Identifying document topics using the Wikipedia category network

Peter Schönhofen
Computer and Automation Research Institute
Hungarian Academy of Sciences
Kende u. 13–17, H-1111 Budapest
schonhofen@ilab.sztaki.hu

Abstract

In the last few years size and coverage of Wikipedia, a freely available on-line encyclopedia has reached the point where it can be utilized similar to an ontology or taxonomy to identify the topics discussed in a document. In this paper we will show that even a simple algorithm that exploits only the titles and categories of Wikipedia articles can characterize documents by Wikipedia categories surprisingly well. We test the reliability of our method by predicting categories of Wikipedia articles themselves based on their bodies, and by performing classification and clustering on 20 Newsgroups and RCV1, representing documents by their Wikipedia categories instead of their texts.

1. Introduction

The goal of topic identification is to find labels or categories (ideally selected from a fixed set) best describing the contents of documents, which then can aid various information retrieval tasks, such as classification and clustering. One possible approach is to utilize an ontology to detect concepts in the document and select the most dominant of them. Unfortunately, the majority of existing ontologies or taxonomies are either too small and does not contain domain specific information (OpenCyc with 47,000 concepts, WordNet with 120,000 synsets vs. 800,000 in Wikipedia, see also [15, 7]), or they do not organize concepts strictly by semantic relationship (e.g. ODP frequently subdivides categories according to the types of web sites they appear on, see [6, 8]). On the other hand, coverage of Wikipedia is general purpose and very wide, containing up to date information about persons, products, technologies etc., making it a much better option, despite the fact that its structure is far less rigorous, rich and consistent than that of ontologies.

Wikipedia consists of articles, and separate pages for images, discussions about article contents, authors, page component templates and so on. Articles have titles, assigned categories, and often refer to other articles. Some articles are accessible through more than one titles; in this case, additional titles are implemented as special articles, or redirections, containing only a single link to the main article. Categories are organized hierarchically into sub-categories and super-categories. Note that the hierarchy is not a tree, some categories have multiple super-categories.

We present a relatively simple method which, by exploiting only the titles and categories of Wikipedia articles, can effectively determine the Wikipedia categories most characteristic of a document. First, we identify all Wikipedia articles possibly related to the document by matching their titles with words of the document. Articles are then weighted by three kinds of factors, concerning

- words which are shared between the document and the article title, such as their frequency, or the number of Wikipedia categories in which they appear;
- strength of the match between the document and the article, like the number of words in common, or percentage of the title words which are present in the document;
- the article itself, for example the number of Wikipedia articles with very similar titles.

Second, it collects categories assigned to these articles, establishing a ranking between them based mainly on the weights of the articles promoting them, but also taking into account how many document words support them (through the articles) and to how much degree is this support shared with other (stronger) categories. We emphasize that we do not exploit the full potential of Wikipedia: we do not use the information contained in the actual text of articles, the links between articles, or the hierarchy of categories, which will be the focus of our planned future research.

We validate our method by carrying out two kinds of experiments. First, we predict categories of the Wikipedia articles themselves, and observe that for 86% of articles, the top ranked 20 categories contain at least one of the original ones, with the top ranked category correct for 48% of articles. Second, we perform classification and clustering on the documents of 20 Newsgroups and RCV1, representing documents by their Wikipedia categories. We find that for RCV1, classification based solely on Wikipedia categories performs equal to using the full text of the documents. If in addition to Wikipedia categories we also use the top 20 $tf \times idf$ words of the document, we obtain much better accuracy for classification and roughly equivalent for clustering compared to using the full text, both for the RCV1 and the 20 Newsgroups corpora.

There are several approaches for topic identification using a fixed set of labels contained in an ontology or taxonomy such as matching important words of a document against Yahoo directory titles [20]; finding WordNet concepts in the text and estimating their importance based on how many times they or their related concepts occur [11]; comparing the language model of documents with those of Yahoo or Google directories [2]; finding the WordNet concept most similar to the document, where similarity is measured between weighted word vectors [16]; assigning ontology nodes to document clusters [19], to name a few examples. For an overview, see [12]. Although some of the proposed methods and in particular those of [11] and [20] are similar to ours, their computation of label weights and handling of the ontology structure is significantly different.

Wikipedia received the attention of the information retrieval community only recently. Several papers describe ways to modify its structure for making it more suitable for machine processing [21], analyze its organization [5, 22], extend its content [1], using it to add new relationships to WordNet [17], or utilize it for various information retrieval tasks, such as question answering [14, 3]; however, to our best knowledge, no attempt were made so far to apply it for either topic identification or document extraction.

2. Proposed method

The goal of our method is to find the Wikipedia categories most characteristic to a given document. To achieve this, it collects all Wikipedia articles suggested by the words present in the document, then determines which Wikipedia categories are the most dominant among these articles.

2.1. Preparing the Wikipedia corpus

The Wikipedia corpus in its original form, either as a set of HTML pages provided by a Wikipedia server, or as a large downloadable XML file containing pages written in Wiki markup, is not suitable directly for our purposes. Before we can work with it effectively, it has to undergo some preparations shown in Figure 1.

```
1: Reduce corpus to articles and redirections
2: Perform stop word removal and stemming on article titles, unite article titles if necessary
3: Remove categories corresponding to Wikipedia administration and maintenance
4: Remove categories containing less than 5 or more than 5000 documents
5: Merge stub categories with regular ones
```

Figure 1. Preparing the Wikipedia corpus.

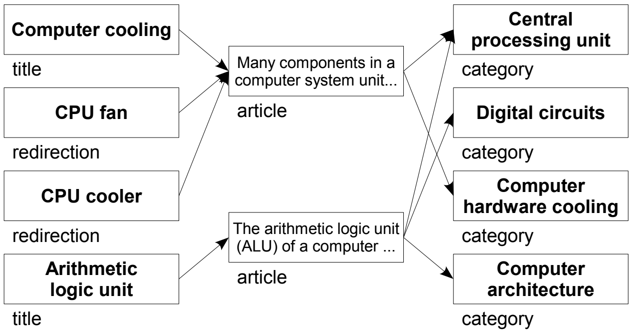

Figure 2. Simplified Wikipedia structure.

In step 1, we discard all special pages (categories, images, talks etc.), and keep only article and redirection pages; from articles, we retain only their titles and the list of assigned category names. For each article we generate an abstract object, in effect a simple ID, to which we link titles, redirections and categories as shown in Figure 2. This way from now on redirections can be handled as if they were additional article titles.

In step 2, to make recognition of Wikipedia terms easier in the documents, we remove stopwords and stem titles. As a consequence it can happen that two or more titles pointing to different articles are mapped to the same word sequence. In this case, the titles are united, and the new entity will point to all of the articles. Finally, a word index is made on the titles, yielding the structure illustrated in Figure 3.

Note that a word can point to many titles (such as "star"),

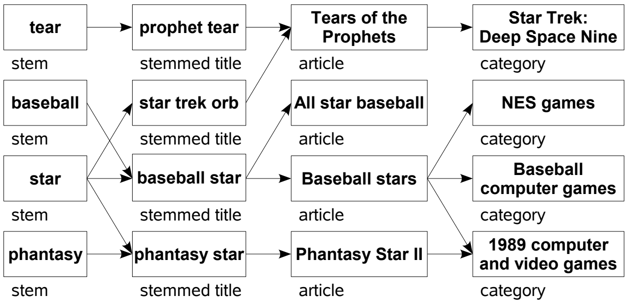

Figure 3. Prepared Wikipedia corpus.

```
1: Remove stop words and perform stemming, remove words not occurring in Wikipedia article titles
2: Collect words of the document and weight them by $R_w = tf_w \times \log \frac{N}{cf_w}$
3: Collect Wikipedia titles whose words (with the possible exception of one) are all present in the document, and weight them by $R_t = \sum_{w \to t} R_w \times \frac{1}{t_w} \times \frac{1}{a_t} \times \frac{S_t}{L_t}$
4: Collect Wikipedia articles pointed to by the titles and weight them by $R_a = \max_{t \to a} R_t$
5: Collect Wikipedia categories assigned to the articles and weight them by $R_c = \frac{v_c}{d_c} \times \sum_{a \to c} R_a$
6: Decrease weights of categories sharing its supporting words with other categories of higher $R_c$ values
7: Select categories with the highest weights
```

Figure 4. Identifying topics of a document.

the same title can point to multiple articles (like "baseball stars"), and also that more than one title may point to the same article (e.g. "Tears of the Prophets").

In addition, in steps 3–4 three minor corrections should be made to remove useless information which would confuse or slow down topic identification. First, "administrative" categories grouping articles by some operational property instead of their meaning, like "Wikipedia maintenance" or "Protected against vandalism", are deleted. Second, categories with too few (less than 5) or many (more than 5000) articles are also removed. Third, "stub" categories containing unfinished articles are merged with their regular counterpart, e.g. "Economics stub" with "Economics"; this has no effect on our algorithm as we only use article titles.

2.2. Identifying document topics

After we have prepared the Wikipedia corpus, everything is ready for the actual processing shown in Figure 4. Before proceeding, let us define a few simple terms. Category c is *assigned* to article a, or c is one of the *official categories* of a, if according to Wikipedia, a belongs to c. Word w *points* to article title t, if it occurs in it; likewise, title t points to article a if it is one of the titles of a. Finally, the set of words occurring in the titles of articles in category c will be called the *vocabulary* of c.

In step 1, we performing stopword removal and stemming on the source document, in exactly the same way as we did during the preparation of the Wikipedia corpus, thus aligning the vocabularies on both sides. Words of the documents not present in any Wikipedia article titles are ignored.

In step 2, we assign an R_w weight to each word w:

$$R_w = tf_w \times \log \frac{N}{cf_w} \tag{1}$$

where tf is term frequency (number of times the word occurs in the document); N is the number of Wikipedia categories; finally, cf_w is the category frequency, specifying how many categories contain word w in their vocabulary. The second factor is inverse category frequency, icf_w defined over category vocabularies similar to inverse document frequency. Note that some papers define inverse category frequency differently, as they count the number of corpus categories and not Wikipedia categories the given word occurs in.

In formula (1), the first factor emphasizes words occurring many times in the document, thus probably being central to the document topic. The second factor gives preference to words which select only a few categories, and therefore do not introduce significant uncertainty and noise into the later analysis steps. We do not utilize the idf measurement, because our goal is to determine categories that best describe a document, not those that are the most advantageous for the classification, clustering or other information retrieval algorithm run on the document corpus.

In step 3, we collect Wikipedia titles supported by words present in the document. Word w *supports* title t if (1) w occurs in t, and (2) out of the other M words of t, at least $M - 1$ are present in the document. Of course if the title contains only a single word, the second condition should be ignored.

Notice that in step 3 we allow a single word mismatch between the title and the document to properly handle documents that refer to persons, places or technical terms in an incomplete way, for example "Boris Yeltsin" may appear as "Yeltsin" or "Paris, France" as "Paris". In addition, Wikipedia titles occasionally contain auxiliary description between parentheses or after a comma, to make clear that of the multiple possible senses of the concept, exactly which one is discussed in the corresponding article, like in "Bond (finance)" and "Bond (masonry)". This auxiliary information does not necessarily appear directly in the document, because it is evident from the context or the document uses another word for disambiguation.

Similarly to words, titles are also weighted in step 3:

$$R_t = \sum_{w \to t} R_w \times \frac{1}{t_w} \times \frac{1}{a_t} \times \frac{S_t}{L_t} \tag{2}$$

where R_w is the weight of a supporting word, as defined in the previous step; t_w denotes the number of Wikipedia titles containing word w; a_t is the number of articles pointed to by title t. Finally, L_t stands for the title length, in words, and S_t specifies how many of the title words are present in the document. Although the second factor could have been computed as part of R_w, since it does not depend on the title, we feel that due to its meaning it belongs rather to R_t.

Through the first factor of (2) titles become preferred or suppressed based on the importance of their supporting words. The last factor simply measures how much percentage of the title words occur actually in the document text, giving less emphasis to only partially supported titles. The reason for strengthening articles with longer titles is quite straightforward: the probability that they were detected by mistake is lower. For instance, it is practically impossible that a document containing the words "comprehensive", "test", "ban" and "treaty" does not discuss arms control.

The purpose of the second and third factor of (2) is to prevent common words pointing to many titles, and titles pointing to many articles from gaining undeservedly high weights during further analysis. Unfortunately, Wikipedia does not discuss every topics in the same detail, for instance, there are a lot more articles about musical albums than the domain of photography. Therefore if a document contains words "album" and "photo", the former will attract many times more Wikipedia articles than the latter, heavily distorting the category distribution. Similarly, as a consequence of stemming, there are titles referring to a large amount of articles, for instance every "Architecture in X", where X is a year number, will become "architecture". Since these articles are about the same topic, and thus usually vote for the same set of categories, without the balancing effect of the third factor, they would easily overwhelm other equally important concepts.

In step 4, we collect articles pointed to by the titles found in the previous step. If the same article are pointed to by different titles (because of redirections), its weight will be simply the maximum of theirs:

$$R_a = \max_{t \to a} R_t \tag{3}$$

Note that we do not add the weights up as the number of titles for an article reflects the structure of Wikipedia, and not the importance of the article.

In step 5, we make a list of the categories assigned to the collected articles, and we weight each one by the sum of the corresponding articles (note than an article might vote for several categories):

$$R_c = \sum_{a \to c} R_a \tag{4}$$

In step 6, we simply select H categories with the highest weight; they will be considered the most characteristic topic(s) of the document content.

2.3. Improvements

By introducing two small modifications to the method described in the previous section we can greatly improve its accuracy; each modification affect only step 5, that is, computation of the R_c category weights. In order to make their explanations easier to follow, let us define the *supporting words* of category c as the set of words supporting articles which pointed to c.

In the first modification, we attempt to suppress categories whose high R_c value is only a consequence of their exceptionally large vocabulary, such as for "Actors" and "Films". This can be regarded as an extension of the effort realized in the second and third factor in formula (2). The modification is realized as an extension of formula (4):

$$R_c = \frac{v_c}{d_c} \times \sum_{a \to c} R_a \tag{5}$$

v_c denotes the number of supporting words of category c; d_c is the number of words in the vocabulary of category c.

With the second modification we prevent that words already "consumed" or "accounted for" by a more significant category promote other less important ones. For example, if "ban" already supported the concept "comprehensive test ban treaty", then it would be obviously a mistake to allow it to strengthen also "Ban (mythology)" with the same degree.

The second modification represents an additional step after step 5 where we have collected the categories and computed their weights. First, we set up a d_w *decay value*, initially 1, for each word of the document. Next, we sort the categories according to their weight, and go through them, starting with those of the highest weight. For each one, we recompute its weight, then recompute the decay values for its B_c set of supporting words:

$$R_c^{'} = R_c \times \frac{\sum_{w \in B_c} d_w}{|B_c|} \tag{6}$$

$$d_w^{'} = \frac{d_w}{2}, w \in B_c. \tag{7}$$

That is, R_c is multiplied by the average decay value of the supporting words of category c, whose decay value will be halved. If none of the supporting words are shared with previously examined (higher weighted) categories, R_c will remain intact.

3. Experiments

To validate our approach for topic identification, we performed two groups of experiments. In the first group, we measured how well our proposed algorithm is able to predict the categories of the Wikipedia articles themselves, based solely on their bodies (which our method did not know). In the second, we assigned Wikipedia categories to the documents of two well known corpora, 20 Newsgroups and RCV1 [10], and then examined how well they represented the documents during classification and clustering.

For our experiments we used the Wikipedia snapshot taken at 19th February 2006, containing 878,710 articles, 1,103,777 redirections, 55,298 categories. Stemming was carried out by TreeTagger [18], for stopwords we used a slightly modified version of the list compiled for the Smart

Table 1. Top 10 Wikipedia categories selected for the article “Analysis of variance”

R_c	Category
1.00	Statistics
0.30	Evaluation methods
0.17	Scientific modeling
0.13	1930s comics
0.10	Probability theory
0.08	Observation
0.07	Social sciences methodology
0.07	Data modeling
0.07	Probability and statistics
0.05	History of boxing

search engine. We deleted numbers from Wikipedia titles, but not tokens containing both digits and letters, like “1930s”, “win95”.

In order to simplify further discussions, let us denote the n categories having the highest R_c weights among the categories collected for a specific document as the *top n categories* (in fact, n is equal to H mentioned at step 6).

3.1. Predicting categories of Wikipedia articles

In the first group of experiments, we ran our algorithm on approximately 30,000 articles randomly selected from Wikipedia, in order to measure how well it can predict their original categories. Note that the prior knowledge of our algorithm about the Wikipedia corpus (article titles, categories) did not overlap with the processed documents (article bodies), so there was no danger of the evaluation results being distorted.

As an example of selected Wikipedia categories with their R_c weights, Table 1 shows the 10 categories deemed as most relevant for the Wikipedia article #1566, “Analysis of variance”, to which officially only a single category, “Statistics” was assigned. Weights are normalized to 1. As can be seen, the single official category received more than three times the weight of the second category on the list, and except from “1930s comics” and “History of boxing”, which were supported by words “1920s” and “1930s”, all proposed categories have a strong relationship with the main topic. In fact, “Probability and statistics” is a super-category of “Statistics”, and in turn, “Probability theory” is a super-category of “Probability and statistics”.

However obvious it may seem, there are several problems with measuring accuracy on the Wikipedia corpus and by the number of official categories present among the top 20 categories:

- Wikipedia articles are more cleanly written than the average documents in the World Wide Web or other electronic document repositories;
- the categorization of Wikipedia articles is not always consistent, for example article “Politics of Estonia” is not linked to category “Estonia”;
- density of the Wikipedia category net is very uneven, some topics are more detailed than others;
- some Wikipedia articles combine semantically unrelated concepts, usually based on the fact that they are chronologically connected (like “April 7”) or their abbreviation is the same (e.g. “CD (disambiguation)”);
- similarly, many Wikipedia categories cover semantically unrelated articles, such as “1924 deaths”.

The first point can be easily resolved by testing our method on other corpora (as we will do in the next section), the others, however, would require extensive human intervention.

The curve “exact match” on Figure 5 shows the number of documents for which the top 20 categories contained at least a specific percentage (indicated on the x axis) of the official categories. We see that the proposed method predicted at least one official category for approximately 86% of the examined documents, and there were about 40% for which all official category was discovered. The sharp drop near 50% can be attributed to the disproportionally large number of documents having exactly two official categories assigned, as illustrated in Figure 6.

If we do not insist that official categories appear directly among the top 20 categories, and allow their substitution by their immediate sub- or super-categories, according to curve “maxdist=1”, accuracy nearly uniformly improves by approximately 4–8%. If we further relax our requirements, and instead of one level of indirection, we allow two, accuracy again increases, as shown by the curve labeled “distmax=2”. The improvement is larger this time, which is no surprise, since n levels of indirection mean roughly a^n possible substitutions, where a is the average number of immediate sub- and super-categories for a category.

Another approach to estimate quality of category selection is to adopt the precision/recall measurement widely used for information retrieval [4]. For our case, precision means that out of the top n categories, how much percentage is actually an official category (or is a sub- or super-category of an official one, if we allow indirect matching); while recall specifies how much percentage of the official categories occurs among the top n categories. Note that if n is smaller than the number of official categories, recall assumes that there are only n official categories; otherwise it would indicate an undeservedly low accuracy.

Figure 7 shows the two measurements for different n values, the strange drop in the latter can be attributed to the

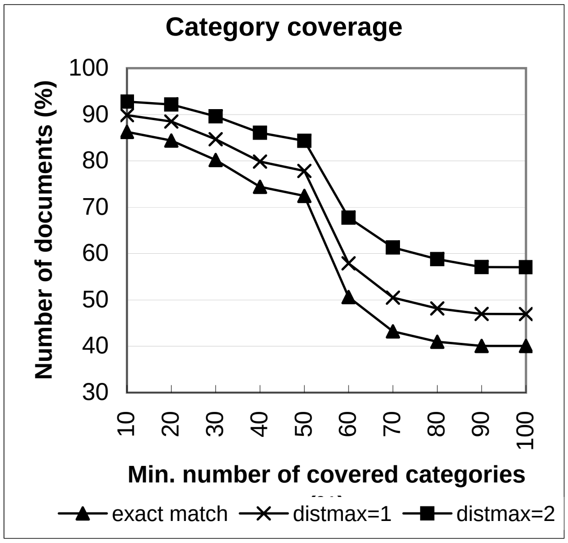

Figure 5. Amount of Wikipedia articles for which at least a given percentage of official Wikipedia categories was present in the top 20 categories. The “distmax=n” curves represent the case when instead of the official category we also accept one of its sub- or super-categories, assuming the level of indirection does not exceed n.

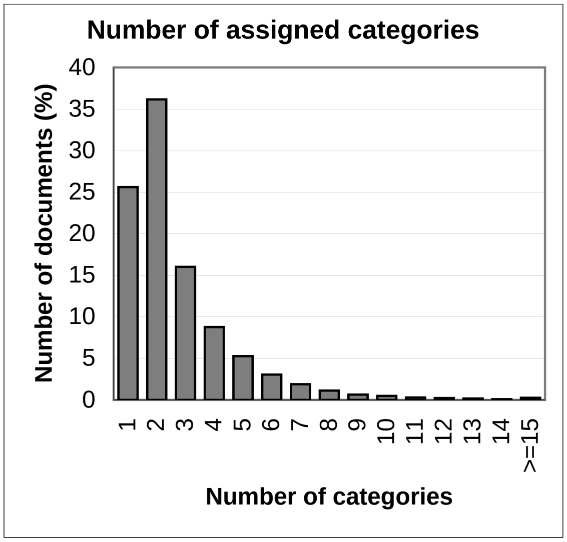

Figure 6. Amount of Wikipedia articles with a given number of official categories.

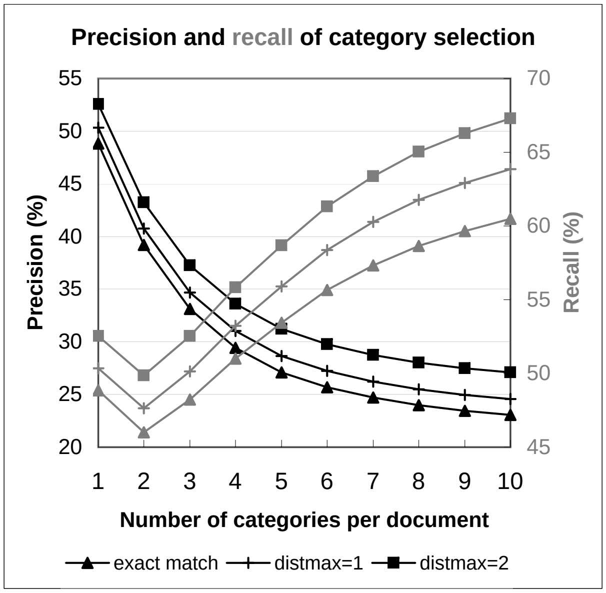

Figure 7. Black: percentage of Wikipedia categories which were correct in the top n categories. Gray: percentage of official Wikipedia categories among the top n categories. In both cases, n is shown on the x axis, values are averaged over processed documents.

special behavior of recall when n is lower than the number of official categories. Allowing indirect category matching has the same effect as we saw in earlier Figure 5, we observe a slightly stronger improvement between “distmax=1” and “distmax=2” than between “distmax=1” and “exact match”.

3.2. Classification and clustering

To test our method on real-world documents, we ran it on both the 20,000 postings of 20 Newsgroups and the first 200,000 news articles of RCV1. Since these documents were categorized according to their own schemas, we could not directly evaluate accuracy of the top 20 Wikipedia categories. Instead, we examined how well the top 20 categories represented documents during classification and clustering.

Of course, stemming and stopword removal was performed in exactly the same way on documents as on Wikipedia titles. In addition, from 20 Newsgroups postings, we removed e-mail and web addresses, since they do not match with any Wikipedia title. In RCV1, we regarded topic codes as categories; if a document had more than one, we kept only the most specific. For classification we used the naive Bayes algorithm of the Bow toolkit [13], and for clustering the CLUTO package [9], both with default parameters.

We compared the performance of four document repre-

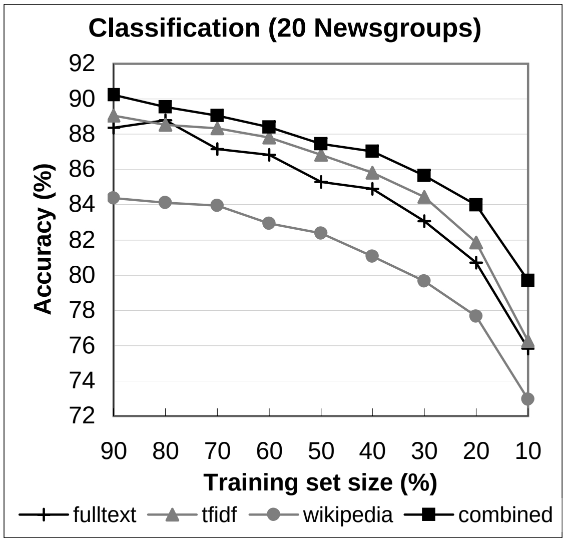

Figure 8. Classification accuracy in 20 Newsgroups at various training set sizes, when documents were represented by full text ("fulltext"), the 20 words with highest $tf \times idf$ ("tfidf"), top 20 categories ("wikipedia"), or combination of the latter two ("combined").

sentation technique: (1) documents represented by their full text, shown as "fulltext" on diagrams; (2) by the 20 words with the highest $tf \times idf$ values, "tfidf"; (3) by the top 20 Wikipedia categories, "wikipedia"; and finally (4) by merging (2) and (4), "combined".

Classification accuracy (the percentage of correctly classified documents in the test set), is illustrated on Figures 8 and 9. As we can see, for 20 Newsgroups Wikipedia categories used in themselves have a much lower accuracy that can be obtained by the full text, because classification exploits names and signatures of posting authors. However, when merging specific ("tfidf") and general ("wikipedia") features, accuracy improves significantly, especially at lower training set sizes. For RCV1, Wikipedia categories are roughly as good as the full text, and when augmented by the top $tf \times idf$ words, similarly to 20 Newsgroups they surpass "tfidf", although to a lesser degree.

Clustering quality was measured by entropy and purity, whose description can be found in [9], observed values for various cluster numbers are shown in Figures 10–11. Using Wikipedia categories alone produces the worst result, as co-occurrence between categories is not as pervasive as between words, and thus they cannot express similarity between documents so effectively. However, when utilized together with top $tf \times idf$ words, their performance approaches, in some cases even exceeds that of the full text.

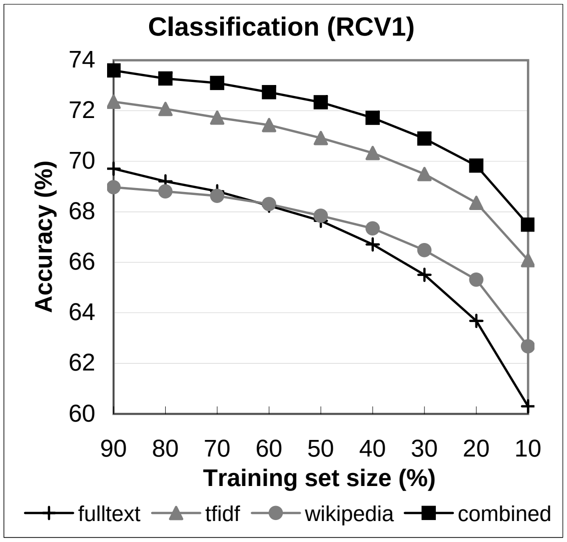

Figure 9. Accuracy of classification on documents of RCV1 at various training set sizes.

4. Conclusion and future plans

We presented a novel approach for determining the most characteristic categories of documents by relying on the Wikipedia on-line encyclopedia. We validated our method by several ways, first by predicting the categories for the Wikipedia articles themselves, then by classifying and clustering documents of 20 Newsgroups, RCV1 based on their Wikipedia categories. We observed that the Wikipedia categories, especially when augmented by words with the highest $tf \times idf$ values, represented documents as good as, and sometimes even better than their full text.

Our method used solely the titles and categories of Wikipedia articles, it did not even tried to exploit the rich information provided by the article texts, the links between articles, or the hierarchical structure of the categories, on which our future research will focus.

References

[1] S. F. Adafre and M. de Rijke. Discovering missing links in Wikipedia. In *Proc. of the 3rd int'l workshop on Link discovery*, pages 90–97, 2005.

[2] M. Aery, N. Ramamurthy, and Y. A. Aslandogan. Topic identification of textual data. Technical Report CSE-2003-25, University of Texas at Arlington, Department of Computer Science and Engineering, 2003.

[3] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, and S. Schlobach. Using Wikipedia at the TREC QA track. In *Proc. of the 13rd Text Retrieval Conf. (TREC)*, 2004.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

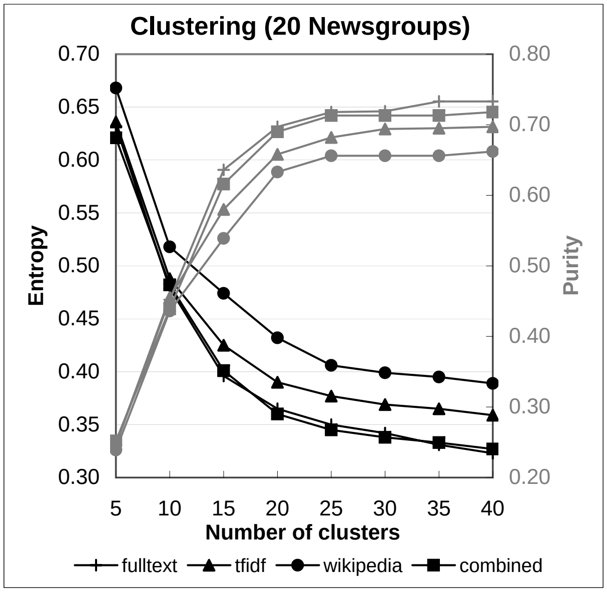

Figure 10. Entropy and purity of clusters in 20 Newsgroups at various cluster numbers.

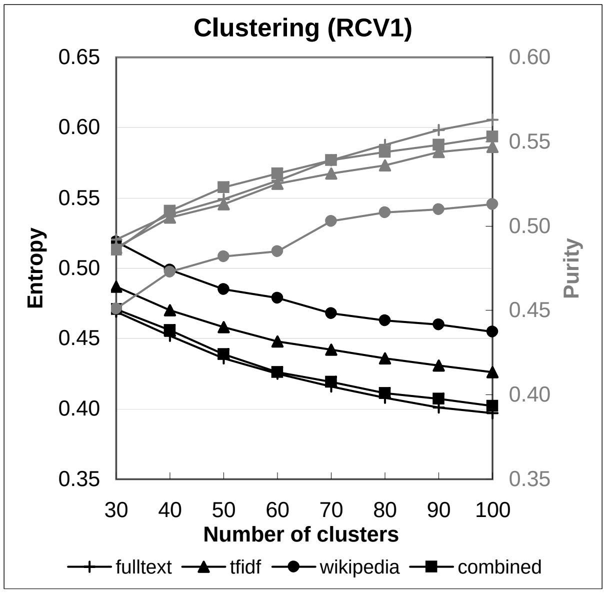

Figure 11. Entropy and purity of clusters in RCV1 at various cluster numbers.

[5] F. Bellomi and R. Bonato. Network analysis for Wikipedia. In *Proc. of Wikimania 2005, the 1st Int'l Wikimedia Conf.*, 2005.

[6] N. Cannata, E. Merelli, and R. B. Altman. Time to organize the bioinformatics resourceome. *PLoS Computational Biology*, 1(7), 2005.

[7] D. Fossati, G. Ghidoni, B. D. Eugenio, I. Cruz, H. Xiao, and R. Subba. The problem of ontology alignment on the web: a first report. In *Proc. of the 11th conf. of the European Association of Computational Linguistics, Workshop on Web as Corpus*, 2006.

[8] A. Gilbert, M. Gordon, M. Paprzycki, and J. Wright. The world of travel: a comparative analysis of classification methods. *Annales UMCS Informatica*, 2003.

[9] G. Karypis. CLUTO: A clustering toolkit, release 2.1. Technical Report 02-017, University of Minnesota, Department of Computer Science, 2002.

[10] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[11] C.-Y. Lin. Knowledge-based automatic topic identification. In *Meeting of the Association for Computational Linguistics*, pages 308–310, 1995.

[12] C.-Y. Lin. *Robust automated topic identification*. PhD thesis, University of Southern California, 1997.

[13] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/˜mccallum/bow, 1996.

[14] G. Mishne, M. de Rijke, and V. Jijkoun. Using a reference corpus as a user model for focused information retrieval. *J. of Digital Information Management*, 3(1):47–52, 2005.

[15] R. Navigli. Automatically extending, pruning and trimming general purpose ontologies. In *Proc. of the 2nd IEEE Int'l Conf. on Systems, Man and Cybernetics*, 2002.

[16] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic assignment of Wikipedia encyclopedic entries to wordnet synsets. In *Proc. of the 3rd Int'l Atlantic Web Intelligence Conf. (AWIC)*, pages 380–386, 2005.

[17] M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from Wikipedia. In *Proc. of the 10th Int'l Conf. on Applications of Natural Language to Information Systems (NLDB)*, pages 67–79, 2005.

[18] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of the Int'l Conf. on New Methods in Language Processing*, Manchester, UK, 1994.

[19] B. Stein and S. M. zu Eien. Topic identification: Framework and application. In *Proc. of the 4th Int'l Conf. on Knowledge Management (I-KNOW 04)*, pages 353–360, 2004.

[20] S. Tiun, R. Abdullah, and T. E. Kong. Automatic topic identification using ontology hierarchy. In *Proc. of the 2nd Int'l Conf. on Computational Linguistics and Intelligent Text Processing*, pages 444–453, London, UK, 2001.

[21] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. of the 15th int'l conf. on World Wide Web*. WWW2006, 2006.

[22] J. Voss. Measuring Wikipedia. In *Proc. of the Int'l Conf. of the Int'l Society for Scientometrics and Informetrics*, Stockholm, Sweden, 2005.