Unsupervised Key-Phrases Extraction from Scientific Papers Using Domain and Linguistic Knowledge

Mikalai Krapivin[1], Maurizio Marchese[1], Andrei Yadrantsau[3], Yanchun Liang[2]
[1]*Department of Information Engineering and Computer Science, University of Trento, Italy*
{last name}@disi.unitn.it,
[2]*College of Computer Science and Technology, Jilin University, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education*
ycliang@jlu.edu.cn
[3]*Department of Computer Science, Belarussian State University,Minsk,Belarus*
{first name}.{last name}@gmail.com

Abstract

The domain of Digital Libraries presents specific challenges for unsupervised information extraction to support both the automatic classification of documents and the enhancement of users' navigation in the digital content. In this paper, we propose a combined use of machine learning techniques (i.e. Support Vector Machines) and Natural Language Processing techniques (i.e. Stanford NLP parser) to tackle the problem of unsupervised key-phrases extraction from scientific papers. The proposed method strongly depends on the robust structural properties of a scientific paper as well as on the lexical knowledge that we are able to mine from its text. For the experimental assessment we have use a subset of ACM[1] papers in the Computer Science domain containing 400 documents. Preliminary evaluation of the approach shows promising result that improves – on the same data-set – on state-of-the-art Bayesian learning system KEA[2] from a minimum 27% to a maximum 77% depending on KEA parameters tuning and specific evaluation set. Our assessment is performed by comparison with key-phrases assigned by human experts in the specific domain and freely available through ACM portal[3].

1. Introduction

Machine learning methods are commonly and successfully used for information retrieval (IR) tasks. The large majority of research work in IR domain is dedicated to the extraction of information from web pages, mails, news and typically short and unstructured type of digital content (see for instance [1]). A specific challenge for unsupervised information extraction lies in the domain of scholarly papers [2]. This challenge is related to the development of autonomous digital libraries in academia domain [3]. Spider systems like Citeseer[4], Google Scholar[5] or Rexa[6] crawl the web seeking for scientific papers. Crawled papers are available in the Web (either freely, as self-published articles, or in commercial digital libraries) and the extracted metadata information can help to categorize the documents while simplifying and enhancing users' searches. After crawling and retrieving, usually papers are in PDF and PS format. Thus they are first parsed and converted to a text format. Then relevant metadata (e.g. title, authors, citations, etc.) are extracted and finally documents are properly analyzed (e.g. ranked, classified, etc). With the exponential growth of the quantity of available information (millions of scientific journal papers, proceeding, workshops or book chapters per year), documents' processing cannot be done any longer manually. To achieve unsupervised (or at least semi-unsupervised) information extraction, classification and categorization processes, machine learning techniques are often used. The task of information extraction from scholarly papers can be separated into two broad cases:

(i) recognition of information which is *structurally* present inside the body of the scientific paper (e.g. authors, mails, institutions, venues, title of a paper, keywords and/or key-phrases assigned by authors, abstract *etc.*);

(ii) extraction of information which is *implicitly* present in the paper but there is no guarantee that (this information) is located inside a document: for example the extraction of a number of generic topics /categories - not explicitly assigned by the authors - from a full text of a document.

In this paper, we focus on the second, more challenging type of information extraction, i.e. extraction of *implicit information*. In particular, we present some

[1] http://www.acm.org/
[2] http://www.nzdl.org/Kea/index_old.html
[3] http://portal.acm.org; available also at http://dit.unitn.it/~krapivin/
[4] http://citeseer.ittc.ku.edu/
[5] http://scholar.google.com
[6] http://rexa.info/

978-1-4244-2917-2/08/$25.00 ©2008 IEEE

novel use of Support Vector Machine (SVM) training method in combination with the Stanford Natural Language Processing (NLP) parser for the unsupervised key-phrases extraction from scientific documents in Computer Science domain. The Stanford NLP parser is able to define not only part of speech like verb or noun, but also more deep relations between tokens like *noun phrase*, *verb phrase* etc. [4]. Another conceptual contribution of the present paper is the treatment of each document text not just as a bag of words, as normally done in state-of-the-art key-phrases extraction system like KEA (see Section 2 for more details on KEA). In fact, we use both the structure of the document, and the fact that every word may be a different part of speech in different contexts - which is successfully captured by Stanford lexical NLP parser - and represented as a *label* following the approach proposed in the Penn Treebank Project [5]. In this approach, skeletal parses, i.e. a bank of linguistic labels and syntactic relations, are produced capturing rough syntactic information.

In the following, we present in Section 2 a brief review of the state of the art in the domain and a discussion of relevant related work. Section 3 provides a detailed description of the dataset used in our experiments. The entire used dataset has been collected (crawled from the web, mainly from Citeseer) and processed (converted from PDF to text, parsed and later analyzed) in a completely autonomous (unsupervised) manner. We are sharing the prepared dataset through the Internet[7], and we welcome the interested communities to use it as benchmarking set to test different information extraction approaches. Section 4 presents the details of the proposed extraction methodology, specific feature set and text processing tasks. We present our preliminary but encouraging results in Section 5 and in Section 6 we compare them with the state-of-the-art system KEA [6]. Section 7 is devoted to the conclusions and discussion of future work.

2. Related Work

In the case of recognition of information which is *structurally* present inside the body of a scientific paper, most information is - with high probability - situated in the top part of each document (header section, first page, etc.). Moreover, there are typically a limited number of patterns for describing such information, i.e. templates of conferences of journals. So the corpus for such investigations is typically a set of headers for a given set of documents. A corpus of this type was firstly proposed by Seymore, et al. [7] and is freely available on the web[8]. One of the best methods for such specific pattern recognition and information extraction is Support Vector Machines (SVM) training. It was used in [2] and provided very good results for this kind (i.e. pattern-based) information extraction. Using the standard measures of performance of information extraction, i.e. Precision, Recall and F-Measure (for the exact definition please see Section 4.3), the authors reported Precision up to 95% with corresponding Recall of 79% and F-Measure of 86%.

Also Hidden Markov Models (HMM) usage could give precision comparable with SVM for the same categories of extracted metadata, as pointed out by Seymore *et al*. Moreover, in [8], they proposed Conditional Random Field (CRF) technique for the information extraction. In brief, CRF is a probabilistic model for segmenting and labeling sequence data. This work reported F-Measure close to 93%, which outperforms all the previous results. All experiments in [7], were performed under the same public dataset[8] composed of 500 training headers and 5000 testing headers of scientific papers. The above-mentioned results are so far the best in the domain. All approaches above [2], [7], [8] focus on the *recognition* of information that is *structurally* inside the document header's text.

A different and challenging task is the information *extraction*, where the goal is to extract information which is implicit. For example, when we need to extract the keywords/key-phrases from a full text of a document and we do not know where and even if they are inside a document or not. To tackle this kind of problem, we can use methods like the one implemented in the system Key-phrase Extraction Algorithm (KEA) [6]: KEA uses a classifier based on Bayes' theorem to classify words of documents (preferably from the same domain) as keywords/key-phrases or not. The most recent results reported by KEA team show 13% for Precision and 12% for Recall [9] in the assignment of key-phrases to generic web pages. This result is very similar to be a state of the art in unsupervised statistical TFxIDF-based key-phrases extraction. Usage of domain specific and *controlled vocabularies* may improve the result up 28.3% for Recall and 26.1% for Precision as reported in [9].

Another interesting approach has been suggested by Tourney using GenEx algorithm [10]. GenEx is based on a combination of parametrized heuristic rules and genetic algorithms. The approach provides nearly the same precision and recall as KEA. In a more recent work [1], the author applies web-querying techniques to get additional information from the Internet as background knowledge to improve the results. This method has some disadvantage, since web-mining for information and parsing of responses is a heavy operation, which is inconvenient for digital libraries, where we have to deal with a large (millions) number of documents. Moreover, the authors measure the results through the *average* number of correctly found phrases *vs.* total number of

[7] http://portal.acm.org, dataset is available at http://dit.unitn.it/~krapivin/
[8] http://www.cs.umass.edu/~mccallum/

extracted phrases. In this approach Precision varies from 0% to 25% with unknown recall.

In the recent works A. Hulth et al. took into account domain [11] and linguistic [12] knowledge in the search of the relevant key-phrases. In particular [11] used thesaurus trying to get domain knowledge. Recall reported in this work is very low, namely just 4-6%. Paper [12] introduced heuristic related to part-of-speech usage, and proposed training based on 3 standard KEA features plus one linguistic feature. Authors reported relatively good results (F-Measure up to 33.9%). However, it is hard to compare them with others results due to the strong specificity of the used data set: short abstract with in average 120 tokens where around 10% of all words in the proposed set were key-phrases.

A recent interesting work in regard to the application of linguistic knowledge to the specific problem is reported in [13]. The authors used WordNet[9] and "lexical chains", structures based on synonyms and antonyms notions. Then, they applied decision trees as a machine learning part and applied it on ca. 50 documents (journal articles) as the training set and 25 documents as the testing set. They reported interesting high precision - up to 45% of recognition - but without any mention about recall, which makes complex any comparisons with other results.

Other machine learning technique, i.e. least square Support Vector Machine [14] produces interesting results. Namely 21.0% for Precision and 23.7% for Recall in the analysis of web mined scientific papers. However, also in this case the described experiments are limited to a relative small testing dataset, namely just 40 *manually collected* scientific papers.

First steps towards the *assignment* of key-phrases to the documents that do not contain any key-phrase inside were performed by Turney in [15]. At present, this cannot be done purely with the help of instance learning methods. To make key-phrases assignment feasible in this challenging case, Turney proposed to find semantic relationships between phrases, which may improve the performance of key phrases recognition. But in this case there is a big question about the validity of the result. If a key-phrase is not assigned by an author or expert, who is able to state that it is indeed a key-phrase?

Many of the techniques presented above, may be applied in combination with other ones, for example with our proposed one, in order to enhance the final result.

3. Dataset description

In our research, we have used a set of scientific papers in the computer science domain and presented in the ACM portal. Full texts for the dataset were crawled automatically by Citeseer autonomous digital library. For the purpose of our investigations (i.e. key-phrases extraction from digital scientific papers), we need to separate two different categories of key-phrases:

1. Assigned by author(s) and located inside each document before the abstract and after the prefix "Keywords:"
2. Editor/Reviewer assigned key-phrases: manually assigned or revised with high quality by human experts in a particular domain.

From a general point of view, both types of key-phrases are of interest.

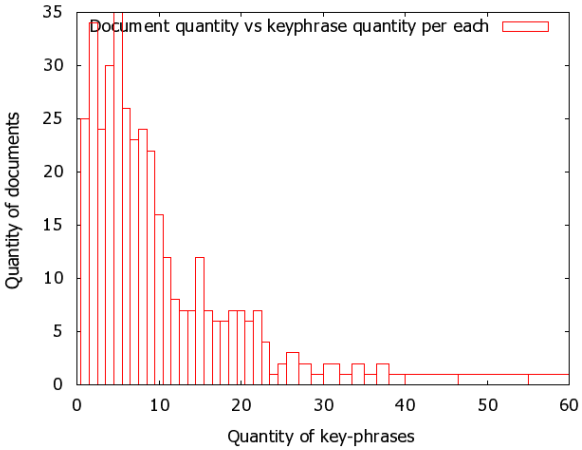

Figure 1. Distribution of human assigned key-phrases per document.

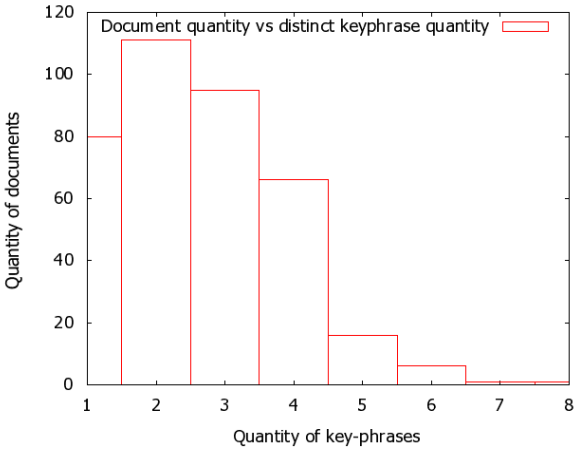

Figure 2. Distribution of *unique* key-phrases present per document

Herein, we have focused only on the (more challenging) second type of key-phrases. It is interesting to note, that in our collection of crawled documents, 95% do not contain author's assigned keywords. For the purpose of our investigation, we have removed the author's assigned keywords in the remaining 5% to obtain a uniform dataset.

Our final dataset contains 400 documents with editor assigned key-phrases. It is important to note that we have selected the above dataset so that *all* of the selected

[9] http://wordnet.princeton.edu/

documents contain *at least* one of the experts' assigned key-phrase in the full text of a document. All of the considered documents are published in the 2002-2005 period. On average a document has around 7 assigned key-phrases. Figure 1 shows the distribution including all possible key-phrases repetitions.

We have also directly analyzed our set for the distribution of existing key-phrases inside the full text of each document. The distribution of unique found key-phrases in the dataset is shown in Figure 2. Table 1 summarizes some relevant statistic on key-phrases in the dataset. As expected, the average number of key-phrases in a document is much lower than the number of all other words and phrases.

Table 1. Dataset statistics: average key-phrases count per document

The measure description	Value
Average quantity of *existing* key-phrases per one document	9.5
Average quantity of *existing unique* key-phrases per one document	2.6
Average quantity of words per one document (excluding stopwords)	3256

Thus, the selected dataset is strongly unbalanced in the ratio between key-phrases and general phrases present in the document. This unbalance is a realistic characteristic for a standard set of documents in a digital library.

4. Experimental methodology

4.1 Preprocessing

Before any extraction task, document texts need to be preprocessed, in order to assure a reasonable quality of the extraction [16]. Preprocessing includes *tokenization*, *refinement*, *stemming*, and recognizing separate blocks inside the article, i.e. Title, Abstract, Section Headers, Reference Section, Body etc.

Tokenization. In the present work we have used the tokenizer embedded into the Stanford NLP suite [4] with some additional heuristics. After recognizing the *tokens* and the *sentences*, we build phrases based upon them. For example from the sentence "*general establishment scheme can perform fast*" our phrase tokenizer will make 8 key-phrase candidates, namely:

"*general*",
"*general establishment*",
"*general establishment scheme*",
"*establishment*",
"*establishment scheme*",
"*scheme*"
"*perform fast*",
"fast".

There can be several heuristics behind key-phrases construction. In our approach, we have considered and applied the following two simple heuristics:

1. In a key-phrase two tokens cannot be separated by a verb in the case the verb is included in the stopword list (please see the above sample with stopword modal verb "can").
2. The end of a sentence indication symbol cannot separate a key-phrase. Usually such symbols are ".","!", "?"). This means that the parts of a key-phrase cannot be located simultaneously in two sentences.

Refinement. Stopwords that do not carry relevant meaning (for example "a", "the", "also", "as", "at", but also specific verbs like "can", "do", "could", "would" *etc*) are not allowed to participate in phrase construction. Moreover, all Unicode symbols except Latin letters, "-" and "'" cannot participate in phrase construction. We have used the list of English language stopwords compiled in the WEKA system [17].

Stemming. To avoid stemming issues – same word written in different forms - we have also used snowball stemmer[10] (which is embedded into KEA [6]).

4.2 Selection of the features set for SVM training

The central and most critical step of all information extraction approaches is the selection of a proper feature space. There are a large number of possible features that may be used for accurate information extraction. Moreover, their characteristics are strongly domain dependent. For example in Turney's work on the extraction of key-phrases from generic text [10], he proposed the following feature set:

I) Number of words in the phrase,
II) Frequency of key-phrase occurrence,
III) First occurrence of a key-phrase or position in text,
IV) Term frequency of each of the words in a phrase,
V) Relative length of a phrase,
VI) Noun, adjective, verb presence in a phrase.

This standard set may be enlarged as is done in Han et al [2], by considering also:

I) The number of line where a key-phrase is found,
II) Features related to the presence of a word in a dictionary,
III) The content of each line, quantity of digits in a line etc.

In our approach, we propose the following feature set, as detailed in Table 2. Features 1) and 2) are common

[10] http://snowball.tartarus.org/

ones and they are widely used in most information extraction systems. Less traditional features that we used are: 3) the length of a phrase, used in [2], and 4), the part of a text, successfully used in [14]. The most important features, are numbered in Table 2 from 5) to 10). They represent the linguistic knowledge about each token in a phrase. 5), 7) and 9) are labels assigned by POS (part-of-speech) tagger to each of the tokens in a phrase. Stanford NLP Parser [4] uses Penn Treebank [5] label definitions.

Penn Treebank is a large storage built upon 4.5 millions of American English words.

Table 2. Proposed Feature Set for SVM+NLP approach

#	Feature	Short description
1	TF	Term frequency
2	IDF	Inverted document frequency
3	LENGTH	Quantity of symbols in a phrase
4	PART-OF-TEXT	Part of a document where phrase was found (i.e. title, reference, body or section header).
5	1-TOKEN-LABEL	Penn Treebank label of the *first* token in a phrase.
6	1-TOKEN-DEPTH	Depth of the *first* token in a sentence, constructed by Stanford NLP parser.
7	2-TOKEN-LABEL	Penn Treebank label of the *second* token in a phrase.
8	2-TOKEN-DEPTH	Depth of the *second* token in a sentence, constructed by Stanford NLP parser.
9	3-TOKEN-LABEL	Penn Treebank label of the *third* token in a phrase.
10	3-TOKEN-DEPTH	Depth of the *third* token in a sentence, constructed by NLP parser.

Each of those words may be *tagged* as a different part of speech by 36 possible tags. Beside POS tagging Stanford NLP parser provides linguistic relationships between parts of a *sentence*. So each sentence may be represented as a tree, where each token inside may have tree-based depth. Such depth is captured as a feature in Table 2 (see features 6), 8), 10)). If a phrase has just one token, features 7)-10) are equal to zero (that implicitly captures the quantity of tokens in a phrase).

According to Table 2, our final features' set contains 10 features. The proposed features set was based on the following heuristic: we seek for key-phrases in the title, references and sections headers parts only. On the other hand, we compute TF and IDF features using the *full text*. The above simplification was done to limit computational complexity: in our case a paper has, on average ca. 3000 words and it may be split into ca. 9000 phrases. Just for only 100 documents, the number of features will grow up drastically to a million of vectors. It is very important to note, that we *do not* consider a paper's text as a *bag of words*, since each of the words may occur in several parts of a document as a different part of speech, for example as a *noun* or as a *verb*, or as a part of a different lexical structure, for example VP (verb phrase) or PP (prepositional phrase) *etc*., [5].

4.3 Result's assessment methodology

There are 3 main measures of performance of information extraction: I) Precision, II) Recall, III) F-Measure. Let us define 3 sets of extracted information. Set A is the key-phrases that are not recognized as key-phrases. Set B is the correctly recognized key-phrases, and set C is the phrases incorrectly recognized as key-phrases. So according to the definition of A, B and C the precision P, the recall R and the F-measure F can be defined as follows:

$$\mathrm{P} = 100\% \cdot B/(B+C) \quad (1)$$

$$\mathrm{R} = 100\% \cdot B/(B+A) \quad (2)$$

$$\mathrm{F} = 2 \cdot PR/(P+R) \quad (3)$$

In the following, we will use the above-defined values to determine and compare the performance of our method. It is important to note, that our assessment is performed comparing the set of extracted key-phrases per document with the set assigned by human experts for the same document (i.e. these are available through ACM portal). Moreover, we consider P, R, F just for *unique* key-phrases occurrence. For example if a recognized key-phrase "web service" occurs in a text 10 times, it would be taken into account only once when computing P, R, F values.

5. SVM+NLP method

In order to assess our proposed approach, hereafter referred to as SVM+NLP method, we have prepared and run several experiments. We have separated the selected dataset (described in Section 3) into 4 equal sets: I) Training Set (TRS), II) Test Set (TS), III) Evaluation Set 1 (ES1) and IV) Evaluation set 2 (ES2). Training set (TRS) and testing set (TS) are used for tuning SVM parameters and for 2-folder cross-validation. Then independent evaluation on the remaining evaluations sets (ES1) and (ES2) have been performed, in order to obtain a first rough estimate of the dispersion of our results. As a selection criteria for the all sets we have chosen the year of publication, thus (TRS), (TS), (ES1), (ES2) have the same quantity of papers published in a given year x.

For the implementation of the SVM, we use LIBSVM [18].

All of datasets (TRS), (TS), (ES1), (ES2) contain exactly 100 of documents and approximately 45000-50000 of constructed vectors per each. For our specific

non-linear extraction problem[11], we have used the Gaussian *RBF* kernel [18] in the form: $K(x,x') = \exp(-\gamma \| x - x' \|^2)$.

Therefore, the parameters for tuning are: γ and *C*. Both tuning ranges are identical, i.e. $\gamma, C \in \{2^{-3},..,2^{10}\}$. The result of the tuning phase for 2-folder cross-validation has provided as optimal values: γ=16, *C*=34.

For the evaluation experiments, we have used the remaining documents in the datasets (ES1), (ES2) which were initially separated from the training sets. In order to improve the analysis of our approach (selected feature set, use of domain and linguistic knowledge) and to better access the method's performance, we have performed a series of runs.

In the first series we have taken into account all the possible key-phrases from 1 to 3 tokens generated by the SVM. We focus on this range of token's length because in our datasets the majority (ca. 91%) of the assigned key-phases falls in this category. Moreover, this is also the threshold for tokens per extracted phrases used in KEA algorithm. This is our baseline experiment and will be compared with KEA approaches in the next section. The results of the proposed SVM+NLP approach are collected in Table 3.

Table 3. SVM+NLP key-phrase extraction for key-phrases from 1 to 3 tokens

	ES1	**ES2**
Precision:	19.9.%	20.2%
Recall:	19.8%	18.8%
F-Measure:	**19.8%**	**19.5%**

In order to explore which part of the proposed SVM+NLP approach has the biggest impact on the final key-phrase recognition precision and recall, we have carried out some more specific experiment. In particular, by omitting the main NLP features (i.e. features listed from 5) to 10) in Table 2), we have obtained significantly poorer results, namely P~10.8%, R~7.4%, therefore measuring the relevant impact of the proposed usage of the linguistic knowledge embedded in the selected linguistic features.

In other experiments, we search for key-phrases in the complete full text (not only in title, references and sections headers, like in the proposed approach). In these experiments, we observed improvements in the Precision but decrease in the Recall and in the F-Measure (ca. 5% for F). We believe that this data supports our heuristic about narrowing the search area to specific parts of the document while maintaining the computation of TF and IDF features using full text.

[11] Initial tests with a linear kernel provided, as expected, very low Precision and Recall, ca. 1-2 %.

The last series of experiments is focused on the extraction of the most relevant key-phrases. Key-phrases longer than 1-token are usually more descriptive. If we restrict our extraction to 2 and 3-token key-phrases, the results are: **26.3%** for F-measure for ES1 and **26.8%** for F-measure for ES2. We believe that this important improvement reflects that some one-token phrases present in our dataset - and assigned by humans – are very generic and do not carry interesting meaning. For example, we have the following keywords: "scheme", "layers", "delay", "fault" etc. that humans *assigned* as key-phrases in our dataset. Without the use of domain specific and controlled vocabularies, these single token key-phrases are not distinguishable from other more relevant keywords. So we expect, as in the case of KEA [9], that the use of a controlled vocabulary will also increase the performance of our approach.

6. KEA training

In order to compare our approach with a state-of-the-art key-phrase extraction system we have selected KEA [6]. KEA is relatively simple and we have used it as a black box with limited parameters tuning. The goal here is to perform training and evaluation on the same sets of documents as we have used for SVM+NLP learning.

KEA training includes two initial steps. I) All full texts should be placed into the files named "id.txt". II) All according key-phrases should be put into files named "id.key". All texts and key-phrases are stored in UTF-8 encoding. We use the same training set (TRS) for training and the same (ES1) and (ES2) sets for evaluation as well as the same results' assessment methodology as described in Section 4.3.

It is important to note that KEA algorithm uses a threshold parameter *q* to define the maximum number of extracted key-phrases. Results for KEA extraction for different values of the threshold parameter are reported in Table 4.

The results for the default value *q=5* show relatively low Precision, but rather high Recall. The overall F-Measure is a relatively good result- in line with general KEA extraction results from generic web pages [9] - but however lower than the SVM-NLP results.

Decreasing the parameter (*q*=3) improves the overall results: higher precision, lower recall but better overall F-measure. However this trend does not continue by an even smaller threshold of *q*=2.

Table 4. Summary results for KEA training for *q* phrases per document

# phrases threshold, *q*		ES1	ES2
5 (default)	**Precision:**	7.47%	9.1%
	Recall:	30.8%	38.5%
	F-Measure:	**12.0**%	**14.7**%
3	**Precision:**	10.7%	10.7%
	Recall:	26.6%	27.4.9%
	F-Measure:	**15.26**%	**15.34**%
2	**Precision:**	9.1%	11.5%
	Recall:	15.0%	19.7%
	F-Measure:	**11.2**%	**14.5**%

KEA results present the following critical characteristics:

1. Precision and Recall have a high dispersion (from 65% up to 323%) for all investigated thresholds and evaluation sets, while the proposed SVM+NLP approach presents more uniform behavior (see Table 3);
2. all computed values (P, R and F) are set dependant: in Table 4 we see up to ca. 29% deviation in F-measure in the two sets; in comparison the SVM+NLP approach is more stable (deviation less than 1,5%);
3. the proposed SVM+NLP approach always outperforms KEA recognition performance: using the F-measure as a good comparison parameter the proposed approach improves – on the same dataset – on state-of-the-art Bayesian learning system KEA from a minimum 27% to a maximum 77% depending on the selected threshold and evaluation set.

On the other hand, KEA maintains some positive characteristics, namely:

1. relatively faster overall computation speed and reduced time for data pre-processing;
2. a probability-based approach to extract key-phrases, which allows the tuning of the threshold to improve the overall results.

7. Conclusion and Future Work

In this paper we have described and analyzed a novel information extraction method, that we have named SVM+NLP – based on a combination of a specific learning method (SVM), distinct kernel (non linear RBF kernel), linguistic knowledge obtained using the Stanford NLP Parser, and characteristic feature set – aiming at capturing both the specific (scientific) document structure and the mined linguistic knowledge. The proposed method shows promising results on completely unsupervised key-phrases extraction from scientific papers. We do believe it may be a basis for an efficient and precise unsupervised key-phase extraction system; a system much needed for in the management of digital content in entirely autonomous digital libraries.

We have performed a detailed evaluation of the performance of the proposed method by comparison with human assigned key-phrases. The evaluation shows good precision of extraction (in the range ~20-27%) with nearly the same Recall and therefore good overall F-Measure. Moreover, we have compared our results with other approach i.e. KEA, for the same dataset and assessment methodology. The proposed SVM+NLP approach improves on state-of-the-art Bayesian learning system KEA from a minimum 27 % to a maximum 77% depending on the selected threshold and evaluation set

One limitation of the present work (and of all the works based on the *instance learning*) is in the assumption of the presence of the searched key-phrases inside the documents (assumption that has been used in the construction of our dataset). Indeed, our learning method cannot find (without additional supporting knowledge) a specific key-phrase in a document when the document does not contain at least one instance of the key-phrase. To tackle also such challenging key-phrase *assignment* task one needs to take into account documents or key-phrases similarities. For example one may forecast that documents with similar topic may have similar key-phrases. Alternatively, we have to move from syntactic to semantic relations between words in order to access (implicitly) related key-phrases.

We believe that the proposed hybrid (SVM+NLP) approach may be also valid with different and more specific datasets like emails, news, abstracts, web pages *etc*. The validation of this assumption will be our immediate future work. The other main direction of work is the use of the unsupervised key-phrases extraction to support the automatic faceted classification of scientific documents in a given Digital Library in order to enhance the final users search, navigation and retrieval tasks.

8. Acknowledgments

Authors want to acknowledge Prof. C. Lee Giles[12] for sharing of the meta-information about papers and for access to the full text of the papers in PDF[13]. The authors would also like to thank for the support of the European Commission within the Erasmus Mundus programme, project of TH/Asia Link/010 (111084).

9. References

[1] P. Turney, "Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from

[12] http://clgiles.ist.psu.edu/
[13] Available from Citeseer http://citeseer.ist.psu.edu/

Labeled and Unlabeled data", NRC-44947/ERB-1096, August 13, 2002

[2] H. Hui, C.L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E.A. Fox, "Automatic document metadata extraction using support vector machines," in Proceedings of the 3rd ACM/IEEE-CS JCDL, 2003, pp. 37–48.

[3] S. Lawrence, C. L. Giles, and K. Bollacker, "Digital libraries and autonomous citation indexing," IEEE Computer, vol. 32, no. 6, 1999, pp. 67–71.

[4] D. Klein, and C. D. Manning, "Accurate Unlexicalized Parsing". Proceedings of the 41st Meeting of the Association for Computational Linguistics, 2003, pp. 423-430.

[5] M Marcus, B Santorini, and M Marcinkiewiecz, "Building a large annotated corpus of English, the penn treebank", Computational Linguistics, Vol. 19, Issue 2, 1993, pp. 313–330.

[6] I.H. Witten, G.W. Paynte, E. Frank, C. Gutwin, and C.G. Nevill-Manning, "KEA: Practical automatic keyphrase extraction." Proceedings of DL '99, 1999, pp. 254-256., http://www.nzdl.org/Kea/

[7] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning Hidden Markov Model Structure for Information Extraction", AAAI-99 Workshop on Machine Learning for Information, 1999.

[8] F. Peng, and A. McCallum, "Accurate Information Extraction from Research Papers using Conditional Random Fields", Proceedings of Human Language Technology Conference, 2004.

[9] O. Medelyan, and I.H. Witten, "Thesaurus Based Automatic Keyphrase Indexing", Proceedings of ACM/IEEE-CS JCDL, Chapel Hill, North Carolina, USA, 2006.

[10] P. Turney, "Learning to Extract Keyphrases from Text", NRC/ERB-1057. February 17, 1999.

[11] A Hulth, J Karlgren, A Jonsson, H Bostrom, and L Asker, "Automatic Keyword Extraction Using Domain Knowledge", Computational Linguistics and Intelligent Text Processing, Springer, 2004.

[12] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge". Proceedings of the 2003 conference on Empirical Methods in Natural Language Processing – Vol. 10, 2003, pp. 216-223

[13] C. Ercan, I. Cicekli, "Using Lexical Chains for Keyword Extraction", Information Processing and Management, Volume 43, Issue 6, Elsevier, 2007, pp. 1705-1714.

[14] J. Wang, and H Peng, "Keyphrases Extraction from Web Document by the Least Squares Support Vector Machine", IEE/WIC/ACM International Conference on Web Intelligence, 2005.

[15] P. Turney, "Coherent Keyphrase Extraction via Web Mining", Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), 2003, pp. 434-439.

[16] A. Ivanyukovich and M. Marchese, "Unsupervised free-text processing and structuring in digital archives," in Proceedings of 1st International Conference on Multidisciplinay Information Sciences and Technologies, 2006.

[17] I.H. Witten and E. Frank "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005. WEKA

[18] R.-E. Fan, P.-H. Chen, and C.-J. Lin. "Working set selection using the second order information for training SVM", Journal of Machine Learning Research 6, 1889-1918, 2005.