Journal of Theoretical and Applied Information Technology
31st May 2018. Vol.96. No 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A GRAPH –BASED APPROACH FOR KEYPHRASES EXTRACTION USING A PRETOPOLOGICAL FORMALISM

[1]FADOUA GHANIMI, [1]ABDLWAHED NAMIR, [1]ILYAS GHANIMI
[1]University Hassan II, Department of Mathematics and Computer Sciences, Morocco
E-mail: [1]ghanimi_fadoua@yahoo.fr,

ABSTRACT

In this paper, we present a new graph -based approach for keyphrases extraction using the pretopology theory. We formalize the notion of neighborhood to express different types of connections that may exist between the terms of a document. The first part of the proposed method consists in first selecting the keywords candidates using a pattern approach; the second part consists in detecting keywords as the most representative candidates by means of graph-based ranking algorithm using a pretopological formalism.

Keywords: *keyphrases extraction, graph–based approach, ranking algorithm, pretopology*

1. INTRODUCTION

Keyphrases are small set of relevant words or word phrases of the document that can best describe his meaning. keywords can be used for automatic indexation, categorization, classification or summarization of documents [1],[2], [3].

The two main categories of keywords extraction methodologies are: Supervised and unsupervised.

The task of keyphrase extraction usually conducts in two steps: (1) extracting a bunch of words serving as candidate keyphrases and (2) determining the correct keyphrases using unsupervised or supervised approaches.

In the unsupervised approach, graph-based ranking methods perform the best. These methods construct a word graph based on word co-occurrences within the document firstly and then ranking the words according to their scores. As a result, the top ranked words are the key words we want.

Such text-oriented ranking methods have been successfully used to automate extraction of keyphrases, to extractive summarization and word sense disambiguation [4].

In this paper a novel graph based keyword extraction algorithm using a pretopological formalism has been proposed. The choice of the pretopology approach is motivated by the fact that it will allow us to decide on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information.

A large amount of research effort on mathematical modeling has been devoted to terms extraction problems. But, even though the pretopological model [19], has not been intensively used for such types of problems.

The pretopology is a mathematical tool for the analysis modeling and construction in various fields: social sciences, game theories, networks... It provides us a structuring process based on adherence and pretopological closures [22], and establishes a powerful tool for data analysis and automatic classification [21].

Given a finite set E, the adherence a(.) defined on its subsets has the advantage to express the neighborhood relationships.

In section 2, we present a state of art of the different automated unsupervised methods for keyterms extraction.

In section 3, the theory of pretopology and it's tools is presented.

In section 4, the new graph-based approach using the pretopological formalism for ranking and its main steps are introduced.

The conclusion of the paper is done in section 5.

2. STATE OF ART

A keyphrase extraction system typically operates in two steps:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1) Extracting a list candidate keyphrases using some heuristics.

2) Determining which of these candidate keyphrases are correct keyphrases using supervised or unsupervised approaches.

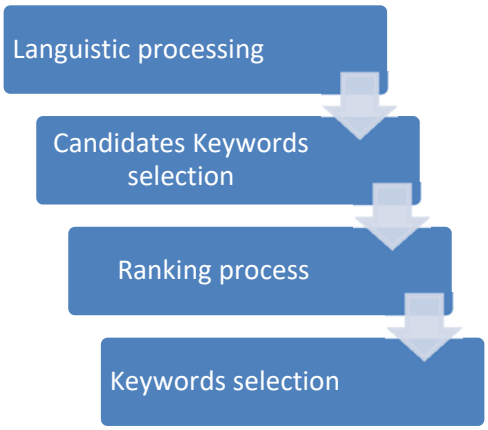

Figure 1: The keywords extraction process in unsupervised context

In this state of the art, we focus on the category of methods to which belongs our new approach, that is, unsupervised methods. The unsupervised methods process the documents one by one. These are first segmented into sentences, the sentences to words, and then grammatically labeled. Candidate keywords are selected, then ranked to extract only the most relevant ones (see Figure 1).

2.1 Selection of the candidate terms

The aim is to extract a set of phrases and words as candidate keyphrases using heuristic rules. These rules are designed to keep the number of candidates to a minimum. Many of these heuristics have proven effective with their high recall in extracting keyphrases from various sources.

Typical heuristics include using a stop word list to remove stop words [17], selecting just words with certain partof- speech tags (nouns, adjectives, verbs) to be candidate keywords[4][7][9], allowing n-grams that appear in Wikipedia article titles to be candidates [12], extracting n-grams [5].

In [5] three terms selection approaches were studied: n-grams; noun phrase (NP) chunks; and terms matching any of a set of part-of-speech (POS) tag sequences.

In the n-grams approach, the terms were defined in a manner similar to Turney[6], All unigrams, bigrams, and trigrams were extracted after or before a stopword removal.

A set of POS tag patterns is defined, and all words/phrases that belong to this set were extracted in the pattern approach, while in the chunking approach the research on term extraction focuses on noun patterns.

The experiments using a corpus of abstracts of scientific articles shows that the extraction of key terms from n-grams with the filtered words tools gives the best results among the three methods proposed.

2.2 Extraction of the keywords/keyphrases

In unsupervised context, to select the keywords/keyphrases, candidates are ordered in terms of their importance or the importance of words that compose them.

Because our approach is unsupervised, we focus merely on unsupervised techniques in this section.

TF-IDF

The most basic unsupervised method for keyphrase extraction is TF-IDF (Term Frequency – Inverse Document Frequency) [20].

Tf-Idf, is a metric for calculating the relevance of terms in documents, very used in Information Retrieval and Text-Mining. Essentially, this technique measures how important a certain word is on a document regarding other documents in the same collection.

Basically, a word gets more important in a certain document the more it occurs in that document. But if that word occurs in other documents, its importance decreases. Words that are very frequent on a single document tend to be more valued than common words that occur on more documents, like articles or prepositions.

Generally, the calculation of Tf-Idf is made in separate, calculating the Tf and Idf components separately, and finally multiplying both components to get the final Tf-Idf value. Tf component (term frequency) simply measures the number of times a word occurs on a certain document. That count is then normalized to prevent word on very long documents to get higher Tf values.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Equation 1 measures the probability that a term i occurs in a document j.

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad (1)$$

where n_{ij} is the number of times the term i occurs in a document j and then it is divided by the total of words in document j. Idf component measures the general relevance of a given term.

Equation 2 consists in the count of the number of documents that a term t_i occurs.

$$Idf_i = \log \frac{|D|}{|\{d_j / t_i \in d_j\}|} \quad (2)$$

where |D| represents the total number of documents in the collection and { : } j i d j d t ∈ the number of documents where the term ti occurs. Tf-idf (equation 3) is then the multiplication of the two previous equations.

$$TFIdf_{ij} = TF_{ij} \times Idf_{ij} \quad (3)$$

However, we must consider that the main goal of this method is to analyze the relevance of a word in a document regarding other documents, instead of analyzing the relevance of a word in corpora.

TF-IDF selects candidate keyphrases based on their statistical frequencies without considering semantic similarity between words. The inconvenient of this method is that it ignores the words which have low frequency.

The unsupervised methods to keyphrase extraction can be categorized into four groups: Topic-Based Clustering [7], Simultaneous Learning [8] [9], Language Modeling [10] and Graph-Based Ranking [4] [9] [11].

In this state of the art, we focus on the category of Graph-Based ranking methods.

Graph-Based Ranking

The main purpose of keyphrase extraction is to identify the important words and phrases from a document. In fact, a candidate is important if it is related to a large number of important candidates. Researchers have computed relatedness between candidates using co-occurrence counts [4] and semantic relatedness [12], and represented the relatedness information collected from a document as a graph [4][9][11].

Graph modeling is an alternative which clearly high-lights relationships of nodes among vertices. It also groups related information in a specific way, and a centrality algorithm can be applied to enhance their efficiency.

The basic idea behind a graph-based approach is to build a graph from the input document and rank its nodes according to their importance using a graph-based ranking method [13]. Each node of the graph corresponds to a candidate keyphrase from the document and an edge connects two related candidates.

The edge weight is proportional to the syntactic and/or semantic relevance between the connected candidates.

Graph representations of text and scoring function definition are two widely explored research topics, but few studies have been focused on graph-based IR in terms of both document representation and weighting models [15].

First, text is modeled as a graph where nodes represent words and edges represent relations between words, defined on the basis of any meaningful statistical or linguistic relation. The importance of a word within a document is estimated by the number of related words and their importance.

A node's score in the graph is defined recursively in terms of the edges it has and the scores of the neighboring nodes. The top-ranked candidates from the graph are then selected as keyphrases for the input document [14].

All methods follow that three steps:

1. Construction of the graph,

2. The calculation of node's score in the graph.

3. The extractions of the key terms.

These approaches differ from ours as they use graphs that are focused on the extraction of relevant words in a document and computing relations between words. In our proposal, a graph is built

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

such that the vertices are terms and the edges are relations between terms. Moreover, we focus especially on a scoring function of relevant terms in a domain rather than in a document.

TextRank

TextRank [4] is a graph-based approach to keyphrase extraction that was inspired by the algorithm Pagerank [13]. It is based on the calculation of the score of nodes by using the principle of vote or recommendation between two nodes. It performs the best for keyphrase extraction task[19].

TextRank first build a word graph according to a given document in which the connections between words depict their semantic connections, which are often computed by the word co-occurrences in the document. After that, we can get the score of each word to get the ranking candidate keyphrase by executing.

TextRank identifies connections between various entities in a text, and implements the concept of recommendation. A text unit recommends other related text units, and the strength of the recommendation is recursively computed based on the importance of the units making the recommendation.

The score for Vi , S(Vi), is initialized with a default value and is computed in an iterative manner until convergence using this recursive formula:

$$WS(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (4)$$

Where $In(V_i)$ denotes the set of vertices that point to V_i (predecessors), $Out(V_i)$ is the set of vertices that the vertex V_i points to (successors) and d is the damping factor set to 0.85[13].

Intuitively, a vertex will receive a high score if it has many high-scored neighbors. TextRank's best score is achieved and the best performance with the filter that selects nouns and adjectives only.

The main weakness of TextRank that it orders only words instead of ranking terms-candidates. Despite this weakness, TextRank remains one of the most well-known graph-based approaches to keyphrase extraction.

SingleRank

SingleRank is a modification of TextRank that weights the edges with the number of co-occurrences and no longer extracts keyphrases by assembling ranked words. Keyphrases are noun phrases extracted from the document and ranked according to the sum of the significance of the words they contain [9].

It is based on three steps: graph construction, calculation of summits and extractions of key terms.

Construction of the graph: the term-candidates can be composed and each word component each term-candidate is considered to be a vertex of the graph. Two vertices are connected by an edge if they co occurred in a window of N words and the weight of this edge is the number of co occurrence of these two vertices.

Calculating vertex scores: SingleRank uses a ranking algorithm to calculate vertex scores with the following formula:

$$S(C_i) = (1-\lambda) + \lambda \times \sum_{C_j \in In(C_i)} \frac{p_{ji}}{\sum_{C_k \in Out(C_j)} p_{jk}} S(C_j) \quad (5)$$

Where λ is an attenuation factor (can be considered as the probability that the node C_i is reached by recommendation); p_{ji} is the weight of the arc going from the node C_j to the node C_i, corresponding to the number of co occurrences between the two words i and j.

Candidate-term scores consisting of several words are calculated by:

$$\text{Termescore}(p_i) = \sum S(C_j) \quad (6)$$

Where p_i is a candidate term consisting of several words; C_j is one of the words composing the term-candidate p_i.

Extraction of key terms: These are the candidate terms with the highest scores which are retained as key terms. So there is no generation of key terms as is the case in TextRank.

In the majority of the cases, this approach gives better results than TextRank. The major inconvenient of this method is that it favors the composed terms as terms-candidats and makes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

appear redundant candidates after the ranking process.

TopicRank

TopicRank is an unsupervised graph-based method for keyphrase extraction.

Based on TextRank, TopicRank is different with regard to the other graph-based methods, because instead of focusing the selection on the important textual units of the document, it selects important subjects [11].

This method clusters the keyphrase candidates into topics, ranks these topics and extracts the most representative candidate for each of the best topics

It follows three steps:

1. Identification of the subjects,

2. The ranking of the subjects,

3. The selection of the keywords.

Compared with TextRank and SingleRank, it presents the advantage of improving the quality of the ranking. It has several advantages over TextRank.

Intuitively, ranking topics instead of words is a more straightforward way to identify the set of keyphrases that covers the main topics of a document. To do so, we simply select a keyphrase candidate from each of the top-ranked clusters.

Clustering keyphrase candidates into topics also eliminates redundancy while reinforcing edges. This is very important because the ranking performance strongly depends on the conciseness of the graph, as well as its ability to precisely represent semantic relations within a document.

Hence, another advantage of TextRank is the use of a complete graph that better captures the semantic relations between topics [11].

Construction of the graph: it is the candidate-words, composed of several words or not, which represent the vertices of the graph and all the vertices are connected to each other.

Identification of subjects: a subject is a specific (most often) or general information transported at least by a textual unit. Two candidate terms C_1 and C_2 are grouped from the Jaccard similarity:

$$Sim(C_1, C_2) = \frac{\|C_1 \cap C_2\|}{\|C_1 \cup C_2\|} \quad (7)$$

Where $C_1 \cap C_2$ represents the set of words common to C_1 and C_2 and $C_1 \cup C_2$ *is* the set of words composing C_1 and C_2.

As soon as the similarity between all pairs of candidate terms is known, the hierarchical ascending classification algorithm is applied. In the beginning, each term-candidate is considered a group and then both groups with the strongest similarity are united in one. This grouping is repeated until the (predefined) number of groups is reached.

The similarity between two groups is obtained by calculating the similarity between candidate terms of each group. The similarity value between two groups can be obtained by choosing from the following three methods:

- Simple: the highest value of similarity is retained;
- Complete: the smallest similarity value is retained;
- Average: the average of all similarities is retained;

Once the subjects are known, a new graph is defined as follows: G = (N,A): N = set of subjects of the document and A = set of links between the nodes.

Ranking of the subjects: the weighting of the edges is very important during this step. It is the strength of the semantic link [11] between the nodes of the graph which is considered as the weight of an edge.

To represent this semantic strength, the distance between the candidate-terms subjects is used.

$$\text{Poids}(S_i, S_j) = \sum_{c_i \in S_i} \sum_{c_j \in S_j} dist(C_i, C_j) \quad (8)$$

Where

$$dist(C_i, C_j) = \sum_{p_i \in pos(C_i)} \sum_{p_j \in pos(C_j)} \frac{1}{|p_i - p_j|} \quad (9)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and pos(C_i) and pos(C_j) are respectively the positions of the term candidate C_i and C_j in the document.

The ranking is based on the principle of voting or recommendation, it is to say that a node (subject) is very important if it is strongly connected with several other important nodes.

$$I(S_i) = (1-\lambda) + \lambda \times \sum_{S_j \in V_i} \frac{poids(S_i,S_j) \times I(S_j)}{\sum_{S_k \in V_j} poids(S_j,S_k)} \quad (10)$$

where V_i is the set of the subjects connected to the subject S_i and λ is the attenuation factor.

Selection of key terms: each important topic will only provide a single term. For the choice of a key word best representing a subject, three methods have been proposed:

- First position: the candidate term of a subject appearing first in the document is selected,

- Frequency: the term-candidate of a most frequent subject in the document is selected,

- Central: the term-candidate of a subject most similar to other candidates the same subject is selected.

Kcore

Kcore [15] is also a graph-based method of key-terms extraction. Unlike the graph-based methods presented above, it does not use the principle of vote or recommendation to calculate the nodes scores but uses the algorithm of Batagelj and Zaversnik [16].

The construction of the graph of words is similar to TextRank or SingleRank. The main inconvenience of the approach is that it depends on the window of co-occurrence of words.

Construction of the graph: The words candidates, composed of several words or not, represent the vertices of the graph and two vertices are considered to be connected if the candidate-terms representing these co-occurrent vertices in a window of N words.

When these connections are weighted by the number of co-occurrences of the words they connect in the document, we talk about WKcore, otherwise (in the case of a not weighted graph) they will all be weighted by 1. In this case, we talk about Kcore.

The degree of a vertex C $\in G$ in G is denoted deg_G (C). In other words, in the case where G is an undirected graph, deg_G(C) is the sum of the weights of edges adjacent to C (unit weight in the case of an unweighted graph).

Selection of key terms: Once the graph is built, a Kcore decomposition with the algorithm of Batagelj and Zaveršnik[17], of the graph is completed. The algorithm attribute a number n to each vertex of the Kcore graph to which it belongs (if v ∈ Kcore then n = k). Then the ranking of the vertices is done in descending order numbers n and the first ones are retained as key terms.

These approaches differ from ours as they use graphs that are focused on the extraction of relevant words in a document and computing relations between words. In our approach, we focus especially on a scoring function of relevant terms in a document of specialized domain rather than in a general document.

3. PRETOPOLOGY: BASIC CONCEPTS

Pretopology is an extension of topology that differs by the non idempotence of the adherency function which is a fundamental property for our purpose of following up the process of extracting keyphrases.
The fact that pretopology integrates fewer axioms makes it more adapted to discrete spaces [20].
We suppose in all that paper that E is a non empty finite space.

3.1 Pretopological adherence

Given a non empty set E, a function a(.) from P(E) into P(E) is called an adherence if and only if:

$$\forall A \in P(E), a(\emptyset) = 0 \quad (11)$$

And

$$\forall A \in P(E), a(A) \subset A \quad (12)$$

Then (E,a) is said a pretopological space.

According to properties of a(.), we obtain more or less complex pretopological spaces from the most general spaces to topological spaces. Pretopological spaces of V type are the most interesting case. In that case, a(.) fulfills the following property:

$$\forall A \in P(E), \forall B \in P(E), A \subset B \Rightarrow a(A) \subset A \quad (13)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2 Closed subsets, pretopological closure

Definitions

Closed subset: Lets (E,a) be a pretopological space. A subset A of E is said a closed subset if a (A) = A.

Pretopological closure: Given a pretopological space (E,a), for any subset a of E, we can consider the whole family of closed subsets of E which contain A. if exists, we determine the smallest element of that family for the including relationship. That element is called the closure of A and denoted F(A) .

In any pretopological space of type V, given a subset A of E, the closure of A always exists. In particular for each singleton {x} of P(E) [10].
If a(.) fulfils on the top of the properties (1), (2), (3) the following property:

$$\forall A \in P(E),\ \ a(A) = U_{x \in A}\{a(x)\} \qquad (14)$$

The pretopological space (E, a) is called a Vs - type space.

The Vs pretopological space (E,a) are very interesting since the pretopological closures of the set E will be made by the pretopological closures of its elements.

4. THE PROPOSED APPROACH

4.1 Selecting Candidate Keywords

As noted before, a set of phrases and words is typically extracted as candidate keyphrases using heuristic rules. These rules are used to keep the number of candidates to a minimum.

In the approach we propose, the following algorithm is used for selecting candidate words and phrases:

- First, a predefined list of stopwords is used to remove stop words [18].
- Allow just nouns, sequences of adjective-noun and sequences of noun-noun to be a potential candidate keywords/keyphrases .
- Select as candidate keywords the N potential candidate keywords the most frequent in the document, while N may be set to any fixed value, usually ranging from 2 to 10 keywords.

4.2 Graph Construction

To enable the application of graph-based ranking algorithms to natural language texts, we have to build a graph that represents the text, and interconnects words or other text entities with meaningful relations.

Depending on the application at hand, text units of various sizes and characteristics can be added as vertices in the graph, e.g. words, collocations, entire sentences, or others. Similarly, it is the application that dictates the type of relations that are used to draw connections between any two such vertices, e.g. lexical or semantic relations, contextual overlap, etc[4].

Regardless of the type and characteristics of the elements added to the graph, the application of graph-based ranking algorithms to natural language texts consists of the following main steps [4]:

1. Identify text units that best define the task at hand, and add them as vertices in the graph.

2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.

3. Iterate the graph-based ranking algorithm until convergence.

4. Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

In this first step, documents are represented as graph. An undirected word graph is constructed for each document in a document corpus, in which each of words of the document is represented as nodes. The co-occurrence relations between words of the documents are represented as edges.

Nodes of graph are filtered by syntactic filters. Edges are weighted according to the co-occurrence count of the words they connect.

The expected end result for this application is a set of words or phrases that are representative for a given natural language text. The units to be ranked are therefore sequences of one or more lexical units extracted from text, and these represent the vertices that are added to the text graph. Any relation that can be defined between two lexical units is a potentially useful connection (edge) that can be added between two such vertices.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We are using a co-occurrence relation, controlled by the distance between word occurrences: two vertices are connected if their corresponding lexical units co-occur within a window of maximum N words, where N can be set anywhere from 2 to 10 words.

Co-occurrence links express relations between syntactic elements, and similar to the semantic links found useful for the task of word sense disambiguation [4], they represent cohesion indicators for a given text.

Moreover, we apply the hypothesis that the term representativeness in a graph, for a specific-domain, depends on the number of neighbors that it has, and the number of neighbors of its neighbors. We assume that a term with more neighbors is less representative of the specific-domain. This means that this term is used in the general domain.

4.3 Ranking Candidate Words and Phrases

This approach aims to select from the candidate terms the more representative for the document. As mentioned above, the graph is built with a list of candidate terms obtained according to the steps described before, where vertices denote terms linked by their co-occurrence in the sentences in the corpus.

To rank the candidates terms, we use a new measure that calculates term weights based on the pretopological adherence. It consists on awarding scores to candidates based on the number of neighbors he has.

The intuition for this new approach is as follows: the more a term T has neighbors directly or by transitivity, the more the weight decreases. Indeed, a term T having a lot of neighbors is considered too general for the domain, and then it has to be penalized via the associated score.

Definition of the pretopological space

Formally, a graph is an ordered pair G = (E, V) where V is the set of vertices and V□ExE is the set of edges.

The key problem is to detect the keywords in the graph constructed from the document by defining a pretopological structure of Vs type on E.

The choice of a pretopological method is explained by the fact that the adherency application a(.) is non idempotent and its successive aggregations lead to produce the closures which characterize homogenous or connected parts of E.

This fundamental property will allow us to define an iterative process that takes into account global information recursively computed from the entire graph, rather than relying only on local importance. Following up this iterative process will led to detect the closures that represent the subsets with a strong neighborhood relationship.

We define an adherence on E by:

$$a(x) = \begin{cases} \{y \in V(x)\} & si\ |V(x)| > K \\ \{x\} & si\ |V(x)| \leq K \end{cases} \quad (15)$$

Where

$$V(x) = \{y \in G\ /\ dis(x,y) > K\} \quad (16)$$

and $dis((x,y)$ designates the the number the vertices between the two nodes x and y, and K is an integer number defined by the tuning algorithm procedure.

We put:

$$a(A) = \cup_{x \in A}\{a(x)\} \quad \forall A \in P(E) \quad (17)$$

(E,a) is a Vs type pretopological space by definition.

As a result, the pretopological closures of a set E will be made by the pretopological closures of its elements [8].

The ranking algorithm

We define the score of the vertices T as:

$$Score(T) = \log_2(1.5 + \frac{1}{F(T) + \sum_{t \in A} F(t)}) \quad (18)$$

Where

$$A = \{t \in G\ /\ T \in F(t)\} \quad (19)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We search the pretopological closure of a vertice T , F (T) following this process:

F(T) = a(T);

While (a(F (T) ≠ F (T));

F(T) = a(F(T))

The ranking process of candidate words and phrases relies on these steps:

1. *Construct first the graph from candidate terms (unweighted and undirected graph).*
2. *Calculate for each vertice T the Score(T).*
3. *Classify the all T in descending order of Score(T).*
4. *The candidate terms with the largest scores are selected as keywords.*

The tuning procedure

The adjustment of the parameter K depends both on the graph size N of. The keywords detection procedure is applied with some values of K; mode stability is then satisfied in the domain resulting from the largest ranges of K where the number of extracted keywords remains constant.

Choosing the parameter K in the middle of the largest range of K belonging to this domain of mode stability has proved to be a good procedure to optimize the proposed keyphrases extraction approach [23][24].

Example

For the purpose of illustration we reproduce the abstract from Inspec collection used by Mihalicia and al displayed in Figure 2.

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

Figure 2: The abstract from Inspec collection used in the example

Method adopted	Keywords extracted
our approach	Linear constraints; linear equations; strict inequations; nonstrict inequations;Solutions; linear systems, natural numbers,compatibility
human annotators	linear constraints; linear diophantine equations; minimal generating sets; strict inequations; set of natural numbers; nonstrict inequations; upper bounds
TextRank	linear constraints; linear Diophantine equations;natural numbers; nonstrict inequations; strict inequations; upper bounds

For this example, the keyphrases extracted by our approach are displayed in Table 1 and for a comparison purpose we give also the keyphrases extracted by TextRank algorithm and by human experts.

Table 1: Results for keyword extraction applying the new approach to the abstract of the example

It appears from the results that the recall, the precision measure of our approach remains very close of the other approaches which prove that our method outperforms the baseline approaches significantly.

5. CONCLUSION

Keyphrases Extraction is widely used in text process such as clustering, classification or information retrieval. Keyphrases should express the core content of the document. In this research paper an attempt has been made to extract keywords

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

using graph method that is taking into account global information recursively computed from the entire graph.

In this method, initially documents are represented as graphs, and then keywords are determined from the graph using a ranking process.

The algorithm evaluates the importance of the graph nodes (node-to-node) and accordingly an overall score (node-to-whole) is calculated. The extraction process undergoes semantic relevance recursion to properly define the keywords/keyphrases that represent the best the document.

We showed that pretopology, despite being conceptually the simplest measure, achieves results comparable to the widely used TextRank algorithm. Moreover, results show that closeness significantly outperforms the other graph-based measures on short documents.

In this paper, and as done by Hulth in [5], we simply treated the manually assigned keywords as the gold standard judging that is the most severe way to evaluate a keyword. Future work will examine alternative approaches to evaluation.

REFRENCES

[1] Peter Turney, "Extraction of Keyphrases from Text: Evaluation of Four Algorithms", National Research Council of Canada, Canada, October 23, 1997

[2] J.Naveenkumar, "Keyword Extraction through Applying Rules of Association and Threshold Values", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), ISSN: 2278–1021, Vol. 1, Issue 5, pp 295-297, July 2012

[3] Jasmeen Kaur and Vishal Gupta, "Effective Approaches for Extraction of Keywords", International Journal of Computer Science Issues (IJCSI), ISSN (Online): 1694-0814, Vol. 7, Issue 6, pp 144-148, November 2010

[4] R. Mihalcea and P.Tarau, "TextRank : Bringing Order Into Texts", in Dekang Lin, Dekai Wu(eds), Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP),Association for Computational Linguistics, Barcelona, Spain, p. 404- 411, July, 2004

[5] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge", Proceedings of the 2003 Conference on EmpiricalMethods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Stroudsburg, PA, USA, p. 216-223, 2003

[6] P.D Turney, "Learning Algorithms for Keyphrase Extraction", Information Retrieval, vol. 2,no 4, p. 303-336, may, 2000

[7] Z. Liu, P.Li, Y.Zheng, M. Sun, "Clustering to Find Exemplar Terms for Keyphrase Extraction", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1 (EMNLP), Association for Computational Linguistics, Stroudsburg, PA, USA, p. 257-266, 2009

[8] H. Zha, "Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering", In Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 113–120, 2002

[9] X. Wan, J.Xiao , "Single Document Keyphrase Extraction Using Neighborhood Knowledge", Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI Press, p. 855-860, 2008.

[10] Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In Proceedings of the ACL Workshop on Multiword Expressions, pages 33–40.

[11] Adrien Bougouin, Florian Boudin, and B´eatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In Proceedings of the 6th International Joint Conference on Natural Language Processing, pages 543–551.

[12] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In Proceedings of the 18th International Conference on World Wide Web, pages 661–670.

[13] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. Computer Networks, 30(1–7):107–117.

[14] Hasan Kazi Saidul, Ng,Vincent. Automatic keyphrase extraction. 2014. A survey of the state of the art. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference. Vol. 1 Association for Computational Linguistics (ACL), 2014. p. 1262-1273.

[15] Rousseau F., Vazirgiannis M., Main core retention on graph-of-words for singledocument

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

keyword extraction, In Advances in Information Retrieval, 2015, p. 382–393.

[16] Batagelj V., Zaveršnik M., Fast algorithms for determining (generalized) core groups in social networks, Advances in Data Analysis and Classification, vol. 5, n° 2, 2011, p. 129-145.

[17] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 257–266.

[18] Peter Turney. 1999. Learning to extract keyphrases from text. National Research Council Canada, Institute for Information Technology, Technical Report ERB-1057.

[19] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 366–376.(2010).

[20] Gerard Salton and Christopher Buckley. Termweighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513–523)(1988).

[21] Kim S. N.,Medelyan O., KanM.-Y., BaldwinT., « SemEval-2010 task 5 : Automatic Keyphrase Extraction from Scientific Articles », Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval), Association for Computational Linguistics, Stroudsburg, PA, USA, p. 21-26, 2010.