The 2014 Conference on Computational Linguistics and Speech Processing
ROCLING 2014, pp. 175-184

Unsupervised Approach for Automatic Keyword Extraction from Arabic Documents

Arafat Awajan *

Abstract

In this paper, we present an unsupervised two-phase approach to extract keywords from Arabic documents that combines statistical analysis and linguistic information. The first phase detects all the N-grams that may be considered keywords. In the second phase, the N-grams are analyzed using a morphological analyzer to replace the words of the N-grams with their base forms that are the roots for the derived words and the stems for the non-derivative words. The N-grams that have the same base forms are regrouped and their counts accumulated. The ones that appear more frequently are then selected as keywords. An experiment is conducted to evaluate the proposed approach by comparing the extracted keywords with those manually selected. The results show that the proposed approach achieved an average precision of 0.51.

Keywords: Keyword extraction, Keyphrase extraction, Arabic Language, N-gram.

1. Introduction

Keyword extraction is the process of identifying a short list of words or noun phrases that capture the most important ideas or topics covered in a document. Keyword extraction has been used in a variety of natural language processing applications, such as information retrieval systems, digital library searching, web content management, document clustering, and text summarization (Rose et al. 2010). Although keywords are very useful for a large spectrum of applications, only a limited number of documents with keywords are available on-line. Therefore, appropriate tools that can automatically extract keywords from text are increasingly needed with the continually growing amount of electronic textual content available online.

In this paper, an unsupervised two-phase approach for keyword extraction from Arabic

* Princess Sumaya University for Technology – Department of Computer Science, Amman – Jordan
E-mail: awajan@psut.edu.jo

documents is described. The proposed method combines the document's statistics and the linguistic features of the Arabic language to automatically extract keywords from a single document in a domain-independent way. In the first phase, all the N-grams are extracted and those considered as potential candidate keywords are retained. In the second phase, the candidate keywords are analyzed linguistically by a morphological analyzer that replaces each term with its base form, which are the roots of the derived words and the stems of the non-derivative words. The candidate keywords are then grouped in such a way that the keywords extracted from similar roots and stems are put together and their counts accumulated.

This paper is organized as follows. In section 2, we present related works and the main approaches to keyword extraction. Section 3 highlights the main Arabic language features used in our technique. A detailed description of the proposed technique and its two phases provided in Section 4 and Section 5. Section 6 consists of the experimental results and the main findings of the evaluation of the proposed method.

2. Related Work

Existing automatic keyword extraction methods can be divided into two main approaches: supervised and unsupervised (Pudota et al. 2010; Hasan and Ng 2010). In the supervised approach, the keyword extractor is trained to determine whether a given word or phrase is a keyword or not. An annotated set of documents with predefined keywords is always used in the learning phase. All the terms and noun phrases in the text are considered as potential keywords, but only those that match with keywords assigned to the annotated data are selected. The main disadvantages of this approach are its dependency on the learning model, the documents used as the training set, and the documents' domains. Furthermore, training data and learning processes are usually time-consuming (Turney 2000; Turney and Pantel 2010; Frank et al. 1999; Hulth 2003; Hulth 2004).

The unsupervised approach for keyphrase extraction avoids the need for annotated documents. It uses language modeling and statistical analysis to select the potential keywords. A candidate keyword is often selected based on features such as its frequency in the document, the position of its first occurrence in a document, and its linguistic attributes, such as its stem and part-of-speech (POS) tag (Matsuo and Ishizuka 2004; Mihalcea and Tarau 2004; Liu et al. 2009). The unsupervised methods are in general domain-independent and less expensive since they do not require building an annotated corpus.

Keyword extraction algorithms from both approaches have been successfully developed and implemented for documents in the European languages (Rose et al. 2010; Liu et al. 2009; Matsuo et al. 2004). However, despite the fact that Arabic is one of the major international languages making up about 4% of the Internet content, not many studies about extracting

Arabic keywords have been performed. El-Shishtawy and Al-Sammak (2009) presented a supervised method that uses linguistic knowledge and machine learning techniques to extract Arabic keywords. The system uses an annotated Arabic data set of 30 documents from a specific domain, compiled by the authors as a training data set. The keywords from the documents' data set used to evaluate their system were assigned manually.

An unsupervised keyphrase extraction system (KP-Miner) was proposed by El-Beltagy and Rafea (2008). This system was basically developed for the English language and then adapted to work with the Arabic language. Statistical analysis of the texts was conducted in order to determine the most weighted terms. Two main conditions are considered; the first states that a phrase has to have appeared at least n times in the document from which the keywords are to be extracted, and the second condition is related to the position where a candidate keyphrase first appears within an input document. The linguistic analyses performed on the texts are limited to stop word removal and word stemming.

The hypothesis defended in this work is that using the linguistic features of the Arabic language — mainly its rich and complex morphological structure — may present an attractive paradigm to improve the extraction of keywords. The proposed approach is designed to work on a single document without any prior knowledge about its content or domain. Typically, a generic unsupervised keyphrase extractor features two steps; the first is to extract as many candidate words as possible, and the second is to apply the linguistic knowledge of the text language to tune the final list of extracted keywords.

3. The Features of Arabic Language

Arabic is a Semitic language with rich morphology that is a combination of non-concatenative morphology and concatenative morphology. Regarding the concatenative aspect, an Arabic word is composed of a stem, affixes, and clitics. The affixes are concatenative morphemes that mark the tense, gender, and/or number of the word (Al-Sughaiyer and Al-Kharashi 2004). A clitic is a symbol consisting of one to three letters that can be attached to the beginning or the end of a word. It represents another part of speech, such as a preposition, a conjunction, the definite article, or an object pronoun (Habash 2010; Awajan 2007; Diab et al. 2007). In terms of their formation, most of the stems obey non-concatenative rules and are generated according to the root-and-pattern scheme. In general, an Arabic word may be decomposed in its components according to the structure shown in figure 1. For example, the word "واللاعبون", or "and the players" in English, consists of the clitics "و" and "ال", the stem "لاعب", and the postfix "ون". Its stem is generated from the root "لعب", according to the pattern "فاعل". Figure 2 shows the steps for a word formation.

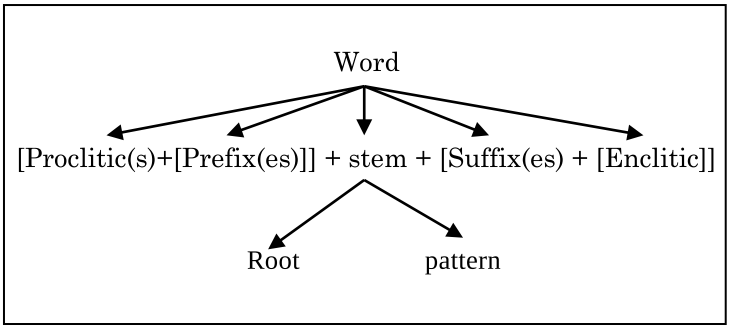
Word
[Proclitic(s)+[Prefix(es)]] + stem + [Suffix(es) + [Enclitic]]
Root
pattern

Figure 1. Arabic derivative word structure

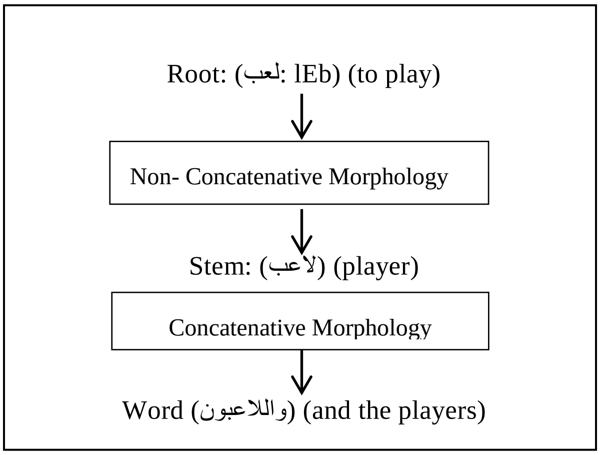
Root: (لعب: lEb) (to play)
Non- Concatenative Morphology
Stem: (لاعب) (player)
Concatenative Morphology
Word (واللاعبون) (and the players)

Figure 2. Arabic word formation (Example)

Arabic words are classified into two categories: derivative words and non-derivative words. The stems of derivative words are generated from the roots according to standard patterns or templates. These standard patterns represent the major spelling rules governing Arabic words. Based on the above, a derivative Arabic word can be represented by its root along with its morphological pattern, and its roots carry its basic conceptual meaning.

Non-derivative words include two sub- categories: fixed words and foreign words. Fixed words are a set of words that do not obey the derivation rules. These words are generally stop words, such as pronouns, prepositions, conjunctions, question words, and the like. The foreign words are nouns borrowed from foreign languages.

The combinatory nature of the Arabic language morphology creates an important obstacle for different natural language processing applications, including keyword extraction. This property, generally known as "data sparseness", results in a large number of words generated from the same root but with different stems (Benajiba et al. 2009). Consequently, the grouping of words according to their surface or stems cannot give keywords that accurately reflect the content of the document.

In order to tackle this problem, we need to conduct a deeper morphological analysis to extract the roots and to consider their properties in order to group related words and increase the weight of those representing the main ideas covered by the text. The linguistic analysis we are proposing will be applied at two different levels of the keyword extraction. The input text is preprocessed to assign each word with its POS in order to detect all the possible N-grams. The detected N-grams are then post-processed to extract the roots, and to group the N-grams generated from the same roots, and to accumulate their weights.

4. N-Gram Extraction

4.1 Part-of-Speech Tagging

This phase consists of several operations: sentence delimiting, tokenization, and POS tagging. The input text is processed to delimit sentences, following the assumption that no keyphrase parts are located separately in two or more different sentences (Pudota et al. 2010). Punctuation marks, such as commas, semicolons, and dots, are used to divide the input documents into sentences.

Tokenization aims at turning a text into a list of individual words or tokens (Manning et al. 2009). As the clitics attached to a word always refer to other entities, such as pronouns, prepositions, conjunctions, and the definite article, a tokenizer is applied to separate all the clitics except the definite article from the word. The tokenizer is repeatedly applied until the word stops changing.

We then assign a POS tag to each token using the Stanford Arabic parser (Green and Manning. 2010). The assigned POS tags are later used to select the possible N-grams, remove the verbs, and remove meaningless terms, such as the stop words.

4.2 N-gram Extraction and Filtering

A keyword is typically a combination of nouns and/or adjectives. Furthermore, the number of terms that are allowed in a keyword is often limited to three words. Thus, each sentence is processed to extract all the possible N-grams that constitute a sequence of adjacent words with a maximum length of three words. All the N-grams that contain verbs, stop words, or clitics are removed. Only the N-grams that have their members labeled with one of the POS tags marking nouns or adjectives are retained. In addition, the unigrams that are not labeled as nouns are removed from the N-gram list. Figure 3 shows the detected unigrams, bi-grams, and trigrams from a sentence.

Input Sentence in Arabic: قام الرئيس الامريكي بزيارة الى المملكة الاردنية الهاشمية

Input Sentence in English: The American president visited the Hashemite Kingdom of Jordan.

Tokenization: قام | الرئيس| الامريكي| ب | زيارة | الى | المملكة | الاردنية | الهاشمية

Unigrams: الرئيس – الامريكي - زيارة - المملكة – الأردنية - الهاشمية

Bi-grams: الرئيس الامريكي - المملكة الاردنية - الاردنية الهاشمية

Tri-grams: المملكة الاردنية الهاشمية

Figure 3. N-Grams Extraction

5. Keywords Selection

5.1 N-gram Normalization

Normalizing N-grams is the process of reducing the words of an N-gram into their base forms. This process will allow the clustering of N-grams carrying the same information, hence reducing the sparseness of the text’s potential keywords. To achieve this objective, a word morphological analyzer is developed based on the Alkhalil Morpho-Syntactic System (Boudlal et al. 2010). It is applied individually to the words on the list of N-grams. The morphological structures produced by the analyzer are used to determine the category of words, derivative or non-derivative. The derivative words are represented by their root along with their morphological pattern, and the non-derivative words are represented by their stem, permitting different N-grams that have common base forms to reinforce each other in scoring and to reduce the number of redundant terms and concepts. Each N-gram is associated with its list of base forms called the normalized N-grams (NNG) at the end of this step.

5.2 N-gram Clustering and Weighting

All the N-grams generated from the same base forms are grouped together, their counts accumulated, and represented by their NNG. A vector representation of the text is produced where each detected NNG and its frequency are listed. In this work, we define the frequency of a normalized N-gram NGi noted Freq (NGi) as the sum of all the N-grams having the same base forms of NGi.

Each normalized N-gram should be assigned a weight that represents its relevance to be selected as a keyword. The keyword frequency and the keyword degree are generally considered for scoring potential keywords (Rose et al. 2010, Mihalcea and Tarau 2004). The weight of a normalized N-gram NGi is given by the following formula:

$$\text{Weight}(\text{NGi}) = \text{Freq}(\text{NGi}) / \sum_{j=1}^{m} (\text{Freq}(\text{NGj}))$$

where m is the number of Normalized N-grams.

As the unigrams are generally more frequent than the bi-grams and bi-grams are more frequent than tri-grams, we need to correct the weight of N-grams by introducing a new measure called score. The N-gram score takes into account the relevance of individual components forming the N-gram. The score of a unigram is equal to its weight since a unigram has one component. The score of other N-grams (bigrams, trigrams, …) is given by the following formula:

$$\text{Score}(\text{NGi}) = \text{Weight}(\text{NGi}) + \sum_{j=1}^{N} (\text{Weight}(\text{Tj})),$$

where the T1, T2,…, TN represent the N roots/stems of the normalized N-gram NGi.

The degree of an N-gram is calculated as the sum of its Weight and the Weights of all the higher structures containing this N-gram. Thus, the degree favors terms occurring frequently in longer candidate keywords, and the score favors the frequent terms regardless of their co-occurrence with other terms.

5.3 Keywords Selection

The list of N-grams is reordered according to their scores since the highest scores determine the potential candidate keywords. The number of extracted keywords is set by the user. The selection of keywords is done according to the following rules.

- If two N-grams have the same score, the longer one will be selected.

- If two candidate keywords have the same number of components and the same score, we select the higher degree.

- If an N-gram is selected, all the possible combinations of its components will be removed from the list of N-grams to guaranty that an extracted keyword will not be included in another one.

The list of keywords is then built by replacing each selected normalized N-gram by the most frequent of its surface N-gram in the original text. Therefore, the list of keywords that will be associated with the document will have more readable form.

6. Experiments and Evaluation

In order to evaluate the performance of the proposed system, an experiment was carried out to test it by comparing the extracted keywords against the manually assigned ones. A collection of 70 journal articles and article abstract selected from six journals and covering different domains was used. The dataset is divided into three groups according to their size [table 1]. The average number of words per article is 3406. Each one of these articles was assigned a list of keywords. The number of keywords varies from 2 to 14, with an average of 5.14 keywords per document. The number of extracted keywords is set to the same number of keywords assigned manually to the documents, so the number of false positive detections and false negative detections will be equal, and the three measures P, R, and F will be identical.

Table 1 shows the main results of the conducted experiment. An average precision of 0.51 was achieved. Since the primary analysis of the dataset showed that only about 73% of the human-generated keywords appear in the document texts, this result can be considered as a good result. The results have shown also that better results are achieved with larger documents.

Table 1: Results

Dataset	Number of Documents	Average of words per article	Precision
1	22	6523	0.56
2	28	3238	0.54
3	20	212	0.41
All	70	3406	0.51

7. Conclusion

This paper proposed an unsupervised two-stage approach for keyword extraction from Arabic texts that avoids the necessity of annotated data. The conducted experiments showed that the proposed method can extract keywords from single documents in a domain-independent way. The linguistic analysis of the texts and the grouping of N-grams according to their linguistic features improve the quality of extracted keywords. An average precision of 0.51 was achieved in despite the fact that that only about 73% of the human-assigned keywords appear in the document texts.

Reference

Al-Sughaier, I., Al-Kharashi, I. (2004). Arabic morphological analysis techniques: A comprehensive survey. Journal of *The American Society for Information Science and Technology (JASIST)*, 55(3), 189-213.

Awajan, A. (2007). Arabic Text Preprocessing for the Natural Language Processing Applications. *Arab Gulf Journal of Scientific Research*, 25(4), 179-189.

Benajiba, Y., Diab, M., Rosso, P. (2009). Arabic Named Entity Recognition: A Feature-Driven Study. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5), 926-934.

Boudlal, A., Lakhouaja, A., Mazroui, A., Meziane, A., Ould Abdallahi, M., Shoul, M. (2010). Alkhalil Morpho Sys: A Morphosyntactic analysis system for Arabic texts, *International Arab Conference on* Information Technology (ACIT). Riyadh, Saudi Arabia.

Diab, M., Hacioglu, K., JURAFSKY, D. (2007). Automatic Processing of Modern Standard Arabic Text. *Chapter in Arabic Computational Morphology*. Springer Ed. 159-179.

El-Beltagy S., & Rafea A. (2008). KP-Miner: A keyphrase extraction system for English and Arabic documents, *Information Systems.* 34(1), 132-144.

El-Shishtawy, T., & Al-Sammak, A. (2009). Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques, In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, The MEDAR Consortium, Cairo, Egypt.

Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G. (1999). Domain-Specific Keyphrase Extraction. Proceedings of *the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 668–673.

Green, S., and Manning, C. D. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In *COLING*- Beijing. 394–402.

Habash, N. (2010). Introduction to Arabic Natural Language Processing. Morgan & Claypool Publishers, USA.

Hasan, K.S., NG, V. (2010). Conundrums in unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. *COLING* 2010, 365-373.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan,

Hulth, A. (2004). Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction. Doctoral dissertation. Department of Computer and Systems Sciences, Stockholm University.

Liu, Z., Li, P., Zheng, Y., Sun, M. (2009). Clustering to Find Exemplar Terms for Keyphrase Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore. 257–266.

Manning, C. D., Raghavan, P., Schütze, H. (2009). Introduction to Information Retrieval. Cambridge University Press.

Matsuo, Y., Ishizuka, M. (2004). Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. International *Journal on Artificial Intelligence Tools,* 13(1), 157-169

Mihalcea, R., Tarau, P. (2004). TextRank: Brining order into texts. In Proceedings of *EMNLP 2004*, Association for Computational Linguistics, Barcelona, Spain. 404-411.

Pudota, N., Dattolo, A., Baruzzo, A., Tasso, C. (2010). A New Domain Independent Keyphrase Extraction System. *Digital Libraries: Communications in Computer and Information Science*, 91, 67-78.

Rose, S., Engel, D., Cramer, N., Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory* edited by Michael W. Berry and Jacob Kogan, John Wiley & Sons, Ltd. 3-20

Turney, P. D. (2000). Learning Algorithm for Keyphrase Extraction. *Information Retrieval*, 2(4), 303-336.

Turney, P. D., & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37, 141-188.