overlap version, but is outperformed by the baseline. The language model approach tends to favour longer snippets than the other methods.

The differences between the two graph-based methods on the one hand and the baseline on the other are significant.[5] The difference between the two graph-based approaches is also significant. The scores for the graph-based method (with link-based retrieval) were best or second-best at WiQA 2006 (English monolingual); the *Avg. Yield* was especially high in comparison with other submitted runs, outperforming them by at least 15% [17].

A closer look All the runs returned more than 5 good snippets for the following topics: *Center for American Progress*, *Atyrau*, *Saitama Prefecture*, *Kang Youwei*, *Philips Records*. All methods tend to return similar sets of good snippets. In contrast, all methods fail to return good snippets for the following topics: *Brooks Williams*, *Chemung County, New York*, *Wing Commander (film)*, *Christian County, Illinois*, *White nationalism*, *Telenovela database*, *Oxygen depletion*. For some of these topics, the retrieval component returned very few candidate snippets, e.g., *Brooks Williams* and *Oxygen depletion*. For others, it returned a large number of similar snippets, e.g., towns and cities for *Christian County, Illinois* and different entries of *Telenovela database*, which are judged irrelevant or redundant by the assessors. Some topics have ambiguous titles, e.g., *Wing Commander (film)*, as indicated by its result set which contains snippets from articles with similar titles, e.g., *Wing Commander (computer game)*.

6. Conclusion

We proposed search aids to harvest the material on a given topic t that is not contained in the Wikipedia article on t. This provides useful functionality to several types of users of Wikipedia. In our approach, we try to use the distinguishing properties of Wikipedia itself to address our task. Our main contribution is a set of algorithms for solving the task. A graph-based method that exploits the knowledge implicit in Wikipedia and that uses both content, layout- and link-based features proved to be the most effective in discovering facts that complement existing articles.

[5] With two-tailed Wilcoxon matched-pair signed-rank test, $p = 0.05$.

Our methods are generic and can be applied to other languages. We have applied the graph-based version of our method to the Dutch WiQA task and obtained good results. We also conducted preliminary experiments on the Dutch-English multilingual task. We added a module that enables computation of similarity across different languages. Here we made use of the cross-language link structure of Wikipedia to generate a bilingual lexicon, and used it to compute multilingual similarity scores; see [13].

Our methods use a limited amount of information, i.e., only the title of the topic, as a starting point in identifying important snippets. We explored different properties of the corpus, particularly the link structure and categories (classification information) in devising the methods. However, using only the title of the topic, the simple retrieval-based ranking tends to give better results than the other corpus-based reranking methods, particularly the language model and word overlap based methods. Future work will investigate ways to use the content of the target article for identifying relevant snippets, and explore different ways of creating reference corpora.

7. Acknowledgments

This research was supported by the Netherlands Organization for Scientific Research (NWO) by grants under project numbers 220-80-001 and 612.000.106.

References

[1] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach. Using Wikipedia at the TREC QA Track. In *TREC 2004*, 2004.

[2] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR '03*, pages 314–321, 2003.

[3] I. Androutsopoulos and D. Galanis. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In *HLT/EMNLP*, pages 323–330, 2005.

[4] R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT/NAACL*, pages 113–120, 2004.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[6] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98*, pages 335–336, 1998.

[7] A. Ciffolilli. Phantom authority, selfselective recruitment and retention of members in virtual communities: The case of Wikipedia. *First Monday*, 8(12), 2003.

[8] T. Clifton and W. Teahan. Bangor at trec 2004: Question answering track. In *TREC 2004*, 2004.

[9] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

[10] DUC. Document Understanding Conference, 2005. URL: `http://www-nlpir.nist.gov/projects/duc/`.

[11] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479, 2004.

[12] E. Filatova and J. Prager. Tell me what you do and i'll tell you what you are: Learning occupation-related activities for biographies. In *HLT/NAACL*, pages 49–56, 2004.

[13] S. Fissaha Adafre and M. de Rijke. Finding similar sentences across multiple languages in wikipedia. In *EACL 2006 Workshop on New Text*, 2006.

[14] M. Fleischman, E. Hovy, and A. Echihabi. Offline strategies for online question answering: answering questions before they are asked. In *ACL '03*, pages 1–7, 2003.

[15] W. Hildebrandt, B. Katz, and J. J. Lin. Answering definition questions with multiple knowledge sources. In *HLT/NAACL*, pages 49–56, 2004.

[16] INEX. Initiative for the evaluation of XML retrieval, 2006. URL: `http://inex.is.informatik.uni\-duisburg.de/2006/`.

[17] V. Jijkoun and M. de Rijke. Overview of WiQA 2006. In A. Nardi, C. Peters, and J. Vicedo, editors, *Working Notes CLEF 2006*, September 2006.

[18] V. Jijkoun, M. de Rijke, and J. Mur. Information extraction for question answering: Improving recall through syntactic patterns. In *COLING 2004*, 2004.

[19] W. Kraaij and M. Spitters. Language models for topic tracking. In B. Croft and J. Lafferty, editors, *Language Models for Information Retrieval*. Kluwer Academic Publishers, 2003.

[20] W. Kraaij, M. Spitters, and M. van der Heijden. Combining a mixture language model and Naive Bayes for multi-document summarisation. In *DUC 2001 workshop*, 2001.

[21] Lucene. The Lucene search engine, 2006. `http://lucene.apache.org/`.

[22] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *ACL 2004*, 2004.

[23] D. R. Radev, H. Jing, M. Sty, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, 2004.

[24] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *NLDB*, pages 67–79, 2005.

[25] B. Schiffman, I. Mani, and K. J. Concepcion. Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *ACL*, pages 450–457, 2001.

[26] I. Soboroff and D. Harman. Novelty Detection: The TREC Experience. In *HLT/EMNLP*, pages 105–112, 2005.

[27] W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[28] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *HLT/NAACL*, 2004.

[29] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. In *WWW '06*, pages 585–594, 2006.