Automatic Extraction and Learning of Keyphrases from Scientific Articles

Yaakov HaCohen-Kerner, Zuriel Gross, and Asaf Masa

Department of Computer Sciences,
Jerusalem College of Technology (Machon Lev)
21 Havaad Haleumi St., P.O.B. 16031,
91160 Jerusalem, Israel
{kerner, zuriel, masa}@jct.ac.il

Abstract. Many academic journals and conferences require that each article include a list of keyphrases. These keyphrases should provide general information about the contents and the topics of the article. Keyphrases may save precious time for tasks such as filtering, summarization, and categorization. In this paper, we investigate automatic extraction and learning of keyphrases from scientific articles written in English. Firstly, we introduce various baseline extraction methods. Some of them, formalized by us, are very successful for academic papers. Then, we integrate these methods using different machine learning methods. The best results have been achieved by J48, an improved variant of C4.5. These results are significantly better than those achieved by previous extraction systems, regarded as the state of the art.

1 Introduction

Summarization is a process reducing an information object to a smaller size, and to its most important points [1, 18]. Various kinds of summaries (e.g.: headlines, abstracts, keyphrases, outlines, previews, reviews, biographies and bulletins) can be read with limited effort in a shorter reading time. Therefore, people prefer to read summaries rather than the entire text, before they decide whether they are going to read the whole text or not. Keyphrases, which can be regarded as very short summaries, may help even more. For instance, keyphrases can serve as an initial filter when retrieving documents. Unfortunately, most documents do not include keyphrases.

Moreover, many academic journals and conferences require that each paper will include a list of keyphrases. Therefore, there is a real need for automatic keyphrase extraction at least for academic papers. There are a few such systems. However, their performances are rather low. In this paper, we present a system that gives results significantly better than those achieved by the previous systems.

This paper is organized as follows: Section 2 gives background concerning extraction of keyphrases. Section 3 describes a few general kinds of machine learning. Section 4 presents our baseline extraction methods. Section 5 describes our model. Section 6 presents the results of our experiments and analyzes them. Section 7 discusses the results, concludes and proposes future directions.

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 657 – 669, 2005.

2 Extraction of Keyphrases

A keyphrase is an important concept, presented either in a single word (unigram), e.g.: 'learning', or a collocation, i.e., a meaningful group of two or more words, e.g.: 'machine learning' and 'natural language processing'. Keyphrases should provide general information about the contents of the document and can be seen as an additional kind of a document abstraction.

There are two main approaches concerning keyphrase generation: keyphrase assignment and keyphrase extraction. In the first approach, keyphrases are selected from a predefined list of keyphrases (i.e., a controlled vocabulary) [4]. These keyphrases are treated as classes, and techniques from text classification are used to assign classes to a given document. The training data associates a set of documents with each phrase in the vocabulary. The given document is converted to a vector of features and machine learning methods are used to induce a mapping from the feature space to the set of keyphrases. The advantages of this approach are simplicity and consistently. Similar documents can be described using the same keyphrases. Furthermore, using a controlled vocabulary ensure the required breadth and depth of the document coverage. The disadvantages of this approach are: (1) controlled vocabularies are expensive to create and maintain, so they are not always available and (2) potentially useful keyphrases that occur in the text of a document are ignored if they are not in the vocabulary. An example for a system that implements keyphrase assignment for given documents is described in [7]. In this system, keyphrases are selected from a hierarchical dictionary of concepts. Using this dictionary, general relevant concepts that are not included in the discussed document can be selected.

In the second approach, keyphrase extraction, the approach used in this research, keyphrases are selected from the text of the input document. All words and phrases included in the document are potential keyphrases. Usually, the keyphrases are extracted using machine learning algorithms based on combinations of several baseline extraction methods. The advantages of this approach are: (1) there is no need for creation and maintenance of controlled vocabularies, and (2) important keyphrases that occur in the text can be chosen. The disadvantages of this approach are: (1) lack of consistently; i.e., similar documents might be described using different keyphrases and (2) it is difficult to choose the most suitable keyphrases; i.e., the required breadth and depth of the document coverage is not ensured. An overview on keyphrase extraction methods is given by Jones and Paynter [15]. Among their results, they show that authors do provide good quality keyphrases for their papers.

Turney [22] shows that when authors define their keyphrases without a controlled vocabulary, about 70% to 80% of their keyphrases appear in the body of their documents. This suggests the possibility of using author-assigned free-text keyphrases to train a keyphrase extraction system. In this approach, a document is treated as a set of candidate phrases and the task is to classify each candidate phrase as either a keyphrase or non-keyphrase. A feature vector is calculated for each candidate phrase and machine learning methods are used to classify each candidate phrase as a keyphrase or non-keyphrase.

Although most of the keyphrase extraction systems work on single documents, keyphrase extraction is also used for more complex tasks. Examples of such systems are: (1) automatic web site summarization [27], and (2) keyphrase extraction for a whole corpus [24]. An overview of several relevant keyphrase extraction systems that

work on single documents, which is the investigated issue in this research, is given in the following sub-sections.

Turney [22] developed a keyphrase extraction system. This system uses a few baseline extraction methods, e.g.: TF (term frequency), FA (first appearance of a phrase from the beginning of its document normalized by dividing by the number of words in the document) and TL (length of a phrase in number of words). The best results have been achieved by a genetic algorithm called GenEx. For a collection of 362 articles collected from various domains, his system achieves a precision rate of about 24%. However, subjective human evaluation suggests that about 80% of the extracted keyphrases are acceptable to human readers. In this paper, he reports that these results are much better than the results achieved by the C4.5 decision tree induction algorithm [20] applied to the same task.

Frank et al. [6] propose another keyphrase extraction system called Kea. They used only two baseline extraction methods: TF_xIDF (how important is a phrase to its document) and distance (distance of the first appearance of a phrase from the beginning of its document in number of words). In addition, they apply the naïve Bayes learning method. They show that the quality of the extracted keyphrases improves significantly when domain-specific information is exploited. For a collection of 110 technical computer science articles, their system achieves a precision rate of about 28%, similar to the precision rate of GenEx, 29%, for the same data-base. However, they show that the naïve Bayes learning method used by them is much simpler and quicker than the genetic algorithm applied in GenEx.

Turney, in a further research [23], presents enhancements to the Kea keyphrase extraction algorithm that uses the naïve Bayes algorithm. His enhancements are designed to increase the coherence of the extracted keyphrases. The approach is to use the degree of statistical association among candidate keyphrases as evidence that they may be semantically related. The statistical association is measured using web mining. Experiments demonstrate that more of the output keyphrases match with the authors' keyphrases, which is evidence that their quality has improved. Moreover, the enhancements are not domain-specific: the algorithm generalizes well when it is trained on one domain (computer science documents) and tested on another (physics documents). The main limitation of the new method is the time required to calculate the features using web mining. Evaluation measures such as: recall, precision and F-measure are not presented.

Humphreys [14] proposes a keyphrase extractor for HTML documents. Her method finds important HTML tokens and phrases, determine a weight for each word in the document (biasing in favor of words in the introductory text), and uses a harmonic mean measure called RatePhrase to rank phrases. Her system retrieves a fixed number of phrases, 9, for inclusion in the summary. Using a test bed of URLs, her conclusion is that RatePhrase performs well as GenEx. However, evaluation measures such as: recall, precision and F-measure are not presented and there is no use of any machine learning method.

Hulth [12] develops a system capable of automatic extraction of keyphrases from abstracts of journal papers. In addition to the use of basic features (such as term frequency and n-grams), she used several basic linguistic features, e.g.: NP (Noun Phrase)-chunks and Pos (Part of Speech) tag patterns. These features serve as inputs to a supervised machine learning algorithm called rule induction. She reports on better results than those of Turney and Frank. For a collection of 2000 abstracts of journal papers, the best precision result 29.7% has been achieved by a combination of the linguistic

features: NP-chunks and the Pos tag patterns. The best F-measure score, 33.9%, has been achieved by a combination of the n-gram features and the Pos tag patterns.

In a further research [13], Hulth has reduced the number of incorrectly extracted keyphrases and achieved an F-measure score of 38.1%. The improvement was obtained by: (1) taking the majority vote of the three classifiers used in her previous work [12]: n-grams, NP-chunks and Pos tag patterns and (2) removing the subsumed keywords (keywords that are substrings of other selected keywords). The classifiers were constructed by Rule Discovery Sys-tem (RDS), a system for rule induction. The applied strategy is that of recursive partitioning, where the resulting rules are hierarchically organized (i.e., decision trees).

An additional keyphrase extraction system that makes use of linguistic features has been developed by D'Avanzo et al. [3]. Their system LAKE (Learning Algorithm for Keyphrase Extraction) uses features such as: PoS tagging, multi-word recognition and named entities recognition. They have trained the naïve Bayes classifier on only two features: TF x IDF and First Occurrence. Their conclusions were: (1) PoS-tagging information proved to be far from exhaustive, introducing a lot of noise. Some candidate phrases turned out to be useless pieces of longer sentences or irrelevant, and (2) A filter containing no verbs, proved to be the most reliable one.

Automatic syntactic analysis for detection of word combinations in a given text is proposed in [8]. Using parsing, this system finds word combinations, such as: keyphrases (e.g., *machine learning*), idioms (e.g., *to kick the bucket*) and lexical functions (e.g., *to pay attention*). However, such a full-scale analysis is not usable in real world applications because of unreliable results.

3 Machine Learning Methods

Machine learning (ML) refers to a capability of a system for autonomous acquisition and integration of knowledge. ML occurs in a system that can modify some aspect of itself so that on a subsequent execution with the same input, a different (hopefully better) output is produced. There are three main kinds of learning that can occur in machine learning systems – supervised, unsupervised and reinforcement learning.

Supervised learning is a learning that is supervised by a set of examples with class assignments and the goal is to find a representation of the problem in some feature (attribute) space that is used to build up profiles of the classes. Well-known classification models are: naïve Bayes classification [26], classification by the C4.5 decision tree induction [20] and neural networks [21].

Unsupervised learning has no guidance (supervision) of known classes. Therefore, it has no training stage. Clustering is an example of unsupervised learning. In this case, data which is similar is clustered together to form groups which can be thought of as classes. New data is classified by assignment to the closest matching cluster, and is assumed to have characteristics similar to the other data in the cluster.

Reinforcement learning is one step beyond unsupervised learning. In this learning, systems are given a limited feedback concerning the utility of the input-output mappings that are made. This feedback comes in the form of a reward function. While the reward function does not reveal the correct output for a given input, it does provide the system with an answer of whether the system output was correct or incorrect.

In our model, in order to find the best combinations of the baseline methods for keyphrase extraction we decide to apply supervised machine learning methods. This

kind of learning is well-investigated and rather successful in many domains. In addition, many supervised machine learning methods are available online. Furthermore, previous systems (Turney [22], Frank et al. [6], Hulth [12, 13] and D'Avanzo et al. [3]) framed their keyphrase extraction as a supervised learning problem.

4 Baseline Methods for Selecting the Most Important Keyphrases

In this section, we introduce the baseline methods we use for keyphrase extraction. Several methods are similar to those used in summarization systems (e.g.: [16, 10]) for selecting the most important sentences. Other methods were formalized by us. Similar methods have been used for Hebrew News HTML Documents in [11].

In all methods, words and terms that have a grammatical role for the language are excluded from the key words list according to a ready-made stop list. This stop-list contains approximately 456 high frequency close class words (e.g.: we, this, and, when, in, usually, also, near).

(1) **Term Frequency (TF):** This method rates a term according to the number of its occurrences in the text [5, 17, 9]. Only the N terms with the highest TF in the document are selected.

(2) **Term length (TL):** TL rates a term according to the number of the words included in the term.

(3) **First N Terms (FN)**: Only the first N terms in the document are selected. The assumption is that the most important keyphrases are found at the beginning of the document because people tend to place important information at the beginning. This method is based on the baseline summarization method which chooses the first N sentences. This simple method provides a relatively strong baseline for the performance of any text-summarization method [2].

(4) **Last N Terms (LN)**: Only the last N terms in the document are selected. The assumption is that the most important keyphrases are found at the end of the document because people tend to place their important keyphrases in their conclusions which are usually placed near to the end.

(5) **At the Beginning of its Paragraph (PB):** This method rates a term according to its relative position in its paragraph. The assumption is that the most important keyphrases are likely to be found close to the beginning of their paragraphs.

(6) **At the End of its Paragraph (PE):** This method rates a term according to its relative position in its paragraph. The assumption is that the most important keyphrases are likely to be found close to end of their paragraphs.

(7) **Resemblance to Title (RT)**: This method rates a term according to the resemblance of its sentence to the title of the article. Sentences that resemble the title will be granted a higher score [5, 18, 19].

(8) **Maximal Section Headline Importance (MSHI):** This method rates a term according to its most important presence in a section or headline of the article. It is a known that some parts of papers are more important from the viewpoint of presence of keyphrases. Such parts can be headlines and sections as: abstract, introduction and conclusions.

(9) **Accumulative Section Headline Importance (ASHI):** This method is very similar to the previous one. However, it rates a term according to all its presences in important sections or headlines of the article.

(10) **Negative Brackets (NBR):** Phrases found in brackets are not likely to be keyphrases. Therefore, they are defined as negative phrases, and will grant negative scores.
(11) **TF x MSHI:** This method serves as an interaction between two rather successful methods TF and MSHI. This method resembles the TL*TF method, which was successful in [22].

5 Our Model

5.1 General Description

Our model, in general, is composed of the six following steps (special concepts used in this algorithm will be explained below):

For each article that is in our database:

(1) Extract keyphrases that do not contain stop-list words.
(2) Transform these keyphrases into lower case.
(3) Apply all baseline extraction methods on these keyphrases.
(4) Compare between the most highly weighted keyphrases extracted by our methods to the keyphrases composed by the authors; analyze the results and present full and partial matches.
(5) Apply several common supervised machine learning methods in order to find the best combinations of these baseline methods.
(6) Compare between the best machine learning results achieved in our system to the best machine learning results achieved in systems, which are regarded as the state of the art.

A full match for a unigram is a repetition of the same word including changes such as singular/plural or abbreviations, first letter in lower case / upper case. A partial match between two different unigrams is defined if both words have the same first five letters (explanation below). All other pairs of words are regarded as failures.

A partial match between different unigrams is defined when the first five letters of both words are the same. That is because in such a case we assume that these words have a common radical. Such a definition, on the one hand, usually identifies close words like nouns, verbs, adjectives, and adverbs. On the other hand, it does not enable most of non-similar words to be regarded as partial matches.

A positive example for this definition is as follows: all 8 following words are regarded as partial matches because they have the same 5-letter prefix "analy": the nouns "analysis", "analyst", "analyzer", the verb "analyze", and the adjectives "analytic", "analytical", "analyzable ", and the adverb "analytically". A negative example for this definition is: all 8 following words: "confection", "confab", "confectioner", "confidence", "confess", "configure", "confinement", and "confederacy" are regarded as non partial matches because they have in common only a 4-letter prefix "conf".

Concerning keyphrases which are not unigrams, a full match is a repetition of the same keyphrase. That is, a repetition of all the words included in the keyphrase. A partial match between two different keyphrases is defined when both keyphrases share at least one word. All other pairs of keyphrases are regarded as failures.

Using each one of the baseline methods (Section 4) our system chooses the N most highly weighted keyphrases. The value of N has been set at 5 and 15 in two

different experiments because these values have been used in the experiments done by Turney [22] and Frank et al. [6].

5.2 Evaluation Measures

In order to measure the success of our baseline extraction methods, we use the popular measures: recall, precision and f-measure. These measures are defined briefly below, using the following table of keyphrases' results, which is relevant to our model.

Table 1. Author's and extraction method's keyphrases

		Author's keyphrases	
		True	**False**
Extraction- method's keyphrases	**True**	a	b
	False	c	d

Precision is defined as $a / (a + b)$. Recall is defined as: $a / (a + c)$ and F-Measure which is an harmonic mean of Precision and Recall is defined as $\frac{(\alpha+1)\times \text{Recall}\times \text{Precision}}{\text{Recall}+(\alpha\times \text{Precision})}$, where $\alpha = 1$ gives the same importance for Recall and Precision. In this case, F-Measure is defined as $\frac{2\times \text{Recall}\times \text{Precision}}{(\text{Recall}+\text{Precision})}$.

In our research, it means that recall is defined as the number of keyphrases that appear both within the system's keyphrases and within the keyphrases composed by the authors divided by the number of keyphrases composed by the authors. Precision is defined as the number of keyphrases that appear both within the system's keyphrases and within the keyphrases composed by the authors divided by the number of keyphrases extracted by the system. F-measure is a common weighed average of the above two measures.

6 Experiments

6.1 Data Sets

We have constructed a dataset containing 161 academic papers in Physics taken from the dataset used in the experiments made by Turney [22]. Each document contains in average 1243 sentences containing 8016 words in average. Each document has its own keyphrases composed by the authors of the original documents. The total number of keyphrases is 669. That is, each document contains in average about 4.16 keyphrases. Table 2 presents various statistics concerning these documents. Table 3 presents the distribution of # words per keyphrase.

In addition, it is important to point that about 28% of the keyphrases do not appear (in an exact form) in their papers. Our baseline methods extract only keyphrases that are found in the papers. Therefore, we are limited in full matches to a maximum of 72%. Full and partial presence of keyphrases in their articles is presented in Table 4.

Table 2. Various statistics concerning our dataset

# of keyphrases	# of articles	Accum. % of articles	% of key-phrases	Accum. # of key-phrases	Accum. % of key-phrases
1	4	2.5	0.6	4	0.6
2	16	12.4	4.8	36	5.4
3	59	49.1	26.5	213	31.8
4	37	72.0	22.1	361	54.0
5	14	80.7	10.5	431	64.4
6	19	92.5	17.0	545	81.5
7	5	95.7	5.2	580	86.7
8	1	96.3	1.2	588	87.9
9	2	97.5	2.7	606	90.6
11	2	98.8	3.3	628	93.9
12	1	99.4	1.8	640	95.7
29	1	100.0	4.3	669	100.0

Table 3. Distribution of # words per keyphrase

# of words per keyphrase	# of keyphrases
1	107
2	332
3	166
4	33
5	5
6	3

Table 4. Full and partial presence of keyphrases in their articles

	# in articles	% in articles
Full presence	481	72
Partial presence	150	22
Absence	38	6

About 72% of the keyphrases appear somewhere in the body of their documents. This is similar to the finding of Turney [22] who reports that typically about 70% to 80% of the authors' keyphrases appear somewhere in the body of their documents.

About 6% of the keyphrases composed by their authors do not even exist in a partial form in their papers. Examples of such keyphrases are: (1) "non-equilibrium kinetics", (2) "protons", and (3) "gamma gamma". These keyphrases may be classified into categories of either more general or more specific keyphrases belonged to the research domain that includes the discussed paper. This kind of keyphrases might be found by the system, by for example mining the web and finding similar documents containing them or searching in dictionaries for synonyms. Another solution for finding general keyphrases is to use keyphrase assignment, as done by [7]. Using a

hierarchical dictionary of concepts, relevant concepts that are not included in the discussed paper can be selected.

About 22% of the keyphrases composed by their authors exist in their papers only partially. Examples for such keyphrases are: (1) "lattice simulation" where both "lattice" and "simulation" were included in the paper but separately, (2) "dynamical fermions" where only "fermions" was included in the paper, (3) "Hart rate" where "heart rate" was found in the paper and (4) "$1 = N$ expansion" where "$1 = Nf$ expansion" was found in the paper. The first two keyphrases can be classified into a category of more specific keyphrases belonged to the domain of discussed paper even though they are not mentioned in their papers. This kind of keyphrases might also be found by the system, by mining the web and finding similar documents containing them. The last two keyphrases do not have full matches because of syntax errors. This kind of errors might be discovered while preprocessing the papers and suggestions for correction can be given in this stage.

6.2 Results of Baseline Extraction Methods

Using each baseline method (Section 4), our system chooses the N most highly weighted keyphrases. The value of N has been set at 5 in the first experiment and 15 in the second experiment. These values of N have been chosen because these are the numbers of the retrieved keyphrases by the two previous related systems GenEx [22] and Kea [6]. Table 5 presents the recall, precision and the f-measures results, respectively, of our baseline extraction methods.

Concerning full matches, the best baseline method was found as MSHI (Maximal Section Headline Importance). That is, this method, which is based on the most important headline or section of a given paper, is very successful for academic papers. In contrast to results discovered by Frank et al. [6], in our model, TF (Term Frequency) and FN (First N) were not the best methods. However, they achieve rather good results. This finding might point that these common methods are not the best for academic papers and unique methods designed for academic papers can be better.

Concerning partial matches and up, the best baseline methods were found as TF x MSHI and ASHI (Accumulative Section Headline Importance). Two additional promising methods were PB (at the Beginning of its Paragraph) and TF.

The results presented in Table 5 are based on the keyphrases composed by the authors of the papers, although some of the keyphrases do not exist in the papers. As mentioned in Section 5.1, the result of full matches is limited to a maximum of 72%. Therefore, the results of our baseline methods are actually better.

6.3 Supervised Machine Learning Results

As mentioned in Section 3, we decide to use supervised machine learning methods. We have applied several well-known supervised classification models: naïve Bayes classification [26], classification by the C4.5 decision tree induction [20] and neural networks [21].

Table 5. Precision/recall/f-measures results for our baseline methods

#	Method	Extracted keyphrases	% of full matches			% of partial matches			% of partial matches and up		
			R	P	F	R	P	F	R	P	F
1	TF	5	**6.8**	**4.0**	**5.0**	33.9	18.4	23.9	**40.7**	**22.4**	**28.9**
		15	**13.5**	**3.6**	**5.7**	51.8	13.0	20.7	**65.3**	**16.6**	**26.4**
2	TL	5	3.6	2.0	2.6	2.0	0.0	0.0	5.5	2.0	2.9
		15	8.4	2.1	3.4	0.0	0.0	0.0	8.4	2.1	3.4
3	FN	5	**10.9**	**6.8**	**8.4**	14.3	7.6	9.9	25.2	14.4	18.3
		15	**24.5**	**5.8**	**9.4**	26.1	6.2	10.1	50.6	12.1	19.5
4	LN	5	1.7	0.9	1.1	3.9	2.2	2.8	5.5	3.1	4.0
		15	5.0	1.2	2.0	7.1	1.9	2.9	12.0	3.1	4.9
5	PB	5	7.2	4.1	5.2	38.6	21.1	27.3	**45.8**	**25.2**	**32.5**
		15	19.9	5.2	8.3	45.8	11.3	18.2	**65.7**	**16.6**	**26.5**
6	PE	5	4.3	2.7	3.3	21.7	11.6	15.1	26.0	14.3	18.4
		15	8.5	2.3	3.7	29.8	7.4	11.9	38.3	9.7	15.5
7	RT	5	8.0	4.8	6.0	31.4	17.1	22.2	39.4	22.0	28.2
		15	18.7	1.1	2.2	37.6	13.0	19.4	56.3	14.2	22.7
8	MSHI	5	**17.5**	**10.2**	**12.9**	17.9	9.8	12.7	35.3	20.0	25.5
		15	**29.4**	**7.1**	**11.4**	30.3	7.5	12.1	59.7	14.6	23.5
9	ASHI	5	8.1	4.8	6.1	**36.6**	**19.8**	**25.7**	**44.7**	**24.6**	**31.7**
		15	14.6	3.9	6.2	**54.8**	**13.7**	**21.9**	**69.4**	**17.7**	**28.1**
10	NBR	5	1.9	1.2	1.5	10.2	5.5	7.1	12.1	6.7	8.6
		15	4.4	1.1	1.8	16.9	4.4	6.9	21.3	5.5	8.8
11	TF x MSHI	5	8.9	5.2	6.6	**43.4**	**23.9**	**30.8**	**52.3**	**29.1**	**37.4**
		15	17.9	4.8	7.5	**54.9**	**13.9**	**22.2**	**72.8**	**18.7**	**29.8**

We applied these methods using the web-site of Weka [25], as done by Frank et al. [6] and D'Avanzo et al. [3]. Weka is a collection of machine learning algorithms programmed in Java for data mining tasks, such as: classification, regression, clustering, association rules, and visualization.

Table 6 presents the optimal learning results achieved by three common machine learning methods: J48[1], multilayer perceptron and naïve Bayes.

The best results in Table 6 have been achieved by J48. Therefore, this method has been selected as the best machine-learning method for our task.

Table 7 compares the precision results for extraction of 5 and 15 keyphrases between our system using J48 to the best results achieved by machine learning methods in GenEx and Kea, which are regarded as the state of the art. The reason why we compare only the precision results is because this is the only common measure used by all three systems. Kea presents only precision results. GenEx, in addition, presents a subjective human measure concerning the acceptance of the extracted keyphrases to human readers.

Our results are significantly better than those achieved by GenEx and Kea. For example, our system achieved a precision rate of 55.4% / 28.5% while GenEx achieved (on the smaller dataset) only 29% / 17% and Kea achieved only 28% / 16.5% for 5 / 15 extracted keyphrases, respectively.

In addition, our F-measure results (in Table 6) are significantly better than the best F-measure scores achieved for extraction of keyphrases from journal abstracts by Hulth [12, 13] 33.9% and 38.1%, respectively.

[1] J48 is a machine learning method in Weka [25] that actually implements a slightly improved version (Revision 8) of C4.5.

Table 6. Learning results in our system

Method	Matches	% of precision	% of recall	% of F_measure	Optimal # of extracted keyphrases
J48	Full	**84.1**	**59.46**	**69.67**	2.94
	Partial	**84.5**	**77.25**	**80.71**	3.80
Multilayer Perceptron	Full	77	45.44	57.15	2.45
	Partial	74.8	62.35	68.01	3.46
Naïve Bayes	Full	62.5	53.78	57.81	3.58
	Partial	80.4	19.90	31.91	1.03

Table 7. Comparison of precision results between learning systems

System	# of papers	# of extracted keywords	Precision
GenEx	362	5	23.9%
		15	12.8%
	110	5	29%
		15	17%
Kea	110	5	28%
		15	16.5%
Our System	161	**5**	**55.4%**
		15	**28.5%**

Explanations to these findings can be: (a) we work on academic papers only and we apply specific extraction methods for them; (b) in contrast to the related systems that used combinations of a low number (2/3) of baseline extraction methods, we have used a combination of a relatively high number (11) of baseline methods; and (c) due to J48 we have found a successful combination of our baseline extraction methods.

7 Conclusions and Future Work

Several unique baseline extraction methods, formalized by us have been found as very successful for academic papers. In contrast to previous extraction systems, we have used a combination of a relatively high number of baseline methods. Machine learning results achieved by J48 have been found significantly better than those achieved by extraction systems, which are regarded as the state of the art.

Future directions for research are: (1) Developing methods based on domain-dependant cue phrases for keyphrase extraction, (2) Applying other machine-learning techniques in order to find the most effective combination between these baseline methods, (3) Conducting more experiments using additional documents from additional domains.

Concerning research on academic papers from additional domains, there are many potential research directions. For example: (1) Which extraction methods are good for which domains? (2) What are the specific reasons for methods to perform better or

worse on different domains? (3) What are the guidelines to choose the correct methods for a certain domain? (4) Can the appropriateness of a method for a domain be estimated automatically?

Acknowledgements. The authors would like to thank Peter Turney for sharing his datasets; Ittay Stern and David Korkus for their support; and Alexander Gelbukh and anonymous reviewers for their fruitful comments.

References

1. Alterman, R.: Text Summarization. In: Shapiro, S.C. (ed.): Encyclopedia of Artificial Intelligence. John Wiley & Sons, New York (1992) 1579–1587
2. Brandow, B., Mitze, K., Rau, L.F.: Automatic Condensation of Electronic Publications by Sentence Selection. Information Processing and Management 31(5) (1994) 675–685
3. D'Avanzo, E., Magnini, B., Vallin, A.: Keyphrase Extraction for Summarization Purposes: The LAKE System at DUC-2004. Document Understanding Workshop (2004)
4. Dumais, S. T., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. Proceedings of ACM-CIK International Conference on Information and Knowledge Management, ACM Press, Philadelphia (1998) 148–155
5. Edmundson, H.P.: New Methods in Automatic Extraction. Journal of the ACM. 16(2) (1969) 264–285
6. Frank, E., Paynter, G.W., Witten I.H., Gutwin C., Nevill-Manning, C.G.: Domain-Specific Key-Phrase Extraction. Proc. IJCAI. Morgan Kaufmann (1999) 668–673
7. Gelbukh, A., Sidorov, G., Guzmán-Arenas, A..: A Method of Describing Document Contents through Topic Selection. Proc. SPIRE'99, International Symposium on String Processing and Information Retrieval, Mexico, (1999), 73–80.
8. Gelbukh, A., Sidorov, G., Han, S.-Y., Hernandez-Rubio, E.: Automatic Syntactic Analysis for Detection of Word Combinations. Proc. CICLing-2004: Intelligent Text Processing and Computational Linguistics, Mexico, Lecture Notes in Computer Science 2945, Springer-Verlag, Berlin Heidelberg New York (2004) 243–247
9. HaCohen-Kerner, Y.: Automatic Extraction of Keywords from Abstracts. Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Vol. 1. Lecture Notes in Artificial Intelligence 2773. Springer-Verlag, Berlin Heidelberg New York (2003) 843–849
10. HaCohen-Kerner, Y., Malin, E., Chasson, I.: Summarization of Jewish Law Articles in Hebrew, Proceedings of the 16th International Conference on Computer Applications in Industry and Engineering, Las Vegas, Nevada USA, Cary, NC: International Society for Computers and Their Applications (ISCA) (2003) 172–177
11. HaCohen-Kerner, Y., Stern, I., Korkus, D.: Baseline Keyphrase Extraction Methods from Hebrew News HTML Documents, WSEAS Transactions on Information Science and Applications, 6(1) (2004) 1557–1562
12. Hulth, A.: Improved Automatic Keyword Extraction Given More Linguistic Knowledge, Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, (2003) 216–223
13. Hulth, A.: Reducing False Positives by Expert Combination in Automatic Keyword Indexing. Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'03), Borovets (2003) 197–203

14. Humphreys, K.J.B.: Phraserate: An HTML Keyphrase Extractor. Technical report, University of California, Riverside, Riverside, California (2002)
15. Jones, S., Paynter, G.W.: Automatic Extraction of Document Keyphrases for Use in Digital Libraries: Evaluation and Applications, Journal of the American Society for Information Science and Technology, 53(8) (2002) 653–677
16. Kupiec, J., Pederson, J., Chen, F.: A Trainable Document Summarizer. Proceedings of the 18th Annual International ACM SIGIR (1995) 68–73
17. Luhn, H.P.: The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 2(2) (1958) 159–165
18. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization, Cambridge, MA: MIT Press (1999) ix–xv
19. Neto, J.L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization Using a Machine Learning Approach. Proc. SBIA (2002) 205–215
20. Quinlan, J. R.: C4.5: Programs For Machine Learning. Morgan Kaufmann, Los Altos (1993)
21. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, Upper Saddle River, NJ: Prentice-Hall (1995)
22. Turney, P.: Learning Algorithms for Keyphrase Extraction. Information Retrieval Journal 2(4) (2000) 303–336
23. Turney, P.: Coherent Keyphrase Extraction via Web Mining. Proceedings of IJCAI'03 (2003) 434–439.
24. Wu, J., Agogino, A. M.: Automating Keyphrase Building with Multi-Objective Genetic Algorithms, Proceedings of the 37th Annual Hawaii International Conference on System Science, HICSS (2003) 104–111
25. Weka: http://www.cs.waikato.ac.nz/~ml/weka (2004)
26. Yang, Y., Webb, G. I.: Weighted Proportional k-Interval Discretization for Naïve-Bayes Classifiers. Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, (2003) 501–512
27. Zhang, Y., Milios, E., Zincir-Heywood, N.: A Comparison of Keyword- and Keyterm-based Methods for Automatic Web Site Summarization, in Technical Report WS-04-01, Papers from the on Adaptive Text Extraction and Mining, San Jose, CA, (2004) 15–20