TurKeyX: Turkish Keyphrase Extractor

Fırat Kalaycılar
Dept. of Computer Engineering
Bilkent University
06800 Bilkent Ankara, Turkey
Email: firatk@cs.bilkent.edu.tr

Ilyas Cicekli
Dept. of Computer Engineering
Bilkent University
06800 Bilkent Ankara, Turkey
Email: ilyas@cs.bilkent.edu.tr

***Abstract*—Keyphrases are useful information extracted from documents. They reflect the main ideas of the text. Therefore knowing the list of keyphrases can save substantial amount of time which can be lost during searching for a document about a particular topic. Unfortunately, there are many documents which do not include a list of keyphrases. Thus automatic extraction of keyphrases becomes an important task. In this paper, a method for Turkish keyphrase extraction is explained.**

I. INTRODUCTION

There are huge amount of documents available in the libraries and on the Internet. Searching and finding a relevant document in these media is not an easy task. This requires a successful indexing and categorization mechanism. At the heart of this mechanism, descriptive phrases found in these documents stay. These phrases which reflect the main ideas of a document best are referred as *keyphrases.* They have several uses such as *summarization, indexing, text retrieval* and *document characterization.*

The problem explained in this paper is *automatic extraction of keyphrases* from a given Turkish text document. Since it is an extraction problem, its results are phrases directly taken from the document. However there is a harder task which is called *automatic keyphrase generation* [1]. It is to generate some keyphrases that do not have to appear in the body of the document. In this paper, this more general task is not handled. Our system, TurKeyX, only attacks the problem of keyphrase extraction for Turkish documents.

There are several solutions for automatic keyphrase extraction problem. KEA (Keyphrase Extraction Algorithm) [2] does keyphrase extraction in two stages. First, using training data, a keyphrase extraction model is created. Secondly, using that model, keyphrases of new documents are extracted. For both training and extraction, documents are analyzed to find candidate phrases and the feature values of these phrases are calculated. The features calculated in KEA are *TFxIDF (Term Frequency x Inverse Document Frequency), first occurrence of the term, length of the term* and *node degree*. In training stage, a corpus with manually assigned keyphrases is processed. While model creation, KEA uses Naïve Bayes technique to learn two sets of numeric weights for the feature values of candidate phrases (a set for keyphrases and a set for ordinary noun phrases). In the extraction stage, these weights are used to identify if a candidate phrase is a keyphrase or not.

Nagehan Pala and Ilyas Cicekli implemented KEA for Turkish keyphrase extraction [3]. In order to do so, KEA's original English stemmer and stopwords lists were replaced with their Turkish counterparts. In addition to these changes, they included a new feature which is not originally included in KEA. The new feature *relative length* 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.

Another approach for automatic keyphrase extraction is Turney's GenEx [1]. It has two main components, *Extractor* and *Genitor*. Extractor is a keyphrase extraction algorithm which uses 12 parameters (thresholds) to process a given document. Algorithm finds single stems, scores them and select top scoring ones. Then it finds stem phrases and again scores them. If there are duplicate stem phrases, they are eliminated. The resulting list of stem phrases has no suffixes, so algorithm tries to put suffixes to them. Next, capitalization is done and the final sorted output is reported to the user. The main problem in Extractor is the determination of the 12 parameters. In order to solve this problem, Whitley's Genitor algorithm [6] is used. It is a steady-state genetic algorithm which is used to tune 12 parameters of Extractor. Therefore after training of the system is completed, Genitor is no longer used.

KEA and GenEx are methods which require machine learning approaches. Ken Barker and Nadia Cornacchia propose a different method. They use noun phrase heads to extract document keyphrases, and they call their extractor as B&C [4]. When compared to KEA and GenEx, B&C is a simpler system which exploits the statistics of noun phrases, noun phrase heads and noun phrase lengths. It does not require training corpus. Therefore it is more generic in terms of domain. Their algorithm begins with the calculation of head noun frequencies. Top N head nouns are selected and for each head noun HN, all noun phrases having HN as its head are collected. Then for these phrases, scores are calculated as the product of their frequency and length. K high scoring phrases are reported to user as the list of keyphrases.

TurKeyX is similar to B&C in the sense that it does not require corpus training. It is also based on the statistics of noun phrases and noun phrase heads. However TurKeyX also borrows some feature values computed in KEA and GenEx.

Throughout the rest of the paper, the design and the performance of TurKeyX are explained. The details of TurKeyX are explained in Section II. The performance results

of TurKeyX are given in Section III, and the concluding remarks are given in Section IV.

II. TURKEYX

In order to extract the keyphrases for a given text, TurKeyX processes the given text at different levels. In the first modules, TurKeyX tries to determine candidate keyphrases. Later, it scores the candidate noun phrases depending on their features. A further filtering operation is performed by consulting a feedback mechanism. The details of each module of TurKeyX are explained in the following sub-sections.

A. Part of Speech Tagger

A Turkish part of speech (POS) tagger [7] is the first module of the system. It processes the given Turkish document. The supervised POS tagger gives the most probable POS tag to each word of the input document. Besides assigning tags, it gives the selected morphological analysis of each word. Therefore the output of this module carries important information to be used for noun phrase skimming. The output of the part of speech tagger is in the form of sequence of tokens (words, punctuation etc.) with their morphological analyses and POS tags.

B. Base Noun Phrase Skimmer

A base noun phrase is a non-recursive structure consisting of a head noun and zero or more premodifying adjectives and nouns [4]. For instance, "kırmızı başlıklı kız" (red riding hood) is a base noun phrase, "kırmızı" (red) and "başlıklı" (with hood) are premodifying adjectives and nouns, and "kız" (girl) is the head noun of this phrase. The task of the *base noun phrase skimmer* is to capture these structures from the sequence of tokens which is the output of POS tagger.

Unfortunately, the base noun phrase skimmer of TurKeyX is not an ideal chunker. For example, sometimes it can even create a sequence of adjective and nouns of length 17. Of course, these phrases are not actually noun phrases. However it is possible that long phrases contain some sub-phrases which are suitable noun phrases. For example, skimmer can detect "deprem araştırma enstitüsü ulusal" (earthquake research institute national) as a noun phrase which is a meaningless phrase. However its sub-phrase "deprem araştırma enstitüsü" (earthquake research institute) is a reasonable one. In these cases, the base noun phrase skimmer is extended to create all possible sub-phrases in order not to skip expected noun phrases. Thus, the base noun phrase skimmer module may mark some meaningless phrases as base noun phrases in addition to correct ones. An important point is that the candidate keyphrases will be among the phrases marked as base noun phrases.

During skimming, the module makes use of a noun phrase filter module. The filter informs the skimmer about the stopwords, so the outputs of the skimmer become rational. The output of skimming process is a sequence of candidate noun phrases to be used in the further steps.

C. Feature Extractor

There are six feature values considered for each candidate noun phrase. These six feature values for each candidate noun phrase are computed using the given text. The list of considered features is as follows:

- *Actual noun phrase rate (ANPR):* The frequency of a noun phrase without any changes in its words divided by total number of candidate noun phrases. For example, during calculation of this value "kalem ucu" and "kalemin ucu" are regarded as different phrases.
- *Stem-based noun phrase rate (SNPR):* The frequency of a noun phrase with possible inflections or derivations divided by total number of candidate noun phrases. Here, "kalem ucu" and "kalemin ucu" are accepted to be same phrases.
- *Head noun rate (HNR):* The frequency of a head noun with possible inflections and derivations divided by total number of candidate noun phrases. For HNR, if "uç" is a head noun, then "kırmızı uç" and "kalem ucu" contributes to the frequency of "uç".
- *Noun phrase length (NPL):* Word count of a noun phrase. "kırmızı uçlu güzel kalem" has a length of 4.
- *Noun phrase first occurrence (NPFO):* Reciprocal (multiplicative inverse) of the first occurrence order among the sequence of candidate phrases.
- *Head noun first occurrence (HNFO):* Reciprocal of the first occurrence order of the head noun.

D. Score Calculator

As mentioned above, TurKeyX does not consult machine learning. Therefore there is no classifier which decides if a phrase is keyphrase or not. So as in B&C, the scores of noun phrases are required to distinguish keyphrases. In TurKeyX, a straightforward formula is used to calculate the score for a noun phrase.

$$NPS = K*(ANPR*SNPR*HNR*NPFO*HNFO) + NPL^2 \quad (1)$$

Equation (1) shows that each feature is given equal importance except the NPL feature. Because NPL is not a normalized value and also does not represent a rate or frequency as the others. On the average NPL^2 is equal to 9. The term multiplied by K is a very small value. In order to make the contributions of both terms similar, the small valued term must be multiplied by a coefficient K. Experimentally K is chosen to be 1.6×10^{10}.

In order to obtain (1) many experiments were done and it was accepted as reasonable choice. Of course, this equation is not the perfect one, but among the choices it is the one which yields the best outputs.

For phrases longer than 3, score is divided by a large number such as 10000. This is an extra penalty for very long phrases. By this way, such phrases become very weak candidates.

E. Noun Phrase Filter

There is a long sequence of candidate phrases and it contains redundant and unexpected noun phrases. Therefore this sequence needs filtering. Filter module has several functions.

The first function is the removal of incorrectly extracted noun phrases. There are hand-coded rules to filter incorrect noun phrases. These rules can filter some incorrect noun phrases such as those ending with a number like "deprem 3", consisting of single letter tokens like "t r b", ending with an unacceptable head noun like "onun" and containing an unacceptable word like "evlerden *hangisi*", "yeşil *galiba*".

The second function is the removal of duplicate noun phrases. Sequence can contain same noun phrases in several places. In these cases, filter removes the low scoring duplicates.

Third role of the filter is the selection of high scoring phrase among a noun phrase and its sub-phrases. As mentioned before, skimmer creates all possible sub-phrases from a detected noun phrase. However they increase the size of the noun phrase sequence very much. Therefore after score calculation, only the highest scoring phrases are kept in the sequence, i.e. others are eliminated by the filter. For example "deprem araştırma enstitüsü ulusal" has a length of 4, therefore its score is penalized and consequently "deprem araştırma enstitüsü" with length 3 has a higher score. This means the longer one is removed by the filter.

Fourth filtering is done according to the grammatical case (nominative, accusative, dative etc.) of a noun phrase. Phrases having a case different than nominative are eliminated by the filter.

F. Noun Phrase Sorter and Reporter

After all the elimination, remaining phrases are sorted using their scores calculated by the score calculator. For longer documents, reporter assumes top 10 scoring phrases are keyphrases and for shorter ones, top 5 scoring phrases are keyphrases.

G. Feedback Interpreter

Sometimes reporter's results are not satisfactory. In those cases, feedbacks can be given about the results in order to maximize the extraction precision. There are 2 available feedbacks. These are:

- *Ends with an unacceptable head noun*: This feedback takes the unacceptable head noun into a list. So while skimmer is using the filter module to decide on stopwords, actually it checks if a word is in this list or not.
- *Contains an unacceptable noun*: This feedback is used for single word noun phrases. These words are put into a black list which prevents the skimmer to include that word in a phrase in the future.

These are used to create a stopword list. Instead of using a static list, this mechanism is preferred.

III. EXPERIMENTS AND EVALUATION

A. Corpora

Experiments for TurKeyX are done using two different corpora. First one is a collection of Turkish scientific papers obtained from the online archives of Journal of The Faculty of Engineering and Arhitecture of Gazi University [5]. The corpus was created by Nagehan Pala [3]. It includes 60 papers in text format and its appropriate keyphrases assigned by the authors.

Second corpus is a collection of news articles taken from the web pages of newspapers and news portals. There are totally 30 news articles. Before processing them with TurKeyX, we manually assigned keyphrases to each article.

B. Performance Evaluation

Firstly the comparison for the results of TurKeyX and Nagehan Pala's Turkish Keyphrase Extraction with KEA (KEA-TR) for Gazi University's journal articles is given. TurKeyX originally returns 10 keyphrases for a given long document. However in the tables, only 5 of them are visible.

In Table I, keyphrases which are written in italics are same as author assigned keyphrases, or they contain author assigned keyphrases. For example, "olasılıklı sismik analiz" contains "sismik analiz", and it is assumed to be a correct match.

Table II can be interpreted as the following. TurKeyX (5) refers to the case when the first 5 keyphrases of TurKeyX are considered. Similarly TurKeyX (10) and KEA-TR (5) have similar interpretations.

TABLE I. SAMPLE EXTRACTION RESULT FOR GAZI UNIV CORPUS

Author Assigned	
kırılganlık analizleri	
sismik analiz	
betonarme çerçeve yapılar	
KEA-TR	**TurKeyX**
sismik	yapı
betonarme çerçeve	*kırılganlık analizleri*
çerçeve yapıların	*betonarme çerçeve yapılar*
Sismik Analizi	*olasılıklı sismik analiz*
betonarme çerçeve yapılar	alanında dinamik analiz

TABLE II. NUMERICAL EXTRACTION RESULTS FOR GAZI UNIV. CORPUS USING 60 ARTICLES

	Average Match Amount	**Average Match Rate**	**Average Actual Keyphrase Amount**
TurKeyX (5)	0.90 (54/60)	22.50%	4.00
TurKeyX (10)	1.37 (82/60)	34.25%	4.00
KEA-TR (5)	1.05	26.25%	4.00
KEA-TR (10)	1.42	35.50%	4.00

From the results, it is clear that both methods perform similarly when they are applied to Gazi University corpus.

After comparing two systems, we want to show how TurKeyX performs for the news corpus. Since the news' texts are shorter when compared to Gazi University's corpus' documents, TurKeyX returns a list of 5 keyphrases. Table III shows a sample extraction result for that corpus. Table IV shows the overall results for the whole news corpus.

While obtaining these results, for proper names like "Islam Kerimov", "Kerimov" is regarded as true match because they identically refer to same object/person.

TABLE III. SAMPLE EXTRACTION RESULTS FOR NEWS CORPUS

Author Assigned	TurKeyX
Avrupa İnsan Hakları Mahkemesi	türkiye
AİHM	başvuru
türkiye	yükü
	aihm
	anka

TABLE IV. NUMERICAL EXTRACTION RESULTS FOR NEWS CORPUS USING 30 DOCUMENTS

	Average Match Amount	Average Match Rate	Average Actual Keyphrase Amount
TurKeyX (5)	0.97 (29/30)	29.13%	3.33

So far overall results for both corpora are given. In the rest of this section, effects of feature values and feedbacks are shown.

C. Effects of Feature Values and Feedbacks

To see the effect of a particular feature value, its contribution to score calculation is disabled while the effects of the others remain untouched.

Removal of ANPR, HNR, NPL or NPFO creates similar negative influences on the result set, i.e. final keyphrases are unacceptable. In the absence of HNFO, results are not satisfactory, either. However they are closer to be target keyphrases. On the other hand, the remaining feature value SNPR is disabled, drastic changes are not observed. Therefore ANPR, HNR, NPL and NPFO can be considered as the most important feature components.

Similarly in order to see the effect of feedbacks, they are ignored. Results show that system can still capture almost the same keyphrases. Explanation for this situation is that hardcoded rules of noun phrase filter almost covers all effects of feedbacks. However this does not mean feedback interpreter should be removed. Ideally, there are some words which are unacceptable and still cannot be eliminated by the rules of the filter.

IV. CONCLUSIONS

In this paper, TurKeyX: Turkish Keyphrase Extractor is explained. TurKeyX is a simple keyphrase extraction system which exploits the statistics of noun phrases, noun phrase heads, lengths and first occurrences. It does not use corpus training, so it is independent from the document domain.

It is shown that, a state of the art approach KEA performs similarly when it is compared with TurKeyX using same corpus. There is not a large gap between their results. However the most important contribution is that KEA's performance depends on the training corpus whereas TurKeyX's performance does not depend on any corpus. TurKeyX can be incorporated into any application without any changes.

For this study, in addition to Gazi University's journal corpus, a news corpus is used to see how the extraction system performs for a different domain. Results point that for news domain TurKeyX still shows a good performance.

Finally, the quality of the output depends on the candidate noun phrases. They are obtained using noun phrase skimmer. TurKeyX uses a POS tagger to implement that component. However if there was a successful NP-chunker for Turkish, performance of the system would definitely enhance.

ACKNOWLEDGMENTS

This work is partially supported by The Scientific and Technical Council of Turkey Grant ``TUBITAK EEEAG-107E151".

REFERENCES

[1] P. D. Turney, "Learning algorithms for keyphrase extraction," in Information Retrieval, vol. 2, pp. 303–336, Kluwer Academic Publishers, 2000.

[2] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, "KEA: Practical Automatic Keyphrase Extraction," In Proceedings of the Fourth ACM Conference on Digital Libraries, 1999

[3] N. Pala, and I. Cicekli, Turkish Keyphrase Extraction Using KEA, in: Proceedings of the 22nd International Symposium on Computer and Information Sciences (ISCIS 2007), Ankara, Turkey, 2007.

[4] K. Barker, N. Cornacchia, "Using noun phrase heads to extract document keyphrases" In Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence (LNAI 1822), (Montreal, Canada, 2000

[5] Journal of The Faculty of Engineering and Architecture of Gazi University, Vol. 21 Nr. 1, Nr. 2, Nr. 3, Nr.4 and Vol. 20 Nr. 1, Nr.2, Nr. 3, 2006.

[6] D. Whitley, "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best", In Proceedings of the Third International Conference on Genetic Algorithms, pp.116-121, December 1989.

[7] T. Daybelge, and I. Cicekli, "A Rule-Based Morphological Disambiguator for Turkish", in: Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), Borovets, Bulgaria, 2007, pp: 145-149.