# Automatic Keyword Extraction: An Ensemble Method

**Conference Paper** · December 2017

DOI: 10.1109/BigData.2017.8258552



**2 authors:**

Tayfun Pay
CUNY Graduate Center
**6** PUBLICATIONS **5** CITATIONS
SEE PROFILE

Stephen Lucci
**10** PUBLICATIONS **10** CITATIONS
SEE PROFILE

# Automatic Keyword Extraction: An Ensemble Method

Tayfun Pay
*Computer Science Department*
*Graduate Center of New York*
*New York, USA*
*tpay@gradcenter.cuny.edu*

Stephen Lucci
*Computer Science Department*
*City College of New York*
*New York, USA*
*lucci.stephen@gmail.com*

***Abstract*—We design and analyze an ensemble method for automatically extracting keywords from single documents. The automatic keyword extractors that we use in our approach are: TextRank [1] RAKE [2] and TAKE [3]. Each one of these automatic keyword extractors provides a set of candidate keywords for the ensemble method. Our approach then prunes this set of candidate keywords by applying a filtering heuristic and then recalculates their scores according to prescribed metrics. Then, dynamic threshold functions are applied to select a set of keywords for a given document. We used the data set in [4] to test the accuracy of our approach. We obtained a better overall performance when compared to each one of the individual keyword extractors that we used in constructing our ensemble method.**

***Keywords*-data mining; text mining; text analysis;**

## I. Introduction

Recently, ensemble based machine learning methods have received much attention in the big data field [5] [6] [7] [8] because they achieve better results. An ensemble based machine learning method combines several machine learning methods, wherein each method is executed independently with the same input and the output is then decided upon by combining their outcomes. Results are obtained either via hard-voting or soft-voting, where the former takes a majority vote while the latter averages the probabilities. It has been observed that the accuracy of ensemble methods is often better than any single method used in its construction. This is especially true when these machine learning methods are independent from each other, such that they make uncorrelated errors.

There has also been growing interest in designing keyword[1] extractors that provide more relevant keywords [9] [10] [11] [12]. Extracting more meaningful keywords is important in many aspects of big data such as classification, data-mining, indexing and text-analysis. Furthermore, extracting more relevant keywords basically improves the subsequently utilized machine learning methods.

In this light, we decided to construct an ensemble of automatic keyword extractors by using the following approaches: TextRank [1], RAKE [2] and TAKE [3]. The way in which each keyword extractor produces a set of candidate keywords for a given document is different in each case, thus yielding a broader pool of candidate keywords for the ensemble method. After applying a filtering heuristic and recalculating the candidate keyword scores, the ensemble method uses dynamic threshold functions to select a set of keywords for the given document. On the data set from [4], we were able to achieve a higher f-measure compared to any of the automatic keyword extractor methods used in the construction of our ensemble method.

[1]We use the words, keyword and keyphrase interchangeably although some authors refer to the former as having a single word and the latter as having more than one word.

## II. Our Methods

We first define the parameters of the automatic keyword extractors that we use. RAKE and TAKE use the fox-stop list that was introduced in [13]. TAKE and TextRANK utilize the NLTK library [14]. And TextRank is set to a co-occurrence window of 2. The details of how these automatic keyword extractors work can be found in the corresponding articles.

For each automatic keyword extractor used in our ensemble method, we strip them of their threshold functions. Then for a given input, each of these automatic keyword extractors provides a set of candidate keywords, along with their scores, to the ensemble method. These scores are then normalized. This is accomplished by dividing each score by the highest score within the set of candidate keywords provided by each automatic keyword extractor. The result is a set of candidate keywords per automatic keyword extractor with a possible score between 0 and 1, inclusive of 1.

We should note that there are several reasons why the threshold functions are removed. For instance, after applying their respective threshold functions, some candidate keywords might be discarded, but these discarded keywords might also be extracted by another method. In another scenario, the score of some candidate keyword extracted by one automatic keyword extractor might be higher than the score of another, but it might not be selected because its score could be less than the one set by the corresponding threshold function. It is therefore best to take into consideration all of the candidate keywords before the respective threshold functions are applied within each automatic keyword extractor.

In the next stage, a filtering heuristic is applied, which is similar in nature to the one that was used in [3]. We remove any candidate keyword that is extracted by a single automatic keyword extractor and consists of a single word.

In the stage that follows, the candidate keyword scores are recalculated in the following manner. First, the scores of candidate keywords that were extracted by more than one automatic keyword extractor are added together. Second, this sum is multiplied by the total number of automatic keyword extractors that extracted them.

We chose to boost the score of the ones that are extracted by more than one automatic keyword extractor because their cumulative score might still be too low to pass the threshold function in the next stage. This can occur even when they have been extracted by all of the automatic keyword extractors.

In the last stage, dynamic threshold functions are applied; these are the same as the ones used in [3]. In one case, the overall mean is calculated and in the other case the median is taken. Then, any candidate keyword that scores higher than the mean or the median is extracted as a keyword for the given document.

## III. Data Set

The data set used was introduced in [4], and subsequently used in [1], [2] and [3] for evaluating their automatic keyword extraction methods, which are the methods that we utilized in constructing our ensemble method. This data set contains 2000 titles and abstracts for journal papers from Computer Science and Information Technology. These items are divided into a training set, validation set and testing set that contain 1000 and two sets of 500 documents, respectively. As our ensemble method is unsupervised, we only used the 500 abstracts from the testing set.

We calculated the following parameters: extracted keywords, correct keywords, precision, recall and f-measure. Precision provides us the percentage of extracted keywords that are correct. $Precision$ = (correct/extracted) Recall provides us the percentage of manually assigned keywords that are extracted. $Recall$ = (correct/manually-assigned) As previously noted in [4], both precision and recall are equally important so that they are given the same weight. $F - Measure$ = (2*precision*recall/precision+recall)

There are 4912 manually assigned keywords of which only 3837 are present in the titles and abstracts. We use the total number of manually assigned keywords in the calculation of recall and f-measure, it was done this way in [1], [2] and [3]. Therefore, the highest obtainable recall for this data set is 78.1.

## IV. Analysis

Tables I and II illustrate the performance of our ensemble method along with the automatic keyword extractors that were used in its development. Our ensemble method has the highest recall as well as the highest f-measure compared to any of the individual automatic keyword extractors when either the mean or the median threshold function is utilized. This also yielded the highest number of correct keywords extracted compared to any of the previously examined approaches.

The only limitation is with precision, where TAKE with the corresponding threshold functions has a higher precision than the ensemble method, but a much lower recall. This is normal with respect to how ensemble methods work. Clearly, each method contributed to the ensemble method certain keywords that were different than one another; and some of these keywords matched the manually assigned keywords and some did not. This increased the recall, but brought down the precision at the same time.

Table I
Precision Recall and F-Measure for Data Set 1

| $method$ | $precision$ | $recall$ | $f\text{-}measure$ |
|---|---|---|---|
| Ensemble - (T=Mean) | 46.7 | 50.9 | 48.7 |
| Ensemble - (T=Median) | 42.1 | 55.9 | 48.0 |
| TAKE - (T=Mean) [3] | 50.4 | 33.7 | 40.4 |
| TAKE - (T=Median) [3] | 44.3 | 46.9 | 45.6 |
| RAKE - (ka-stoplist) [2] | 33.7 | 41.5 | 37.2 |
| RAKE - (fox-stoplist) [2] | 26.0 | 42.2 | 32.1 |
| TextRank (UnD. w=2) [1] | 31.2 | 43.1 | 36.2 |
| TextRank (UnD. w=3) [1] | 28.2 | 38.6 | 32.6 |

Table II
Extracted and Correct Keywords for Data Set 1

| $method$ | $extracted$ | $correct$ |
|---|---|---|
| Ensemble - (T=Mean) | 5353 | 2501 |
| Ensemble - (T=Median) | 6523 | 2746 |
| TAKE - (T=Mean) [3] | 3279 | 1653 |
| TAKE - (T=Median) [3] | 5197 | 2304 |
| RAKE - (ka-stoplist) [2] | 6052 | 2037 |
| RAKE - (fox-stoplist) [2] | 7893 | 2054 |
| TextRank (UnD. w=2) [1] | 6784 | 2116 |
| TextRank (UnD. w=3) [1] | 6715 | 1897 |

## V. Conclusion

We presented an ensemble method for automatically extracting keywords from single documents with better overall performance than the methods utilized in its development. We hope to extend our studies to other data sets that are different in form as well as length. We believe that there is a need for better keyword extractors because the quantity of data being collected in our society is growing exponentially. It therefore becomes critical that more quality keywords be extracted to facilitate greater efficiency in the handling of these vast repositories.

## REFERENCES


[1] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts." *Association for Computational Linguistics.*, 2004.

[2] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents." *Text Mining.*, pp. 1–20, 2010.

[3] T. Pay, "Totally automated keyword extraction," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3859–3863.

[4] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge." *Proceedings of the 2003 conference on empirical methods in natural language processing.*, pp. 216–223, 2003.

[5] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: a survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[6] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 23, 2017.

[7] Z. Wu, W. Lin, Z. Zhang, A. Wen, and L. Lin, "An ensemble random forest algorithm for insurance big data analysis," in *Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on*, vol. 1. IEEE, 2017, pp. 531–536.

[8] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[9] S. K. Bharti, K. S. Babu, A. Pradhan, S. A. Devi, T. E. Priya, E. Orhorhoro, O. Orhorhoro, V. Atumah, E. Baruah, P. Konwar *et al.*, "Automatic keyword extraction for text summarization in multi-document e-newspapers articles," *European Journal of Advances in Engineering and Technology*, vol. 4, no. 6, pp. 410–427, 2017.

[10] N. Giamblanco and P. Siddavaatam, "Keyword and keyphrase extraction using newton's law of universal gravitation," in *Electrical and Computer Engineering (CCECE), 2017 IEEE 30th Canadian Conference on*. IEEE, 2017, pp. 1–4.

[11] F. Yang, Y.-S. Zhu, and Y.-J. Ma, "Ws-rank: Bringing sentences into graph for keyword extraction," in *Asia-Pacific Web Conference*. Springer, 2016, pp. 474–477.

[12] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, "An overview of graph-based keyword extraction methods and approaches," *Journal of information and organizational sciences*, vol. 39, no. 1, pp. 1–20, 2015.

[13] C. Fox, "A stop list for general text." *ACM SIGIR Forum.*, vol. 24, pp. 19–21, 1989.

[14] S. Bird and E. Loper, "Nltk: the natural language toolkit." *ETMTNLP 02 Proceedings of the ACL-02.*, vol. 1, pp. 63–70, 2002.