CMPUT 497 Project Draft Report: RAKE - Key Word Extraction Replication

Shouyang Zhou
University of Alberta
Edmonton, Alberta, Canada
shoyang@ualberta.ca

Sharon Hains
University of Alberta
Edmonton, Alberta, Canada
hains@ualberta.ca

Sharif Bakouny
University of Alberta
Edmonton, Alberta, Canada
albakoun@ualberta.ca

1 Introduction

We aim to replicate the main evaluation from the article ”Automatic Keyword Extraction from Individual Documents” by Rose et al. (2010). This paper devises an unsupervised method for keyword extraction titled ”RAKE” and compares it to a previous well performing unsupervised method called ”TextRank”. Details to follow in the evaluation section.

Keyword extraction is the automated process of extracting important words and phrases from a document. The importance of keyword extraction is in application, in information retrieval, feature engineering, and augmenting human classification tasks. At its most basic level, it helps humans process unstructured data in a more efficient and digestible manner. The importance of this replication study is to verify the results of Rose et al. (2010), so we can be more confident in using RAKE as keyword extraction method. Confirming the results of RAKE testing is relevant as keyword extraction is a widely used application as discussed above.

1.1 Input-Output

The overarching output will be a table in the form of Table 1.2 in Rose et al. (2010), please see the appendix for details. This will be a table comparing the performance of RAKE and TextRank variants (parameters) listing the metrics: extracted keywords (total, mean), correct keywords (total, mean), precision, recall, f-measure summarizing the replication-evaluation.

The unit output per trial will be a simple experiment comparing the keywords extracted by RAKE and TextRank variants by correctly extracted keywords, correct keywords, precision, recall, and f-measure over an input text and input sequence of ”truth/reference” keywords.

2 Related Work

The original paper “Automatic Keyword Extraction from Individual Documents” by Rose et al. (2010) is the primary work of interest. As mentioned, they evaluate their method ”RAKE” and compare it with ”TextRank” over two datasets. RAKE requires a set of phrase and content-word delimiters called a stoplist, a list of stopwords. Stop words are punctuation, numbers, conjuctions, and user specified terms which are used to delimit candidate keyword/phrases.

A brief summary of the RAKE algorithm:

1. Split the text into an array of words using the word delimiters.

2. Split the array into sequences contiguous words using stop words and phrase delimiters.

3. Candidate keywords are words in a sequence that are assigned the same position in the text.

4. Assign scores to each keyword candidate using ratio of degree to word-frequency.

5. Keywords that contain stop words:

   (a) A pair of candidate keywords must be adjoined at least twice in the text in the same order.

   (b) Create a new keyword which contains the pair of keywords with interior stopwords between them.

   (c) The new keyword’s score is the sum of the scores of its keywords components.

6. The keywords of the text are the top T keywords from the keyword candidates list.

Rose et al. (2010) also develops methods for stoplist generation. These stoplist generation methods leverage supervised datasets to generate dataset specific (thus domain specific) stoplists.

RAKE was compared to TextRank and seminal supervised learning methods (Hulth, 2003) over a dataset of human keyword annotated scientific paper abstracts originating from Hulth (2003). The performance of RAKE depends on the stoplist used. RAKE was found to outperform all previously used keyword extraction algorithms in precision, efficiency and simplicity when using a domain specific stoplist. Using a generic stoplist, RAKE was found to be no worse performing than TextRank.

Mihalcea and Tarau (2004) discusses TextRank, a graph-based ranking algorithm for keyword extraction, where the importance of a vertex (phrases) is decided by considering global information value of the phrase recursively computed from the entire graph (text). Implementing this algorithm goes as follows:

1. Identify text units and add them as vertices to the graph.

2. Edges of the graph are relations between text units.

3. Iterate the algorithm until convergence below a given threshold.

4. Rank vertices based on their final scores (values).

The authors of TextRank evaluate their method to Hulth (2003) using the same dataset in RAKE, again, originating from Hulth (2003).

We will not be recreating the results of Hulth (2003), however as we are including their results in our evaluation for reference. Rose et al. (2010) describe their method where ”Hulth (2003) compares the effectiveness of three term selection approaches: noun-phrase (NP) chunks, n-grams, and POS tags, with four discriminative features of these terms as inputs for automatic keyword extraction using a supervised machine-learning algorithm.” Rose et al. (2010) noted difficulty in finding training materials used by Hulth (2003).

3 Methodology

Since our inital proposal, we have found implmentations of RAKE and TextRank in python (Sharma (2017), Barrios et al. (2016)). Since these are already availible to us, we will replicate Rose et al.'s (2010) evaluation of the two using third party libraries.

In essence, we will implment an evaluation script that feeds a dataset into these third party libraries to extract then aggregate the resultant metrics. This task will involve data collection, understanding the interface to the RAKE and textrank implementations, preprocessing datasets to be fed into the two methods, and extracting and aggregating the results.

4 Evaluation

As our project is based on replicating the results of an article, we will evaluate the aggregate measures recorded as per the inital study and conduct an error analysis from samples from the replicating evaluation.

Again, the measures reported in the original study by method were: number of extracted keywords, correct number of extracted keywords, precision, recall, and f-measure (f-score). We will compare our recorded measures to that of the original study to what extent are Rose et al.'s (2010) results reproducible. We will consider Rose et al.'s (2010) results reproduceable if the ordinal performance between RAKE and TextRank variants can be verified.

For this reproduction we require three datasets:

1. Hulth (2003)'s dataset of human keyword annotated scientific paper abstracts.

2. Fox's Stoplist, a generic stopword-list.

3. Keyword Adjecency stoplist, a stoplist generated by the authors of RAKE.

Thus far, we have found the dataset used by Hulth (2003) and Fox's Stoplist.

5 Remaining Work

We have collected the referenced datasets and mapped out pseudo code, as mentioned above in our Implementation section, to obtain the key words of each abstract. We have also begun preprocessing our datasets to allow for keyword extraction. We still need to write the code based off of our pseudo code, extract the keywords from our data sets, evaluate the results from the keyword extraction, and discuss the results in our report. The schedule for our remaining work goes as follows:

Task	**Target Date**	**Person Responsible**
Create code to extract keywords	Nov 13	All
Extract keywords from datasets	Nov 17	All
Evaluate results	Nov 24	All
Complete report writing	Dec 6	All

As for the distribution of work currently completed, Shouyang has created the code for our preprocessing of our data sets. Sharif summarized the papers in the Related Works section. Sharon formatted this paper. We all had a hand in writing this draft of the report.

References

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR* abs/1602.03606. http://arxiv.org/abs/1602.03606.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. pages 216–223. https://www.aclweb.org/anthology/W03-1028.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. pages 404–411.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining* page 1–20. https://doi.org/10.1002/9780470689646.ch1.

Vishwas B Sharma. 2017. rake-nltk. https://pypi.org/project/rake-nltk/.

A Table 1.2 from Rose et al. (2010)

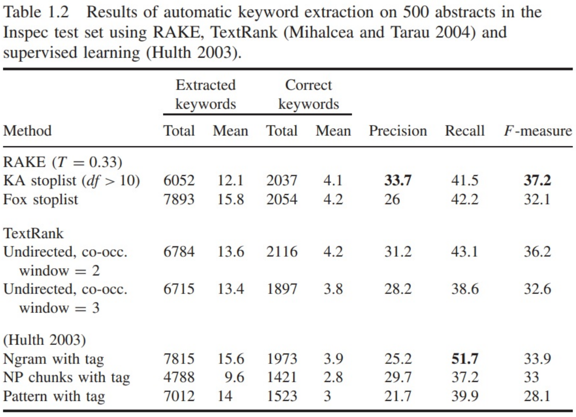

Table 1.2 Results of automatic keyword extraction on 500 abstracts in the Inspec test set using RAKE, TextRank (Mihalcea and Tarau 2004) and supervised learning (Hulth 2003).

	Extracted keywords		Correct keywords				
Method	Total	Mean	Total	Mean	Precision	Recall	*F*-measure
RAKE ($T = 0.33$)							
KA stoplist ($df > 10$)	6052	12.1	2037	4.1	**33.7**	41.5	**37.2**
Fox stoplist	7893	15.8	2054	4.2	26	42.2	32.1
TextRank							
Undirected, co-occ. window = 2	6784	13.6	2116	4.2	31.2	43.1	36.2
Undirected, co-occ. window = 3	6715	13.4	1897	3.8	28.2	38.6	32.6
(Hulth 2003)							
Ngram with tag	7815	15.6	1973	3.9	25.2	**51.7**	33.9
NP chunks with tag	4788	9.6	1421	2.8	29.7	37.2	33
Pattern with tag	7012	14	1523	3	21.7	39.9	28.1

Figure 1: Table 1.2 from Rose et al. (2010)