RapidKey Phrase Extraction For Turkish

Murat Oruc1\*,Uğur Mertoglu1\*\*

1Department of Computer Engineering, Hacettepe University, Ankara, Turkey

\*muratoruc@hacettepe.edu.tr \*\*umertoglu@hacettepe.edu.tr

Abstract— **A keyword/keyphrase is a word or phrase used as a search term. So keyphrases have useful information extracted from documents. They carry the main ideas of the text. Knowing the list of keyphrases help us by saving time which can be lost during searching for a document about a particular topic. But, there are many documents especially on the Internet that do not have a list of assigned keyphrases. So automatic extraction of keyphrases becomes an important task. In this paper, we present a key phrase extraction method for Turkish which includes some features and learning stage. After learning we show that our algorithm is successful in terms of extracting keyphrases in test stage. Of course it has some defective sides and we explained them in future work.**

*Keywords- key phrase extraction, naive bayes,*machine learning.

# **I.INTRODUCTION**

As a brief explanation key phrases are formed by a word or few words occurring with a high frequency within a text document. They have very important benefits. For example by the help of the keyphrases we can reach semantic metadata to characterize or summarize the document. We use key phrases for different kind of purposes including; summarization, indexing, making more precise searches, and also not to examine the document much etc.

Keyphrases are meant to serve multiple goals. For example, when they are printed on the first page of a journal article, the goal is summarization. They enable the reader to quickly determine whether the given article is in the reader’s fields of interest. When they are printed in the cumulative index for a journal, the goal is indexing. They enable the reader to quickly find a relevant article when the reader has a specific need. When a search engine form has a field labelled *keywords*, the goal is to enable the reader to make the search more precise. A search for documents that match a given query term in the keywordfield will yield a smaller, higher quality list of hits than a search for the same term in the full text of the documents. Keyphrases can serve these diverse goals and others, because the goals share the requirement for a short list of phrases that captures the main topics of the documents.

It is so good that many academic articles have a list of keyphrases that are assigned by their authors, unfortunately, it is worth to consider that there are also many digital documents especially on the Internet that do not have such a list. Assigning keyphrases to these documents manually is a tedious process and requires knowledge of the subject. It can be solved by automatic keyphrase extraction. Automatically extracting the important phrases from the document differs from automatically generating the important phrases from the document. The main difference is what we deeply examine in the rest of the paper; automatically extracted keyphrases should appear in the document.

In this study we took this problem as a machine learning topic. Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data[1]. Also we tried some algorithms to find the best the way to solve the problem. As a consequence we believed that this study can be best achieved by supervised learning algorithms. Supervised learning that generates a function from training data and this function maps inputs to desired outputs.

Supervised learning algorithms first takes the training data and creates a function. After generating the function, it takes the test data. Test data consist of only input vectors. It predicts the desired output of the given input by using this function.

We used supervised learning algorithm to extract keyphrases from a document. We tried to decide for new documents finding a set of phrases that must be classified either as a keyphrase or not according to training data. From the start, we first prepare a training data which contain the documents with the assigned keyphrases. After creating the function, the algorithm takes the test data which contain the documents without the assigned keyphrases. We expect the algorithm to extract keyphrases from these documents rapidly.

Our study is upon Naive Bayes which is a major

supervised machine learning algorithm. The methods we used is explained in detail in the following sections. We make some use of Keyphrase Extraction Algorithm(KEA)[1] study which is used for automatically extracting keyphrases from documents.

KEA is developed by a group from Waikato University. Their extracting reasonable summaries from text documents. KEA is developed for English documents. We use it to extract keyphrases from Turkish text documents by making some changes, for example changing its stemmer, stopwords list other techniques. Rest of the paper continues with the Related Work Section II. The differences between the original KEA algorithm and our algorithm are explained in detail in Section III. The experiments in this paper are based on a document collection which consists of 60 Turkish articles. The corpora is explained in Section III.

Our performance measure is the number of correctly identified author-assigned keyphrases. After extracting the keyphrases from the test data, we calculate the total correctly identified keyphrases with respect to the total author assigned keyphrases. Performance analysis of Turkish keyphrase extraction is given in Section IV. We conclude (in Section V) that KEA algorithm can be used to extract keyphrases from Turkish text documents.

# **II.RELATED WORK**

Keyphrase Extraction has been studied in synchronization with the topics which care about the extractions such as text summarization, relationship extraction ,information extraction etc. in terms of machine learning.

With improvements in topic different methods has occurred. These are some techniques used in different hybrid methods also with the other techniques that is not mentioned here :Word Frequency Analysis, Word Co-Occurrence Relationships, Using a Document Corpus, Frequency-Based Keyword Extraction, Content-Sensitive Keyword Extraction, Keyword Extraction Using Lexical Chains, Keyphrase Extraction Using Bayes Classifier[2].Also there are number of previous works on this which branch that we used [3],[4],[5],[6].

Some supervised and unsupervised keyphrase extraction methods have already been reported by the researchers. Analgorithm to choose noun phrases from a document as keyphrases has been proposed in [7]. Phrase length, its frequency and the frequency of its head noun are the features used in this work.

Chien[8]developed a PAT-tree-based keyphrases extraction system for Chinese and other oriental languages.

HaCohen-Kerner et al [9] proposed a model for keyphrase extraction based on supervised machine learning and combinations of the baseline methods. They applied J48, an improved variant of C4.5 decision tree for feature combination.

Hulth et al [10] proposed a keyphrase extraction algorithm which a hierarchically organized thesaurus and the frequency analysis were integrated. The inductive logic programming has been used to combine evidences from frequency analysis and thesaurus.

A graph based model for keyphrase extraction has been presented in [11]. A document is represented as a graph in which the nodes represent terms, and the edges represent the co-occurrence of terms. Whether a term is a keyword is determined by measuring its contribution to the graph.

A Neural Network based approach to keyphrase extraction has been presented in [12] that exploits traditional term frequency, inverted document frequency and position(binary) features. The neural network has been trained to classify a candidate phrase as keyphrase or not.

Turney [13] treats the problem of keyphrase extraction as supervised learning task. Turney’s program is called Extractor. In this task, nine features are used to score a candidate phrase; some of the features are positional information of the phrase in the document and whether or not the phrase is a proper noun. Keyphrases are extracted from candidate phrases based on examination of their features.

A keyphrase extraction program called Kea, developed by Frank et al. [14][15], uses the Bayesian learning technique for keyphrase extraction task. A model is learned from the training documents with exemplar keyphrases and corresponds to a specific corpus containing the training documents. Each model consists of a Naive Bayes classifier and two supporting files containing phrase frequencies and stopped words. The learned model is used to identify the keyphrases from a document. In both Kea and Extractor, the candidate keyphrases are identified by splitting up the input text according to phrase boundaries(numbers, punctuation marks, dashes, and brackets etc.).

An n-gram based technique for filtering keyphrases has been presented in [16]. In this approach, authors compute n-grams such as unigram, bigram etc for extracting the candidate keyphrases which are finally ranked based o the features such as term frequency, position of a phrase in a document and a sentence. In the rest of the paper we utilized from these documents[17],[18],[19],[20].

**III.KEY PHRASE EXTRACTIONALGORITHM**

In this part, we will consider our key phrase extraction algorithm step by step. Firstly, we superficially define the stages and its subsections and then explain them in detail.

We suggest a supervised learning method. It has two stages:

1. Training Stage: In this stage, our algorithm learns on the training data and present us a model. Training data comprises 40 documents with the author-assigned key-phrases.
2. Test Stage: Our learned model use test data consisting of 20 documents and it extracts key-phrases of the documents by the help of Naive Bayes algorithm.

Two stages, both, have two sections.

1. Part of Speech Tagger.
2. Form a pool of candidate phrases.
3. Calculate the features below.
4. TF-IDF feature
5. Key Phrase Position feature
6. Relative Length feature

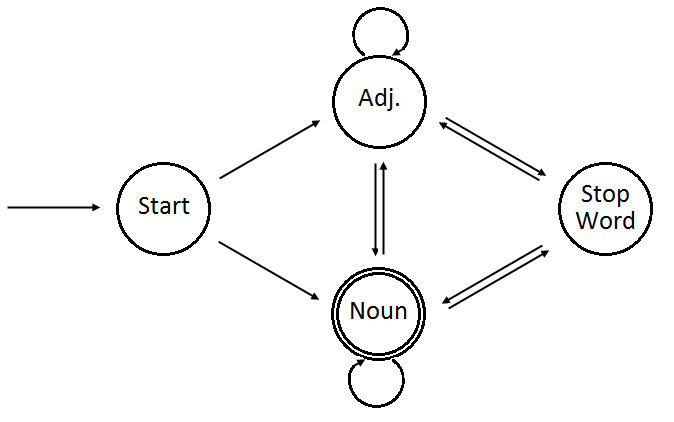
Now we will explain those common sections and two stages respectively in detail.

1. **Part of Speech Tagger**

In this section, we scan the incoming text and remove the numbers, all kinds of punctuation marks and abbreviations that exist in our abbreviations file taken form Turkish Language Corporation's site. After that, we parse the incoming text into sentences according to the delimiter chars ('.', '?', '!'). Then we parse these sentences into words whose length equals to three or more. Finally, we tag these words according to their stem by using Zemberek which is a strong Turkish morphological analyzer. We also tag stop words according to our stop-list. Finally we form a word pool which contains only the types of noun, adjective and stop words.

1. **Form a Pool of Candidate Phrases**

Candidate key phrases are formed as shown in the Figure-1(Formation of Candidate Key Phrase by a Finite State Machine). Noun phase also consists of proper nouns.



**Figure-1: Formation of Candidate Key Phrase by a Finite State Machine.**

If a key phrase is one word, then we add the stem of that word to the candidate key phrase pool. Otherwise we add original words to the pool. Then, we generate the candidate all possible variations of candidate key phrases. After that, we filter the key phrases whose length equals to five words or less.

1. **TF-IDF feature**

This feature is calculated by using two frequencies: the frequency of a phrase in a document and the frequency of that phrase in our set consisting of 60 documents. If a phrase is used in a document with a high frequency, the probability of being a keyphrase of this phrase increases. If a phrase is used in general use with a high frequency, the probability of being a keyphrase of this phrase decreases. We calculate this feature each of candidate key phrases according to the Formula-1 below. We modify the original formula by adding 1 to document frequency and adding size of the global corpus once more to itself as a smoothing technique.

**Original TF-IDF formula.**

**Modifies TF-IDF formula.**

**Formula-1: Calculating TF-IDF feature.**

Components of (Formula-1) are as follows:

1. freq(P,D) is the number of times P occurs in D,
2. size(D) is the number of words in D,
3. df(P) is the number of documents containing P in the global corpus,
4. N is the size of the global corpus.

After calculation process, we normalize the TF-IDF values of each key phrase as being the maximum value will equal to 1. While this process, we smooth the division of second part of the formula by adding 1 to document frequency and size of the documents once more to size of the documents.

1. **Key Phrase Position feature**

As we know, our candidate key phrase pool is formed by ordering them according to their original position in the text. So when we find the place of each candidate key phrase from the pool, this place number becomes the key phrase position number. After that we calculate the key phrase position feature of each candidate key phrase by the Formula-2 shown below.

**Formula-2: Calculating Key Phrase Position feature.**

1. **Relative Length feature**

Relative length feature is calculated as the number of characters in the phrase is divided by the number of characters in the candidate phrase that has the

maximum. We calculate the relative length feature of each candidate key phrase.

Now it is time to begin training stage to create a model and test this model in test stage. We have a data corpus consisting of 60 articles and their author assigned keyphrases. We select 40 of them as our training data and 20 of them as test data.

1. **Training Stage**

In this stage, we take training data and their author assigned keyphrases. After selecting candidate phrases and calculating their feature values, we mark these phrases as key phrases or non-key phrases by using the author assigned key phrases. If a phrase is given in the author assigned key phrases list, then this phrase is marked as a keyphrase, otherwise it is marked as a non-keyphrase. Being a keyphrase or not being a keyphrase is the class value for Naive Bayes algorithm. It generates a model using training data to predict the class.

For each feature, we calculate the mean, variance and standard deviation values of each document. After summation of these values cumulatively for 40 documents, we find the average of mean, variance and standard deviation. Normal Distribution(Formula-3) of each feature by using these metrics is calculated and this value has been the weights of each feature.

1. **Test Stage**

In this stage, we take test data which contains only text documents. After selecting candidate phrases and calculating their feature values, we determine the probability of being a keyphrase for each candidate phrase by using prior knowledge we gained in the training stage. First, we find normal distribution of each feature to determine the probability of each candidate key phrase, and we use the mean, variance and standard deviation of each feature we found in the training stage in the Formula-3(Calculating normal distribution).



**Formula-3: Calculating probability of each feature by normal distribution.**

After that, Naive Bayes algorithm calculates Formula-4 and Formula-5 for each candidate phrase.

**Formula-4: Calculating probability of being a key phrase.**

**Formula-5: Calculating probability of not being a key phrase.**

Components of the equation are as follows:

1. t is TF×IDF
2. d is distance
3. r is relative length
4. Y is the number of positive phrases in the training documents
5. N is the number of negative instances in the training documents

After calculating these equations, their results are substituted in Formula-6. This is the final score of a key phrase.

**Formula-6: Calculating the final score of a key phrase.**

Candidate phrases are ranked according to the values calculated from Formula-6 and the top specific number of key phrases are selected.

**IV. EVALUATION STAGE**

In this stage, we analyzed our test stage and show some statistics about the accuracy of our algorithm. We process test data which contains text documents and their author-assigned key phrases.

We use Laplace and accuracy metric types (Formula-7) to evaluate our algorithm by using our test data. One of the document's extracted and author-assigned keyphrases are shown in chart-1 when the extracted key phrase count is 10 and in chart-2 when the count is 5.

**Formula-7:Laplace metric.**

Components of (Formula-7) are as follows:

***n*** *: Number of instances*

*: Number of instances covered by algorithm*

***k*** *: Number of classes*

|  |  |
| --- | --- |
| **Author-Assigned KPs** | **Extracted KPs** |
| ***Kesme kuvvetleri*** | ***Kesme kuvvetleri*** |
| *Kesici takım kaplaması* | *Takım* |
| *Paslanmaz çelik* | *Kesme hızı* |
| ***Yüzey pürüzlülüğü*** | *Çelik* |
|  | *Kesme hızının* |
|  | *Kaplan* |
|  | *Karbür* |
|  | ***Yüzey pürüzlülüğü*** |
|  | *Hız* |
|  | *Takım kaplamasının* |

**Chart-1: 10-KeyPhrases Extraction Similarity.**

|  |  |
| --- | --- |
| **Author-Assigned KPs** | **Extracted KPs** |
| *Karınca metasezgiseli* | *Karınca* |
| ***Gezgin satıcı problemi*** | *Karınca kolonileri* |
|  | *Gezgin satıcı* |
|  | ***Gezgin satıcı problemleri*** |
|  | *Karınca sistemi* |

**Chart-2: 5-KeyPhrases Extraction Similarity.**

To see effectiveness of our algorithm, we calculate Laplace and accuracy value of the algorithm by extracting top 10 and top 5 keyphrases of a document. In two condition we get different Laplace values and it is logical to get higher values in top 10 key phrases' condition.

The statistics are shown in chart-2 and chart-3.

|  |  |
| --- | --- |
| **Extracting n Key Phrases** | **Laplace value** |
| **5** | **0,267559523809524** |
| **10** | **0,291369047619048** |

**Chart-2:Effectiveness of the key phrase extraction algorithm.**

# **V. CONCLUSION AND FUTURE WORK**

In this study, a generic rapid key phrase extraction is proposed with the supervised learning techniques and some reasonable and positive results are produced conclusively. It has mainly four stages: Data Preprocessing Stage, Training Stage, Test Stage and Evaluation Stage. TF-IDF, Key Phrase position and Relative length features are the ones that our algorithm uses during the extraction process. As we showed the success of algorithm in the evaluation stage that Laplace metric.

As future work, some additional study may be added to augment the accuracy of the automatic keyphrase program. Adding some extra features to make accuracy higher will be a good improvement in this research area. Also, using some extra or the other learning algorithms will be useful in this arena.

# **VI. REFERENCES**

[1]http://en.wikipedia.org/wiki/Keywords.

[2] Survey of Keyword Extraction Techniques Brian Lott bnlott@cs.unm.edu December 4,2012

[3] Y. B. Wu, Q. Li, Document keyphrases as subjectmetadata: incorporating document key concepts in searchresults, Journal of Information Retrieval, 2008, Volume 11,Number 3, 229-249

[4] S. Jones, M. Staveley, Phrasier: A system for interactivedocument retrieval using Keyphrases, In: proceedings ofSIGIR, 1999, Berkeley, CA

[5] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, E.Frank, Improving browsing in digital libraries withkeyphrase indexes, Journal of Decision Support Systems,2003, 27(1-2), 81-104

[6] B. Kosovac, D. J. Vanier, T. M. Froese, Use of keyphraseextraction software for creation of an AEC/FM thesaurus,Journal of Information Technology in Construction, 2000,25-36

[7] K. Barker, N. Cornacchia, Using Noun Phrase Heads toExtract Document Keyphrases. In H. Hamilton, Q. Yang(eds.): Canadian AI 2000. Lecture Notes in ArtificialIntelligence, 2000, Vol. 1822, Springer-Verlag, BerlinHeidelberg, 40 – 52.

[8] L. F Chien, PAT-tree-based Adaptive Keyphrase Extractionfor Intelligent Chinese Information Retrieval, InformationProcessing and Management, 1999, 35, 501 – 521.

[9] Y. HaCohen-Kerner, Z. Gross, A. Masa, AutomaticExtraction and Learning of Keyphrases from ScientificArticles, In A. Gelbukh (ed.): CICLing 2005. Lecture Notesin Computer Science, 2005, Vol. 3406, Springer-Verlag,Berlin Heidelberg, 657 – 669.

[10] A. Hulth, J. Karlgren, A. Jonsson, H. Boström, AutomaticKeyword Extraction Using Domain Knowledge, In A.Gelbukh (ed.): CICLing 2001. Lecture Notes in ComputerScience, 2001, Vol. 2004, Springer-Verlag, BerlinHeidelberg, 472 – 482.

[11] Y. Matsuo, Y. Ohsawa, M. Ishizuka, KeyWorld: ExtractingKeywords from a Document as a Small World, In K. P.Jantke, A. shinohara (eds.): DS 2001. Lecture Notes inComputer Science, 2001, Vol. 2226, Springer-Verlag,Berlin Heidelberg, 271– 281.

[12] J. Wang, H. Peng, J.-S. Hu, Automatic KeyphrasesExtraction from Document Using Neural Network., ICMLC2005, 633-641

[13] P. D. Turney, Learning algorithm for keyphrase extraction,Journal of Information Retrieval, 2000, 2(4), 303-36

[14] E. Frank, G. Paynter, I. H. Witten, C. Gutwin, C. Nevill-Manning, Domain-specific keyphrase extraction. Inproceeding of the sixteenth international joint conference onartificial intelligence, 1999, San Mateo, CA.

[15] I. H. Witten, G.W. Paynter, E. Frank et al, KEA: PracticalAutomatic Keyphrase Extraction, In E. A. Fox, N. Rowe(eds.): Proceedings of Digital Libraries’99: The FourthACM Conference on Digital Libraries. 1999, ACM Press,Berkeley, CA , 254 – 255.

[16] N. Kumar , K. Srinathan, Automatic keyphrase extractionfrom scientific documents using N-gram filtrationtechnique, Proceeding of the eighth ACM symposium onDocument engineering, September 16-19, 2008, Sao Paulo,Brazil.

[17] Kamal Sarkar, Mita Nasipuri and Suranjan Ghose;A New Approach to Keyphrase Extraction Using Neural NetworksComputer Science and Engineering Department, Jadavpur University, Kolkata-700 032, India

[18]Firat Kalaycilar, Ilyas Cicekli; TurKeyX: Turkish Keyphrase Extractor; Dept. of Computer Engineering Bilkent University;06800 Bilkent Ankara, Turkey

[19] Nagehan Pala, Ilyas ÇiçekliTurkish Keyphrase Extraction Using KEA; Dept. of Computer Engineering Bilkent University;06800 Bilkent Ankara, Turkey

[20] KEA: Practical Automatic Keyphrase Extraction Ian H. Witten,1 Gordon W. Paynter,1 Eibe Frank,2 Carl Gutwin and Craig G. Nevill-Manning3; 1 Dept of Computer Science, University of Waikato,Hamilton, New Zealand., 2Dept of Computer Science, University of Saskatchewan, Saskatoon, 3Canada; Dept of Computer Science,Rutgers University,Piscataway, New Jersey

**THE INTERFACES OF THE RAPID KEY PHRASE AXTRACTION APPLICATION**



