From ambiguous words to key-concept extraction

Márius Šajgalík, Michal Barla, Mária Bieliková
Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova, 842 16 Bratislava, Slovakia
Email: fsajgalik, barla, bielik@fiit.stuba.sk

***Abstract*—Automatic acquisition of keywords for given document is still an area of active research. In this paper, we consider shift from keyword-based representation to other perspective on representation of document's focus in form of key-concepts. The advantage of using concepts over simple words is that concepts, apart from words, are unambiguous. This leads to better understanding of key-concepts than keywords. We present novel method of key-concept extraction, which provides an efficient way of automatic acquisition of key-concepts in machine processing. We evaluate our approach on classification problem, where we compare it to baseline TF-IDF keyword model and present its competitive results. We discuss its potential of its utilisation on the Web.**

I. INTRODUCTION

In this modern information age, we are overflowed with huge amounts of data. The Indexed Web is estimated to contain over 14:35 billion web pages1 and is still growing. We strive to develop efficient ways of organising the data. It is also one of the goals of Semantic Web to provide such semantic metadata for web pages, which would help the machines to understand the semantics of the web page focus. There have been multiple initiatives like Microformats, Microdata, or RDF. However, there is still not enough explicit semantic information of sufficient quality included in the web page content2, which often forces us to incorporate some kind of ontology to understand the content of the "wild web" better [6].

Most commonly used are still the "old classic" keywords and description metatags, the former representing traditional keywords most relevant for given web page and the later some short natural language text describing the web page content. These metatags provide a way by which computers can categorise the content of web pages. Keyword representation of documents is rather old and still widely used [8], [13], [22].

In our work, we take another approach to automatically extract some metadata from raw text. We do not focus on extracting keywords, we extract key-concepts instead. The use of key-concepts as the most relevant concepts is advantageous in that meaning of each concept is very well-defined in an exact and clear way. This implies another advantage over keywords, which is greater information content of concept vector than keyword vector. This claim is based on work of [18], which shows that concept vector representation has greater information content than simple words or TF-IDF vector. To delve a little deeper, the computation of information content of vector representation is based on mutual information between vector items (in our case concepts) and documents, which has been shown to have positive impact on performance in multiple information retrieval tasks [14].

To obtain concepts from plain text, we utilise WordNet [16], which can be considered as a form of a lightweight ontology. It should be clarified that in fact, not all WordNet synsets are concepts. Some of them represent instances of concepts. Author of [4] proposes a method how to distinguish concepts from their instances in WordNet hierarchy. Although outdated in present, since starting from version 2:1 WordNet differentiate hypernym (hyponym) and instance hypernym (instance hyponym) relations [17], it points out rather intuitive observation that vertices corresponding to concept instances always lie on the bottom of the hierarchy. In our approach, we do not consider instance hypernym relations, thus cutting off concept instances, which results in acquisition of pure concepts.

In this paper, we present the notion of key-concepts and propose novel method of discovering them in text. In evaluation, we focus on classification problem. As we already mentioned, an efficient classification is very important in today's world of big data. Via classification, we demonstrate the main power of our extracted key-concepts, which is a substantial dimensionality reduction of document's feature space, while achieving greater information content than simple keywords.

II. RELATED WORKS

To be able to extract some concepts, we need to disambiguate the meaning of words. There are multiple methods for word sense disambiguation [1], [11], [12]. Some of them also use WordNet [2], [5] to infer the most probable word sense. Approach in [18] is little different from others, since it does not rely on making hard decisions, but ranks the WordNet synsets by relevance to the text (soft sense disambiguation). Its authors describe the representation of text by WordNet synsets instead of words. They point out two major drawbacks of "bag of words" representation polysemy and synonymy. Polysemy cause the ambiguity of the words since a single word can have multiple meanings. In case of synonymy, several words can have the same meaning and bag of words just lack the information about such relations among them. On the other hand, in synset representation, each synset has unique and clear meaning. Evaluation in [18] compares several different approaches to rank synsets in order to infer the most probable meaning and shows that the best results are achieved by using the PageRank algorithm.

Use of PageRank algorithm [9] in text mining has been researched by multiple researchers [23], [10], [7]. One of the pioneer methods is TextRank [15], which is an unsupervised algorithm for keyword extraction. Its promising results even caused its authors to apply for a patent on it. Another example of utilising PageRank is in [3], where it is also used to word sense disambiguation.

III. KEY-CONCEPT EXTRACTION

Our method is based on disambiguating the word senses using PageRank algorithm. The principal idea of our approach is to infer the relevance of latent concepts hidden in text by observing words. To infer the relevance, we take into account several factors like collocations of words, hypernym and holonym relations and information content of concepts.

To preprocess the text, we perform part-of-speech (POS) tagging3 to choose only the nouns (including compound nouns) as candidates. With these feasible terms extracted, we take all the noun synsets of WordNet, which contain at least one of these feasible terms. We call these synsets the basis synsets. Then we create the concept graph G = (V;E), where vertices V are all the basis synsets plus those reachable by following hypernym or holonym relations. This aims to influence also the more general concepts (WordNet synsets) to get to the broader topics discussed in the extracted article. We do not consider instance hypernym relations, since without it we observed better results. This makes our extracted synsets proper concepts, since concept instances are not propagated. These hypernym and holonym relations constitute the graph's edges E.

After we have built the concept graph, we perform PageRank algorithm to infer the relevance of individual concepts. Since the concept graph is undirected and weighted, we adapted the combination of PageRank modifications for both undirected and weighted graphs presented in [15] (see Equation 1).

The principle of our proposed approach is to do a two-pass ranking. In the first pass, we propagate the authority of a synset to all hypernyms and holonyms via existing edges to obtain the most probable word senses. In the second pass, we enrich the concept graph with edges representing the collocation relations too (besides the hierarchical links). That is, after performing the page ranking in the first pass, we link also the synsets that contain some neighbouring terms in the second pass as well to support the collocated word senses and thus, get the key concepts. We adapted this idea from TextRank [15], which is an unsupervised method to extract keywords. Such propagation of collocation relations can be seen as if terms (synsets) were voting for their neighbours. The inference is done iteratively using formula in Equation 1 to compute a new vertex score.

ROVNICA CISLO 1

There, V S(v) denotes the vertex score of vertex v, Adj(v) denotes the set of all adjacent vertices to v. Edge between vertices Vi and Vj is weighted with value wij and d is the damping factor usually set to 0:85 [9], [15]. We note that this equation variant differs from that presented in [15], where its authors consider only binary valued edge weights. We use a little more general formula, which allows us to tweak these values a little, since in our case, we have multiple possible relations between concepts (hypernyms, holonyms), not just word collocations. We observe more favourable results with higher weights assigned to hypernym-hyponym relations, although we did not conduct any experiments to support this statement. For reference, we set edge weights to 1 for hypernym-hyponym relations and 0:7 for holonym-meronym relations. It should be noted that SemanticRank [21] is based exactly on manipulating these weights to extract keywords and key-sentences from text.

After each run of PageRank algorithm, we do not take just the vertex values of graph. We multiply the vertex score obtained from both runs of PageRank by the information content of the corresponding concept to get the final ranking.

A. Notion of information content

We consider the information content [19] of concepts to account for different commonness of different concepts. The information content is a measure of specificity for a concept. The higher value of information content, the more specific is the concept (e.g. violin), whereas the lower values signify more general concepts (e.g. object). The information content IC(c) of a concept c is defined as the negative logarithm of the probability of this concept:

ROVNICA CISLO 2

The probability of a concept P(c) is computed as relative frequency of it:

ROVNICA CISLO 3

In general, N is the total number of nouns observed in some text corpus and freq(c) denotes the concept frequency:

ROVNICA CISLO 4

There, words(c) is the set of words assigned to the WordNet concept c and count(n) is the total number of occurrences of the noun n. In our computations, we used Google N-gram corpus4 to compute concept probabilities.

The utilisation of information content in the key-concept computations above turns out to be quite intuitive. The information content as a measure of concept specificity in context of key-concept extraction is pretty analogical to inverse document frequency as a measure of word specificity in context of keyword extraction. To see this analogy better, we can write the inverse document frequency as:

TABULKA 1

TABULKA 2

ROVNICA CISLO 5

As we can see, the main distinction is that the inverse document frequency is proportional to the probability of occurrence of word w within a document, whereas the information content is proportional to the overall probability of a concept c. More detailed discussion on TF-IDF from probabilistic perspective can be found in [20].

IV. EVALUATION

We empirically evaluated the quality of extracted keyconcepts in classification of plain texts. We used 20 newsgroups dataset5, which is commonly used in text classification. It contains 20 topic categories, each with 1000 documents, yielding 20; 000 documents altogether. As a baseline to compare to, we used standard TF-IDF vector representation. We evaluated the performance of two standard classifiers - k- nearest neighbours and Na¨?ve Bayes classifier. We can see detailed results of achieved classification accuracy in Figure 1 (on vertical axis, there are classification categories).

According to these results, our method achieves better classification accuracy than the standard TF-IDF. In construction of TF-IDF vector, we do some common preprocessing. We remove English stopwords, do POS-tagging to get only nouns (including compound nouns) as candidate words and we use Porter stemmer to obtain stems6. Finally, we prune some very common (corpus frequency above 30frequency below 3vector contains substantially greater number of words compared to at most 20 key-concepts obtained by our method.

We experimented with different number of extracted keyconcepts. It turns out that we succeed to extract the most relevant concepts as the very first few ones, which we can see from results in Table I. Even for only top 5 extracted key-concepts, the accuracy scarcely changes and even for just top 3 key-concepts, the results are still reasonable compared to baseline. We used Na¨?ve Bayes classifier to obtain these results. We also experimented with different setting of k in k-nearest neighbours classifier. We tried multiple values ranging from 1 up to 20, but we got the best results with k set to 1.

To summarise, we present a comparison of achieved classification accuracies with different methods and classifiers in Table II. We achieved better results with Na¨?ve Bayes classifier than k-nearest neighbours. This means that Na¨?ve Bayes classifier able to learn better weights than concept weights inferred by our method, which indicates that they are not quite optimal yet and are subject to our next research. We can observe the worst results achieved by our method compared to TF-IDF in classifying documents from misc:forsale category. This category deals with miscellaneous items for sale. These items are rather diverse and the concept of selling them gets lost. This causes TF-IDF vector to perform substantially better in this category, since it captures also the words related to selling.

To discuss the benefits of our approach in a real-world scenario, we showcase sample results of extracted key-concepts for Wikipedia article about “data structure” (see Table III). To present our contribution in the proposed concept ranking method, we point out the difference in results when considering the information content and when the concept ordering ignores it. This difference turns out to be a key-factor of performance gain in the classification above. In the first approach, we perform PageRank of the concepts using just the hypernymy, holonymy and collocation relations among them (inspired by TextRank keyword extraction). The second approach considers also the information content of the concepts, as an analogy to notion of inverse document frequency, which supplements the PageRank value of the first approach in the final value of concept relevance. We can see that using the second approach (the right part of Table III), the results are more reasonable compared to the more noisy concepts produced by the first approach (the left part of Table III).

V. CONCLUSION

We present the promising results of our key-concept extraction algorithm. It provides very efficient representation of document content - very concise, yet still of sufficient quality, as we demonstrate in evaluation of classification task. Such key-concept representation has several advantages compared to standard keyword vector. The major drawback of keyword vector is its representation - it is just a vector of words, where many of them can be (and often are) ambiguous. As for the key-concept representation, every concept has very concrete interpretation. Since it corresponds exactly to some WordNet synset, we can easily retrieve exact meaning of it.

Every synset contains a gloss, which may contain definition or some example sentences of it. In addition, we know exact relations to other synsets, like hypernym, hyponym, holonym, meronym, etc7. With all these information at hand, we can easily derive the exact meaning of given synset and thus, also of every extracted key-concept.

Another advantage of our key-concept representation is its space efficiency. We can extract high-quality key-concepts, which are satisfyingly accurate to be used in classification. We can summarise the main focus of given article with just a few extracted key-concepts and still retain the conceptual notion of its focus. Such concise representation is quite convenient for use on the Web. It also contributes to better performance, which is crucial for today's big data. It could help in building "more semantic" Web due to its relatively simple and fast acquisition. Moreover, it should be more feasible to use in personalisation due to its unambiguous exact interpretation and greater information content. Last, but not least, since we used Google N-gram corpus to approximate concept probabilities, it allows us to perform online key-concept extraction (i.e. we do not need to have the whole corpus in advance), which supports the idea of using it on the Web.

ACKNOWLEDGMENT

This work was partially supported by the the Scientific Grant Agency of Slovak Republic, grants VG1/0675/11, VG1/0971/11 and by the Slovak Research and Development Agency under the contract No. APVV-0208-10.

REFERENCES