Extracting Key Terms From Noisy and Multi-theme Documents

Maria Grineva
Institute for System Programming of the Russian Academy of Sciences
maria.grineva@gmail.com

Maxim Grinev
Institute for System Programming of the Russian Academy of Sciences
maxim@grinev.net

Dmitry Lizorkin
Institute for System Programming of the Russian Academy of Sciences
lizorkin@ispras.ru

ABSTRACT

We present a novel method for key term extraction from text documents. In our method, document is modeled as a graph of semantic relationships between terms of that document. We exploit the following remarkable feature of the graph: the terms related to the main topics of the document tend to bunch up into densely interconnected subgraphs or communities, while non-important terms fall into weakly interconnected communities, or even become isolated vertices. We apply graph community detection techniques to partition the graph into thematically cohesive groups of terms. We introduce a criterion function to select groups that contain key terms discarding groups with unimportant terms. To weight terms and determine semantic relatedness between them we exploit information extracted from Wikipedia.

Using such an approach gives us the following two advantages. First, it allows effectively processing multi-theme documents. Second, it is good at filtering out noise information in the document, such as, for example, navigational bars or headers in web pages.

Evaluations of the method show that it outperforms existing methods producing key terms with higher precision and recall. Additional experiments on web pages prove that our method appears to be substantially more effective on noisy and multi-theme documents than existing methods.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Wikipedia, community detection, graph analysis, keywords extraction, semantic similarity

1. INTRODUCTION

Key terms (sometimes referred to as keywords or key phrases) are set of significant terms in a text document that give high-level description of its content for readers. Key term extraction is a basic step for various tasks of natural language processing, such as document classification, document clustering, text summarization and inferring a more general topic of a text document [10].

Key term extraction is the core task of Internet content-based advertising systems, such as Google's AdSense and Yahoo! Contextual Match. They find relevant key terms on a web page, and then display advertisements based on those key terms. The quality of key terms extracted from a web page is a crucial issue for such systems: 10% improvement of key terms quality lead to nearly 10% higher click-through rate, directly increasing the overall effectiveness of a given advertisement [25]. Such systems raise the following new challenges for key term extraction techniques. First, they need to extract key terms from web pages that are typically *noisy*, i.e. overburden with information irrelevant to the main topic of the page, such as navigational information (e.g. side bars/menus), comments, future announces, etc. Second, they need to extract key terms from portal home pages which usually include many articles on different topics (for example, the homepage of BBC News [1]). Thus, the key terms extraction technique must be *noise* and *multi-theme stable*.

State-of-the-art approaches for key terms extraction are based on *statistical learning* that requires hand-created training set. In such approaches a document is treated as a set of semantically independent terms that must be classified as either positive or negative examples of keyphrases. The classifier is trained using statistics about term occurrence patterns found in the training set. In this paper, we propose a new approach to key terms extraction that is different in two points. First, instead of using statistics gathered from a training set, which might be stale or domain-specific, we use semantic information derived from a universal knowledge base (namely, Wikipedia), which provides up-to-date and extensive coverage for a broad range of domains. Second, our method utilizes more information from within the processed document by identifying and analyzing semantic relatedness between terms in the document. We show that our method produces key terms with quality not lower compared to some state-of-the-art methods, at the same time being more effective on extracting key terms from noisy and multi-theme documents.

The fundamental brick of our method is Wikipedia-based

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

[1] http://news.bbc.co.uk/

semantic relatedness of terms. Wikipedia [2] is a free online encyclopedia that has grown to become the largest online repository of encyclopedic knowledge. It contains millions of articles available for a large number of languages. As for September 2008, English Wikipedia contains over 2.5 millions articles (over 6 millions if consider redirects). In addition to being the largest vocabulary ever existed, Wikipedia is also a rich source of terms relationships expressed via extensive number of cross-references and hierarchical categories. Recent works on computing semantic relatedness measure of terms using Wikipedia [18, 17, 6, 23, 26] turned Wikipedia into a usable and exceptionally powerful tool for our work and many other natural language processing (NLP) and information retrieval (IR) applications. We discuss these works in more detail in Section 2 of this paper.

The main ideas behind the proposed method are as follows. We generate candidate terms by detecting all Wikipedia terms in the content of a document. At this step each ambiguous term is assigned with its proper meaning using one of the existing word sense disambiguation techniques [12, 22, 11, 26]. The document is then modeled as a *semantic graph* of the candidate terms. Semantic graph is a weighted graph where *vertices* are terms, *edge* between a pair of terms means that these two terms are semantically related, the *weight* of the edge is the semantic relatedness measure of the two terms. Analyzing semantic graphs constructed in this way for various documents we observed that they have a remarkable property: the terms related to the common topics tend to bunch up into dense subgraphs or *communities*, and the most massive and densely interconnected groups of terms typically correspond to the main topics of the processed document! We exploit this property to distinguish between key terms and non-important terms in a document: we choose terms from the most dense groups as key terms discarding terms from sparse groups as irrelevant to the main topic of the document. To detect communities in the semantic graph of a document we apply Girvan-Newman network analysis algorithm that has been proved to be highly effective at discovering community structure in various computer-generated and real-world networks [19].

Our method provides the following main advantages:

- *No training.* Instead of training the system with hand-created examples, we use semantic information derived from Wikipedia.

- *Thematically grouped key terms.* The output of the method is groups of semantically related terms, and each group relates to one of the main topics of the document. Thematically grouped key terms can significantly improve further inferring of document topics using, for example, spreading activation over Wikipedia categories graph as described in [24]. Applying this technique ([24]) to each group instead of all key terms allows inferring topic more accurately.

- *High accuracy.* Our evaluation using human judgments shows that our method produces key terms with overall precision and recall higher than baseline (TFxIDF [21]) and several state-of-the-art systems (Yahoo! keyword extractor, Wikify! [14] and TextRank [15]).

- *Noise and multi-theme stability.* Noisy and multi-theme documents are common among web pages. To test noise and multi-theme stability of our method we have conducted specialized evaluation on web pages. When applied to web pages our method produces key terms with substantially higher accuracy than the baseline and state-of-the-art systems noted above.

[2]www.wikipedia.org

2. BACKGROUND

The background of our method consists in the following techniques: measure of terms *semantic relatedness* computed over Wikipedia corpus; and network analysis algorithm for detecting community structure in networks. We discuss each of the techniques in the following subsections.

2.1 Computing semantic relatedness over Wikipedia

A measure of semantic relatedness is computational mean for calculating the association strength between terms. More precisely it assigns a score for a pair of Wikipedia terms that represents the strength of relatedness between the terms. Terms semantic relatedness can be inferred from a dictionary or thesaurus (for example, WordNet [16]), but here we are interested in terms semantic relatedness derived from Wikipedia. During the last three years, there have appeared a good few of works on computing Wikipedia-based semantic relatedness using different methods [18, 17, 6, 23, 26]. Wikipedia-based semantic relatedness measure for two terms can be computed using either the links found within their corresponding Wikipedia articles [18, 17, 26], or Wikipedia categories structure [23], or the article's textual content [6]. See [17] for an insightful comparison of the many of existing Wikipedia-based semantic relatedness measures. There is a bunch of works on using Wikipedia-based semantic relatedness to solve the many basic NLP/IR tasks such as: word sense disambiguation [12, 22, 11, 26], document topic inferring [24], document categorization [7], coreference resolution [23]. In our method we use semantic relatedness to build a semantic graph of candidate terms, which are Wikipedia terms found in a document. The structure of the graph is then analyzed to derive key terms. While our method does not provide any requirements on the way of computing semantic relatedness, the effectiveness of our method depends on the effectiveness of the exploited semantic relatedness measure. For the evaluation of our method in this paper we use the semantic relatedness measure proposed in [26].

2.2 Detecting communities in networks

To automatically detect densely interconnected groups of terms (communities) in semantic graph we need to apply a proper graph clustering technique to such graph. It is crucial that the number of communities is different for different semantic graphs, and is not known in advance. Thus, a class of clustering techniques that require some predefined number of clusters for input (for example, different variations of k-means, minimum-cut graph partitioning algorithm) does not fit our task.

We turn to the hierarchical clustering approach focusing at techniques that specialize in discovering a natural division of networks into communities, when the number of communities is uncertain. Fortunately, this field has been well studied. Many algorithms have been proposed and applied with great success to social networks [27], citation networks

[20, 4], purchasing network [3], biochemical networks [8] and many others. However, to the best of our knowledge, there are no applications of community detection algorithms to the Wikipedia-based networks.

Among these algorithms the one invented by M. E. J. Newman and M. Girvan [19] and its further optimization for large networks [3] are most commonly used and have been proved to be highly effective at discovering community structure in both computer-generated and real-world network data. We use this algorithm in our method. This algorithm iteratively removes edges from the network to split it into communities. The edges removed being identified using the graph-theoretic measure of betweenness, which assigns a number to each edge that is large if the edge lies "between" many pairs of nodes. To estimate the goodness of the certain graph partition, the authors of [19] propose the notion of modularity. *Modularity* [19] is a property of a network and a specific proposed division of that network into communities. It measures when the division is a good one, in the sense that there are many edges within communities and only a few between them. In practice, modularity values that fall in the range from about 0.3 to 0.7 indicate that network has quite a distinguishable community structure.

3. RELATED WORK

There are a number of classical approaches to extracting key terms. The simplest approach is to use a frequency criterion (or TFxIDF model [21]) to select keywords in a document. However, this method was generally found to lead to poor results, and consequently other methods were explored. The state-of-the-art in this area is currently represented by supervised learning methods, where a system is trained to recognize keywords in a text, based on lexical and syntactic features. In this class of systems the most prominent one is KEA [5]. In KEA the candidate terms are represented using three features: TDxIDF, distance (the number of words that precede the first occurrence of the term, divided by the number of words in the document) and keyphrase-frequency (the number of times a candidate term occurs as a key term in the training documents). The classifier is trained using the naive Bayes learning algorithm. Thus, KEA analyses only simple statistical properties of candidate terms and considers each candidate term independently from the other terms in the document. In contrast to this, our method analyses semantic term relationships in the document. Another difference is that KEA depends on the training set and may provide poor results when the training set does not fit well the processed documents while our method does not require training.

Recently a number of new methods have been proposed that extend classical methods with new features computed over Wikipedia corpus. For example [11] introduces *node degree* feature that indicates how a candidate term is connected via Wikipedia links to other candidate terms in the document (i.e. the number of links between the Wikipedia article corresponding to a candidate term and the Wikipedia articles corresponding to other candidate terms). So candidate terms with high node degree are those that have many related terms in the document. Wikify! system [14] introduces *keyphraseness* feature of a candidate term that is defined as the number of Wikipedia articles in which the term appears and is marked up as a link divided by the total number of Wikipedia articles where the term appears. This feature can be interpreted as probability that the candidate term is selected as a key term in a Wikipedia article as according to the Wikipedia editorial rules only key terms should be used as links to other articles. Wikify! uses keyphraseness as the only feature to select key terms. Comparison of keyphraseness with TFxIDF demonstrates improvements in both precision and recall by about 10% each. In our method we use keyphraseness to compute community informativeness (see Section 4.5).

There is an alternative class of approaches that selects key terms by analyzing syntactic or semantic term relatedness in a document. In the approaches a document is modeled as a graph of candidate terms in which edges represent a measure of term relatedness. Some graph analysis technique is used to select key terms. Usually the graph is analyzed using a graph-based ranking algorithm (such as PageRank [2], HITS [9], etc) to range candidate terms. Top N terms with the highest scores are selected as key terms. For example, in [15] the graph is constructed using a syntactic term relatedness (namely, co-occurrence relation) defined as follows: two terms are related if they co-occur within a window of maximum N words. To rank candidate terms in the constructed graph Google's PageRank algorithm [2] is applied. The authors of this work reported that their method outperforms classical statistics-based approaches with respect to precision and F-measure although the recall is lower then in the statistics-based methods. Another variation of the same approach was proposed in [7] as a part of a document categorization algorithm. In this work the graph is constructed using a semantic term relatedness computed over an ontology generated from Wikipedia as proposed in [1]. Candidate terms are ranked with centrality scores computed using the geodesic closeness measure. The author of the latter work does not evaluate the proposed key term extraction method itself but shows superior quality comparing their document categorization method with statistics-based categorization methods. Both of the methods described above are unsupervised and does not require training. Our method also takes the same approach based on analysis of semantic graph constructed from a document. But in our work we perform more sophisticated graph analysis by applying Girvan-Newman algorithm. It provides the following benefits. First, as we demonstrate in the evaluation section, our algorithm outperforms those based on a graph-based ranking algorithm. Second, the result of our method is semantically grouped key terms that are useful for further processing (such as document topic inference) or interpretation by humans.

4. METHOD FOR KEY TERMS EXTRACTION

The method consists of the five steps that we describe in detail in the following subsections: (1) candidate terms extraction; (2) word sense disambiguation; (3) building semantic graph; (4) discovering community structure of the semantic graph; and (5) selecting valuable communities.

4.1 Candidate Terms Extraction

The goal of this step is to extract all terms from the document and for each term prepare a set of Wikipedia articles that can describe its meaning.

We parse the input document and extract all possible n-grams. For each n-gram we construct its variations using

different morphological forms of its words. We search for all n-gram variations among Wikipedia article titles. Thus, for each n-gram a set of Wikipedia articles can be provided.

Constructing different morphological forms of words allows us not to miss a good fraction of terms. For instance, *"drinks"*, *"drinking"*, and *"drink"* can be linked to the two Wikipedia articles: *"Drink"* and *"Drinking"*.

It is a typical problem with traditional key term extraction techniques when nonsense phrases such as e.g. "using", "electric cars are" appear in the result. Using Wikipedia articles titles as a controlled vocabulary, allows us to avoid this problem, all of the key terms produced by our method are acceptable phrases.

4.2 Word Sense Disambiguation

At this step we need to choose the most appropriate Wikipedia article from the set of candidate articles for each ambiguous term extracted on the previous step.

It is an often situation in natural language when a word is *ambiguous*, i.e. carries more than one meaning, for example: the word "platform" can be used in the expression "railway platform", or it can refer to a hardware architecture or a software platform. The correct sense of an ambiguous word can be selected based on the context where it occurs, and correspondingly the problem of *word sense disambiguation* is defined as a task of automatically assigning the most appropriate meaning (in our case, the most appropriate Wikipedia article) to a word within a given context.

There are a number of works on disambiguating terms using Wikipedia [26, 11, 22, 12, 13]. For evaluation in this paper we used the method described in [26]. In [26] authors make use of Wikipedia disambiguation and redirect articles to obtain candidate meanings of ambiguous terms. For each ambiguous term disambiguation page contains all of the meanings of the term, which are separate articles in Wikipedia with their own link structure. For example, the article "platform (disambiguation)" contains 17 meanings of the word "platform". Then in [26] semantic relatedness measure is used to pick the meaning that has the highest relevance to the context where the ambiguous term appears.

The result of this step is a list of terms, where each term is assigned with a single Wikipedia article that describes its meaning.

4.3 Building Semantic Graph

At this step we build a semantic graph from a list of terms obtained at the previous step.

Semantic graph is a weighted graph where each *vertex* is a term, *edge* between a pair of vertices means that the two terms corresponding to these vertices are semantically related, the *weight* of the edge is the semantic relatedness measure of the two terms.

Figure 1 shows semantic graph built from a news article *"Apple to Make ITunes More Accessible For the Blind"*. This article tells that the Massachusetts attorney general's office and the National Federation of the Blind reached an agreement with Apple Inc. under which it will make its music download service (ITunes) accessible to the blind consumers using screen-reading software. In Figure 1 you can see that terms related to *Apple Inc.* and *Blindness* constitute two dominant communities, and terms like *Time*, *Month*, *Massachusetts* or *Consumer* fall into peripheral and weakly connected communities.

An important observation is that disambiguation mistakes (in Figure 1: *Home Office*, *Free agent*, *Grocery store*) tend to fall into weakly connected communities or even become isolated vertices in a semantic graph and not to adjoin to dominant communities.

4.4 Discovering Community Structure of the Semantic Graph

We use algorithm proposed by M. E. J. Newman and M. Girvan [19] to discover community structure of the semantic graph built on the previous step. The algorithm divides the input graph into a number of subgraphs that are likely to be dense communities.

We observed that semantic graphs constructed from an average text document (one page news article, or a typical academic paper) have modularity values between 0.3 and 0.5. That is an indication that application of the Girvan-Newman algorithm to semantic graphs indeed makes sense as they have quite a distinguishable community structure.

4.5 Selecting Valuable Communities

The goal of this step is to rank term communities in a way that highest ranked communities would contain terms semantically related to the main topics of the document (key terms), and the lowest ranked communities contain not important terms, and possible disambiguation mistakes (terms which meaning was chosen wrong on the second step).

Ranking is based on the *density* and *informativeness* of communities. Density of a community is a sum of weights of all inner-community edges divided by the number of vertices in this community.

While experimenting with existing approaches discussed in Section 3, we observed that using keyphraseness measure of terms can help ranking communities in a proper way. Keyphraseness measure gives higher values to the named entities (for example, *Apple Inc.*, *Steve Jobs*, *Braille*) than to general terms (*Consumer*, *Agreement*, *Information*). We compute keyphraseness measure of terms using Wikipedia corpus as described in [11]: the number of Wikipedia articles in which the term appears and is marked up as a link divided by the total number of Wikipedia articles where the term appears. *Informativeness* of a community is a sum of keyphraseness measure of all terms in a community divided by the number of terms.

Eventually, the rank value assigned to each community is its density multiplied by its informativeness. Communities are then sorted according to this value, thus, we obtain a sorted sequence of communities each assigned with its rank value.

We have observed the following important feature of the communities rank values. In the sorted sequence communities rank values do not decrease evenly. Instead, there is almost always a single evident **decline**. Figure 2 demonstrates the decrease in community rank scores taken from our example. Communities are numerated according to the marks in Figure 1. In Figure 2 such decline is observed between second and third communities. According to the semantic graph shown in Figure 1 this decline separates two communities with terms related to the main topics of the news article (*Apple Inc.* and *Blindness*) from other communities with less important terms.

On average, such a decline is 20 - 25 times more than the difference between rank values of other neighbouring com-

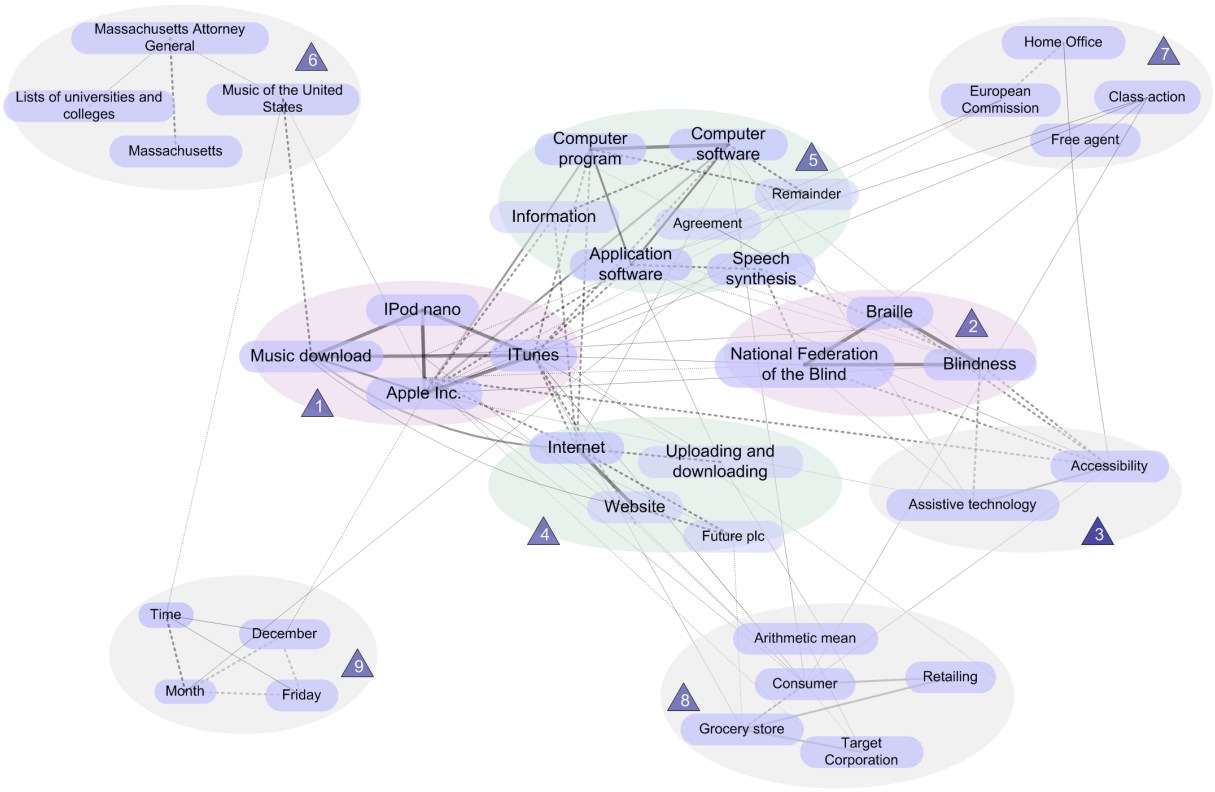

Figure 1: Semantic graph built from the news article *"Apple to Make ITunes More Accessible For the Blind"*

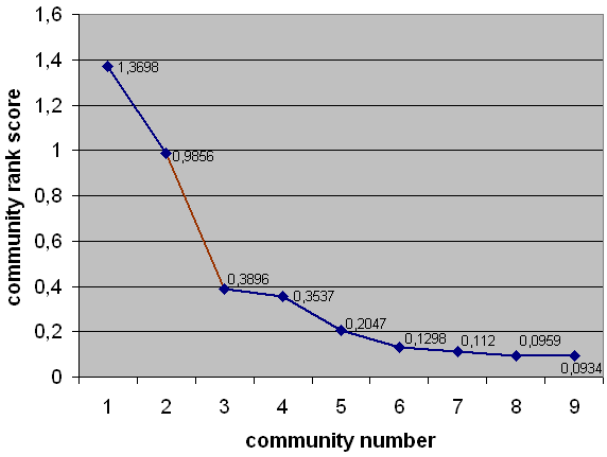

Figure 2: Decline in community scores

munities in the sorted sequence. In section 5 we prove experimentally that this decline often separates communities of important terms (that go before the decline) from communities of non-important terms (that go after the decline). It makes sense to use this decline as an indicator when selecting valuable communities, thus, determining how many of the communities to accept, while rejecting the rest of them.

5. EXPERIMENTAL EVALUATION

This section reports the experimental results comparing our method with several baseline and state-of-the-art methods. Since these methods are not specifically designed to handle noisy content and compound multi-theme documents, we first evaluate the methods on presumably single-topic documents represented in plain text and being noise-free. We then conduct twofold evaluation of the methods on web pages containing noise and compound web pages with diverse topics.

Since there is no standard benchmark for evaluating the quality of the extracted key terms, we conducted a human study by asking annotators to manually select key terms from a test collection. We found that reasonable examples of single-topic noise-free documents are posts from technical blogs obtained via RSS feeds. Our test collection consists of 252 posts from the following technical blogs: *Geeking with Greg* by Greg Linden [3], *DBMS2* by Curt Monash [4] and *Stanford Infoblog* by people from Stanford Infolab [5]. Twenty-two

[3]http://glinden.blogspot.com/
[4]http://www.dbms2.com/
[5]http://infoblog.stanford.edu/

annotators took part in the human study. These included 5 developers having either M.Sc. or Ph.D. degree in computer science, and 17 undergraduate students in the field of computer science. Each document was analyzed by 5 different annotators. Eventually, we considered a key term to be valid if at least two of the participants identified this key term in the document. For each document two sets of key terms were built. To build the first set, called *uncontrolled key terms*, each annotator was instructed to identify from 5 to 10 key terms as they appear in the document. The second set, called *controlled key terms*, is constructed on the basis of the first one by the same annotator as follows. All the key terms from the first set which have a corresponding Wikipedia article have to be replaced with the title of the article. The key terms that do not match Wikipedia articles remain in the second set as they appear in the first one. Thus constructing the second set implies manual disambiguation of key terms against Wikipedia as an allowed vocabulary. Doing that annotators were instructed to consider Wikipedia redirects and the articles, which they direct to as being the same term since they inherently represent synonyms. These two sets of key terms are required so that we can evaluate various methods based on uncontrolled or controlled Wikipedia-based vocabulary using the same test collection. As a result, we have got 2009 key terms for the 252 blog posts, with about 93% of them being Wikipedia titles.

The techniques presented in the paper were implemented with the following architectural design principles. For achieving the best computational efficiency, all necessary data concerning Wikipedia articles names, Wikipedia link structure and statistical information about the Wikipedia corpus (for instance, terms keyphraseness) are kept in main memory. With the recent Wikipedia being quite a large knowledge base, our Wikipedia knowledge base takes 4.5 Gigabytes in the main memory. A dedicated machine with 8Gb RAM was used for the evaluation, and client applications access the knowledge base via remote method invocation. With the requirement for having access to similarity scores for virtually *every* term pair in Wikipedia, similarity scores are not computed offline in advance, but are rather computed on demand on the fly using Wikipedia link structure [26].

We have set up a web service upon this implementation, so that our key terms extraction method is accessible online via a simple web interface [6].

Before we proceed with the comparisons we consider experimental proof of the hypothesis stated in Section 4.5 - that decline in community scores can be used as a criteria to determine the number of term communities to be returned as key terms.

5.1 What Does Decline in Communities Rank Scores Mean?

The result of our method is a sequence of communities of term with decreasing rank scores. As we mentioned in Section 4.5, there is almost always an evident decline in communities rank scores. What does this decline mean and how it can help in determining the number of valuable communities? We presumed that this decline can serve as a border between communities of important terms (with higher rank scores) and the communities of non-important terms (with lower rank scores), so, we conducted the following evaluation to check this presumption.

For each document from our test collection we conducted the following evaluation. We applied our method for a document, thus obtaining a sorted sequence of communities of terms. Then we, in serial, accepted every possible number of communities: first, we accepted only the single highest ranked community, then two highest ranked communities, and at least we ended up with all communities. For every case we computed F-measure comparing the accepted terms with the manually extracted terms for this document. We then found out the maximum F-measure. Eventually, we checked if the number of communities that give the maximum F-measure is indeed the number of communities that go before the decline.

For our test collection we found out that the decline coincides with the maximum F-measure in **73%**. Thus, we consider decline as a good indicator for selecting important communities. In the evaluation of precision, recall and F-measure that follows in this section, we rely on the decline instead of accepting some predefined number of communities (or key terms).

5.2 Evaluation of Key Terms

In this section, our method is evaluated against existing methods on the collection of single-topic noise-free documents described above.

We chose the following methods for the comparison:

TFxIDF is a conventional baseline used in scientific literature for comparison of key term extraction algorithms. For extracting candidate terms that are then ranked using TFxIDF measure, the technique described in Subsect. 4.1 was applied. Having the same candidate term extraction technique for both methods makes the comparison more illustrative. For obtaining the document-frequency component required for the TFxIDF method, the entire Wikipedia was used as a training corpus. As the TFxIDF method allows merely ranking candidate terms but provides no notion about the *number* of key terms to be selected, the top-K terms are selected with K equal to the average number of manually assigned terms within the test set. Since the TFxIDF method, as it is described above, returns key terms from Wikipedia corpus we use the controlled key terms to evaluate the method.

Yahoo! Terms Extractor is chosen for comparison as being a state-of-the-art industrial tool for key term extraction. Its implementation is available via open API [7]. It takes as input a text document and returns a list of significant words or phrases extracted from the document. Since Yahoo! implements a method with uncontrolled vocabulary, we calculate precision, recall and F-measure using the set of uncontrolled key terms.

Wikify! uses keyphraseness computed over Wikipedia to select key terms (see Section 3 for details). Its implementation is available online as a demo[8]. The number of key terms selected by the method is specified by the special external parameter "density" as a fraction of the overall number of words in a document being

[6] http://www.modis.ispras.ru:7005/demo/keywords/

[7] http://developer.yahoo.com/search/content/V1/termExtraction.html

[8] http://wikifyer.com/

processed. Depending on a particular document size, we specified the density value in a way to obtain the number K of key terms equal to the average number of manually assigned terms. As Wikify! uses controlled vocabulary based on Wikipedia, we use the controlled key terms to evaluate the method.

TextRank. As the last baseline, we implemented the TextRank approach to key terms selection as proposed in [15] (see Section 3 for details). For adapting the approach to the controlled vocabulary of Wikipedia terms, only document terms which have corresponding Wikipedia articles were chosen as vertices in the TextRank model, alternatively to *all* words passing syntactic filters originally used in [15] due to uncontrolled vocabulary. Since titles of Wikipedia articles are generally noun phrases, the modification made to the approach fully corresponds to the recommendation given on syntactic filters in the original TextRank proposal. Further parameters were exactly those that performed best in the TextRank experiments: the co-occurrence window of 2 for term relationships and undirected treatment of edges between vertices. As TextRank was modified to work with Wikipedia-based vocabulary, we use the controlled key terms to evaluate the method.

The methods outlined above were compared using the traditional measures of precision, recall and F-measure calculated with respect to key terms extracted by a method and by human annotators. Namely, *precision* was calculated as a fraction of terms automatically extracted by a method that were also extracted by humans:

$$\text{precision} = \frac{|\{\text{manually selected}\} \bigcap \{\text{machine-selected}\}|}{|\{\text{machine-selected}\}|} \ ,$$

with {manually selected} denoting the set of all terms identified for a document by humans, {machine-selected} denoting the set of all terms extracted for the same document by a method and $|S|$ denoting the number of items in a set S.

Recall was calculated as the fraction of the manually extracted key terms that were also extracted by an automatic method:

$$\text{recall} = \frac{|\{\text{manually selected}\} \bigcap \{\text{machine-selected}\}|}{|\{\text{manually selected}\}|} \ .$$

The weighted harmonic mean of precision and recall, *F-measure* was calculated traditionally as:

$$\text{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \ .$$

Table 1 summarizes the performance of the methods for the test data set. The results presented in the table show that the method presented in this paper (communities-based) exhibits the best performance among all the other methods considered in the experiment. Although TFxIDF method was based on the same candidate terms extraction technique as the communities-based method, TFxIDF method showed the worst performance due to weaker consistency between candidates ranking produced by the TFxIDF scheme and human judgments. TextRank method shows the second to the best performance, which confirms our earlier assumption that considering pair-wise semantic relationships between candidate terms and treating them as a graph to analyze improves the quality of the extracted key terms.

	Performance, %		
Method	Precision	Recall	F-measure
TF×IDF	26.3	44.8	33.1
Wikify!	31.7	50.7	39.0
Yahoo! terms extractor	33.0	52.5	40.5
TextRank	34.6	54.8	42.4
Communities-based	**35.1**	**61.5**	**44.7**

Table 1: Performance of different key term extraction methods for noise-free data set

It is worth noting that while TextRank treats all candidate terms as a *single* graph when ranking the terms, the communities-based method analyses each community independently that allows more representatively choosing candidate terms from different thematic groups.

5.2.1 Revision of Precision and Recall

However, we have revisited the measuring of precision and recall according to the specifics of our method. The important thing is that our method, on average, extracts more terms than a human. More precisely, our method typically extracts more related terms in each thematic group than a human. For example, consider Figure 1, for the topic related to *Apple Inc.* our method extracts terms: *Music download*, *Apple Inc.*, *ITunes*, *IPod nano*; while a human typically identifies less, and tends to identify named entities: *Apple Inc.*, *ITunes* and *IPod nano*. That means that, possibly, sometimes our method produces better terms coverage for a specific topic than an average human. And this is a reason that we measure the precision and recall in another way also.

Each participant of the evaluation was asked to revisit his key terms in the following way. For each document he was provided with key terms extracted automatically for this document. He had to review these automatically extracted key terms and, if possible, extend his manually identified key terms with some from the automatically extracted set. It appeared that humans indeed found out relevant key terms that they had not extracted before, and extended their key terms.

After this revision we obtained 389 additional manually selected key terms, that gives 1.2 additional terms per each blog post in average. With the new manually selected terms added, precision of the communities-based method becomes **46.1**%, recall becomes **67.7**%.

5.3 Evaluation on Web Pages

In this section key term extraction is evaluated in the presence of noise in the processed documents and performance of different methods is compared. As discussed in the introduction, automatically extracting key terms from noisy documents is an important task for content-targeted advertisement.

Pages on the web provide a representative and practical test data for performing the evaluation. Even with HTML markup removed, most web pages contain plenty of text irrelevant to the main content of the page such as menus and navigation bars, comments, footers, etc., that have to be filtered out by an automatic method.

In the first subsection below, the methods of key term extraction are evaluated on web pages for noise stability. In the

News	Blogs	Forums	Social networks	Product reviews
161	127	92	76	53

Table 2: Number of web pages of each kind considered for evaluation

Method	Performance, %		
	Precision	Recall	F-measure
TF×IDF	17.4	34.6	23.2
Wikify!	22.9	39.8	29.1
Yahoo! terms extractor	25.6	40.4	31.3
TextRank	26.1	43.9	32.7
Communities-based	**31.2**	**60.7**	**41.2**

Table 3: Performance of different key term extraction methods for noisy data set

second subsection, key term extraction is evaluated for an even more complicated area of compound pages, with each page containing several articles that cover diverse topics.

5.3.1 Noise stability

For evaluating noise stability of key term extraction, we collected 509 real-world web pages. In order to additionally investigate performance of key terms extraction for various kinds of web pages our test collection includes 5 kinds of Web pages. Table 2 shows the detailed break down.

The selected web pages were processed by annotators in the same manner as described in Section 5. In order to quantitatively evaluate the performance of the methods the same measures of precision, recall and F-measure were used as defined in Section 5.2.

Table 3 summarizes the performance of different methods of key terms extraction for the noisy test set. Compared to Table 1 considered in the previous section, it can be observed that our communities-based method is more stable to noise than other methods that have their performance degrading faster. The reason for the observed tendency is that noise tends to exhibit less semantic relatedness with the main topic of a page and thus falls outside dominant communities in the communities-based method, while other methods have virtually no mechanism for distinguishing noisy candidate terms from correct ones.

Table 4 shows performance of the communities-based method for different kinds of web pages considered in the experiment. The results presented in the table show that key terms for product reviews and technical blogs can be extracted with better precision. This is probably due to these kinds of web pages having more formal textual content and thus being more stable to noise. Forums, news articles and social networks generally have less formal textual content and are more affected by noise that leads to somewhat worse performance.

5.3.2 Multi-theme stability

For testing the ability of our method to correctly extract key terms from compound texts consisting of several separate articles, we chose 50 web pages with diverse topics. These included front web pages of popular news sites and

Kind of web pages	Performance, %		
	Precision	Recall	F-measure
News	30.5	59.4	40.3
Blogs	34.3	64.8	44.9
Forums	27.6	56.2	37.0
Social networks	28.4	56.9	37.9
Product reviews	36.0	67.9	47.1

Table 4: Performance of the communities-based key term extraction method for different kinds of web pages

Method	Performance, %		
	Precision	Recall	F-measure
TF×IDF	9.2	15.0	11.4
Wikify!	11.1	23.3	15.0
Yahoo! terms extractor	20.8	32.2	25.3
TextRank	15.4	20.5	17.6
Communities-based	**28.7**	**48.3**	**36.0**

Table 5: Performance of different key term extraction methods for compound texts

home pages of Internet portals with lists of featured articles. In addition to noisy content inherent to web pages, each of the selected pages additionally contained from 2 to 10 separate articles that were generally focused at describing diverse topics.

For obtaining the set of manually selected key terms that properly cover the content of a web page, each annotator was instructed to treat each article on a web page independently and to select key terms for that article disregarding the remaining content of the page.

Having this set of manually assigned key words, the following two experiments were conducted.

In the first experiment, performance of the communities-based method suggested in this paper was compared to the performance of existing methods on the test set. Since existing methods have no ability to semantically separate key terms obtained from different articles on a single page, we observed that candidate terms extracted by these methods from one–two articles on a page tend to dominate over all the candidate terms from the other articles. It results to incomplete sets of key terms for web pages with whole thematic groups of important terms missing in the result. Table 5 illustrates this tendency: difference in precision and recall between the communities-based method and existing methods becomes even more significant with compound texts considered.

In the second experiment, the ability of the communities-based method to select relevant key terms on different topics was specially evaluated using an additional methodology. In this experiment, communities of key terms produced by the method were shown to annotators to evaluate the relevance between these communities and the articles located on a compound page. Since several articles on a single page can have the same topic, e.g. Politics, Sport, Technology, etc., there are generally fewer communities extracted by the method than there are articles. However, we observed that *different* topics covered by articles usually have their corre-

sponding communities extracted by the method. To obtain a quantitative characteristics of such a correlation, each annotator was asked to revisit the pages he evaluated and to verify whether each article has a corresponding community of key terms extracted by the method. This additional user study showed that for 78% articles presented on compound pages the communities-based method was able to extract relevant thematic communities.

5.4 Computational Efficiency

When experimenting with the implementation, we observed that most computation time was consumed by (i) text parser for extracting candidate terms from input document and (ii) semantic graph construction that mainly consists in obtaining similarity scores for candidate term pairs. Compared to these preliminary steps, the running time for the remaining steps of the algorithms is negligible, with both community discovery and selection of valuable communities being essentially linear [3] in the number of edges in the semantic graph.

On average, it takes about 4 minutes to extract key terms from 100 blog posts.

6. CONCLUSION

We presented a novel method for extracting key terms from a text document. One of the advantages of our method is that it does not require any training, as it works upon the Wikipedia-based knowledge base. The important and novel feature of our method is that it produces groups of key terms, while each group contains key terms related to one of the main topics of the document. Thus, our method implicitly identifies main document topics, and further categorization and clustering of this document can greatly benefit from that. From implementation viewpoint the novel feature of our method is that, for the first time, an algorithm for detecting community structure of a network is applied to analyze a semantic graph of terms extracted from a document.

Our experimental results show that our method produces high-quality key terms comparable to the ones produced by state-of-the-art systems developed in the area. Evaluation proved that our method produces key terms with 67.7% recall and 46.1% precision, that we consider being significantly high. We also conducted experiments for multi-theme and noisy web pages with performance figures significantly higher than competitive methods. It allows us to conclude that a promising application of our method is to improve content-targeted advertising systems, which have to deal with such web pages. The implementation of our method is accessible online via a simple web interface [9].

7. ACKNOWLEDGEMENTS

We thank the developers and students in our department who participated in the test creation, Alexander Boldakov and Pavel Velikhov for valuable discussions.

8. REFERENCES

[1] S. Auer and J. Lehmann. What have innsbruck and leipzig in common? extracting semantics from wiki content. pages 503–517. 2007.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[3] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[4] D. J. de Solla Price. Networks of scientific papers. *Science*, 169:510–515, 1965.

[5] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-manning. Domain-specific keyphrase extraction. pages 668–673. Morgan Kaufmann Publishers, 1999.

[6] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*, pages 1606–1611, Hyderabad, India, 2007.

[7] M. Janik and K. J. Kochut. Wikipedia in action: Ontological knowledge in text categorization. *International Conference on Semantic Computing*, 0:268–275, 2008.

[8] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, 22(3):437–467, March 1969.

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[10] C. D. Manning and H. SchÃijtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999.

[11] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Wikipedia and AI workshop at the AAAI-08 Conference (WikiAI08)*, Chicago, US, 2008.

[12] R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[13] R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Proceedings of NAACL HLT 2007*, pages 196–203, 2007.

[14] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, New York, NY, USA, 2007. ACM.

[15] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[16] G. A. Miller, C. Fellbaum, R. Tengi, P. Wakefield, H. Langone, and B. R. Haskell. Wordnet: a lexical database for the english language. http://wordnet.princeton.edu/.

[17] D. Milne. Computing semantic relatedness using wikipedia link structure. In *Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC)*, Hamilton, New Zealand,

[9] http://www.modis.ispras.ru:7005/demo/keywords/

2007.

[18] D. Milne and I. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Wikipedia and AI workshop at the AAAI-08 Conference (WikiAI08)*, Chicago, US, 2008.

[19] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[20] S. Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, 4:131, 1998.

[21] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[22] R. Sinha and R. Mihalcea. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 363–369, Washington, DC, USA, 2007. IEEE Computer Society.

[23] M. Strube and S. Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, pages 1419–1424, Boston, Mass., July 2006.

[24] Z. Syed, T. Finin, and A. Joshi. Wikipedia as an Ontology for Describing Documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[25] W. tau Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 213–222, New York, NY, USA, 2006. ACM.

[26] D. Turdakov and P. Velikhov. Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation. In *Colloquium on Databases and Information Systems (SYRCoDIS)*, 2008.

[27] S. Wasserman, K. Faust, and D. Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.