The 2014 Conference on Computational Linguistics and Speech Processing
ROCLING 2014, pp. 110-124

Collaborative Ranking between Supervised and Unsupervised Approaches for Keyphrase Extraction

Gerardo Figueroa*, Yi-Shin Chen*

Abstract

Automatic keyphrase extraction methods have generally taken either supervised or unsupervised approaches. Supervised methods extract keyphrases by using a training document set, thus acquiring knowledge from a global collection of texts. Conversely, unsupervised methods extract keyphrases by determining their relevance in a single-document context, without prior learning. We present a hybrid keyphrase extraction method for short articles, HybridRank, which leverages the benefits of both approaches. Our system implements modified versions of the TextRank (Mihalcea and Tarau, 2004)—unsupervised—and KEA (Witten *et al.*, 1999)—supervised—methods, and applies a merging algorithm to produce an overall list of keyphrases. We have tested HybridRank on more than 900 abstracts belonging to a wide variety of subjects, and show its superior effectiveness. We conclude that knowledge collaboration between supervised and unsupervised methods can produce higher-quality keyphrases than applying these methods individually.

Keywords: Keyword extraction, Keyphrase extraction, Hybrid approach, Supervised methods, Unsupervised methods

1. Introduction

Keyphrases—also called keywords[1]—are highly condensed summaries that describe the contents of a document. They help readers know quickly what a document is about, and are generally assigned by the document's author or by a human indexer. However, with the massive growth of documents on the Web each day, it has become impractical to manually assign keywords to each document. The need for software applications that automatically assign keywords to documents has therefore become necessary.

* Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan
E-mail: {gerardo.ofc, yishin}@gmail.com

[1] A keyphrase is a phrase composed of one or more keywords. We will use the terms keyphrase and keyword interchangeably in this paper.

In this work we apply efficient and effective practices from supervised and unsupervised methods to produce a hybrid system *HybridRank*. On the supervised side, we implement an extension of the Naïve Bayes classifier originally proposed in KEA (Witten *et al.*, 1999). This classifier has shown to be practical to implement and can be extended for improved effectiveness. On the unsupervised side, we apply the well-known TextRank (Mihalcea and Tarau, 2004) algorithm with some modifications. TextRank is similarly practical to implement, and can effectively extract keyphrases from texts regardless of their size or domain.

Each method contributes by providing a list of keyphrases for a particular text, sorted by their rank or relevance as seen from each approach. Finally, a collaborative algorithm is executed, in which the two keyphrase lists are merged to create an overall list of keyphrases for that text. The merging algorithm thus takes into account the ranks given by both approaches to each keyphrase and produces a final, collaborative score reflected by these ranks.

We have tested HybridRank on a large number of abstracts belonging to scientific papers across different domains. The results of our experiments show the effectiveness of the proposed method and of the improvements made to the KEA and TextRank algorithms. Our system obtained a higher precision and recall than both KEA and TextRank in most cases, and obtained a higher precision and recall than at least one of these two methods in all the cases. The evaluation of our system also shows how knowledge from supervised and unsupervised approaches can be shared to produce keyphrases of better quality.

2. Related Work

Recent work on the automatic generation of keyphrases has been categorized as either *supervised* or *unsupervised*.

Supervised methods for keyphrase extraction, in essence, make use of training datasets—a large corpus consisting of texts and their corresponding (previously assigned) keyphrases—to classify candidate terms as keyphrases. Two traditional methods in this category are KEA (Witten *et al.*, 1999) and GenEx (Turney, 2000). KEA uses a Naïve Bayes classifier constructed from two features extracted from phrases in documents: the TFIDF and the relative position of the phrase. GenEx uses a steady-state genetic algorithm to build an equation consisting of 12 low-level parameters. Even though KEA and GenEx perform similarly well, KEA has shown to be more practical to implement, and has served as the base for other supervised keyphrase extraction methods (Turney, 1999; Hulth, 2003; Nguyen and Kan, 2007).

Other innovative supervised approaches have been proposed in recent years, ranging from the application of neural networks (Jo, 2003; Wang *et al.*, 2006; Jo *et al.*, 2006; Sarkar *et* *al.*, 2010) to conditional random fields (Zhang, 2008). Yih *et al.* (Yih *et al.*, 2006) proposed a multi-class, logistic regression classifier for finding keywords on web pages.

Unsupervised methods for keyphrase extraction rely solely on implicit information found in individual texts. Simple approaches are based on statistics, using information such as term specificity (Kireyev, 2009), word frequency (Luhn, 1957), n-grams (Cohen, 1995), word co-occurrence (Matsuo and Ishizuka, 2004) and TFIDF (Salton *et al.*, 1975). Other approaches are graph-based, where a text is converted into a graph whose nodes represent text units (e.g. words, phrases, and sentences) and whose edges represent the relationships between these units. The graph is then recursively iterated and *saliency scores* are assigned to each node using different approaches.

Mihalcea and Tarau (Mihalcea and Tarau, 2004) developed TextRank, a graph-based ranking model that applies the PageRank (Brin and Page, 1998) formula into texts for assigning scores to phrases and sentences. Wan *et al.* (Wan *et al.*, 2007) proposed a method that fuses three kinds of relationships between sentences and words: relationships between words, relationships between sentences, and relationships between words and sentences. Wan and Xiao (Wan and Xiao, 2008) also developed CollabRank, which improves the keyphrase extraction task by making use of mutual influences of multiple documents within a cluster context.

To our knowledge, all previous work has been either supervised or unsupervised. Supervised methods have the advantage of learning from an already classified collection of documents in order to find keyphrases for a new document, but in essence make no analysis of individual text structure as done by unsupervised methods. HybridRank leverages the benefits of both approaches for keyphrase extraction, applying a supervised keyphrase extraction algorithm (KEA) and an unsupervised graph-based algorithm (TextRank).

3. Background

HybridRank makes use of two well-known and effective keyphrase extraction methods: KEA (Witten *et al.*, 1999) and TextRank (Mihalcea and Tarau, 2004). Each of these methods extracts a list of keyphrases ranked according to each method's approach. A final list of keyphrases is constructed from the collaboration between these two methods and the application of a merging algorithm.

This section will explain the general frameworks for the KEA and TextRank algorithms. The modifications made for these two methods in our work will be described in Section 4. For briefness purposes, we present only a brief explanation of each algorithm, and suggest the reader to refer to the original papers for more details.

3.1 The KEA Algorithm

The KEA algorithm consists of a Naïve Bayes classifier that ranks phrases in order of their probability of being keyphrases as learned from a training document set. KEA is divided into four stages: *candidate phrase generation*, *feature extraction*, *training* and *ranking*.

3.1.1 Candidate phrase generation

The first stage in the KEA algorithm is the selection of phrases that are suitable for training and extraction. To avoid overfitting, this filtering process is applied on both the training document set and the input text to be analyzed.

3.1.2 Feature extraction

The features extracted from the candidate phrases generated in the previous stage are the heart of the KEA algorithm; they serve as the learning base for the Naïve Bayes classifier and are used for the extraction of keyphrases. The features originally extracted by Witten *et al.* (Witten *et al.*, 1999) for each phrase in their KEA algorithm were the *TFIDF* and the *relative position* in the text.

3.1.3 Training

The training stage uses the training document set, which is composed of a collection of documents with their manually-assigned keyphrases. First, phrases are generated from each document in the set. The features for each phrase are then extracted and stored in a training model.

3.1.4 Ranking

With the model having been trained, the Naïve Bayes classifier can extract keyphrases from a new text by first selecting its candidate phrases and then extracting each phrase's features. The model determines the probability of each phrase being a keyphrase using Bayes' formula with the two extracted features.

The probability that a phrase is a keyphrase given that it has TFIDF T and relative position R is then calculated as:

$$P(k \mid T, R) = \frac{P(T \mid k) \cdot P(R \mid k) \cdot Y}{Y + N}, \qquad (1)$$

where $P(T \mid k)$ is the probability that a keyphrase has TFIDF score T and $P(R \mid k)$ is the probability that it has relative position R. Y is the number of phrases that were manually assigned as keyphrases in the training document set and N is the number of phrases that were not. An expression similar to equation (1) is used to calculate the probability that a phrase is *not* a keyphrase ($P(\neg k \mid T, R)$).

The overall probability that a phrase is a keyphrase is then calculated with the following formula:

$$P = \frac{P(k \mid T, R)}{P(k \mid T, R) + P(\neg k \mid T, R)} \qquad (2)$$

The phrases are finally sorted in descending order of their probability scores.

3.2 The TextRank Algorithm

The TextRank algorithm was proposed by Mihalcea and Tarau (Mihalcea and Tarau, 2004). It is a graph-based, unsupervised method for keyphrase extraction. We have divided the TextRank algorithm into two stages to allow an easier comparison with our modifications: *graph construction* and *phrase ranking*.

3.2.1 Graph construction

The first step carried out in the TextRank algorithm is the construction of a graph that represents a text. The resulting graph is an interconnection of words and phrases – the vertices – with significant relations – the edges.

3.2.2 Ranking

With the constructed graph in hand, a recursive algorithm is applied on it which assigns scores to each node on the graph on each iteration until convergence is reached. This algorithm is derived from Google's PageRank (Brin and Page, 1998), which determines the importance of a vertex within a graph by recursively taking into account global information. In other words, the score of one vertex in the graph will affect the scores of all vertices connected to that vertex, and vice-versa.

Before starting the recursive ranking algorithm, all vertices in the graph are initialized with a score of 1. Next, the algorithm is run on the graph for several iterations until it converges within a certain threshold. In each iteration, the original PageRank formula is calculated for each vertex V_i in graph G, as follows:

$$S(V_i) = (1-d) + d \cdot \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j), \quad (3)$$

where $In(V_i)$ is the set of vertices that point to V_i, $Out(V_i)$ is the set of vertices that V_i points to, and d is a damping factor, which is usually set to 0.85.

With final scores assigned to each vertex, they are sorted in descending order of this score.

4. Framework

HybridRank is divided into four main components:

1. **Preprocessing**
2. **Supervised Ranking**
3. **Unsupervised Ranking**
4. **Merging**

This section will describe the *Preprocessing* and *Merging* components in detail. For the *Supervised ranking* and *Unsupervised ranking* components, only the specific modifications made in our work will be detailed.

4.1 Preprocessing

All documents in the training document set, as well as the input text, are cleaned before being processed by the other components. The following steps are performed in this stage:

1. HTML tags are removed.
2. All non-alphanumeric characters are removed, with the exception of punctuation marks relevant to text structure and word meaning.
3. The cleaned text is sent to the supervised and unsupervised components.

4.2 Supervised Ranking

The supervised component of our system consists of a modified and extended version of the KEA algorithm proposed by Witten *et al.* This section will describe the modifications we made in each of the stages of KEA.

4.2.1 Candidate phrase generation

The way candidate phrases are selected in HybridRank has some variations from the procedure followed by the original KEA method. We have carefully inspected the training document set and have used this knowledge to construct a more effective filter for phrase selection (as is later shown in the experimental evaluation). The following procedure is carried out:

1. Phrases composed of 1 to 4 words are extracted from each sentence when they comply with the following criteria:

   a. They do not contain any of a list of 539 predetermined stopwords.

   b. They are composed of nouns, adjectives and/or verbs in their gerund or past participle forms.

   c. They do not contain words with less than 3 letters.

   d. They do not contain words composed only of numbers and/or other non-letters.

   e. They do not end with an adjective.

   f. One-word phrases cannot be an adjective or a verb.
2. Each word in the extracted phrases is then converted to its stemmed form.
3. The phrases are passed as candidate phrases to the Feature Extraction stage.

4.2.2 Feature extraction

We have included two additional features to the learning scheme as proposed in other works: the *keyphrase frequency* in the whole collection of texts (Frank *et al.*, 1999) and the *PoS tag pattern* (Hulth, 2003). Adding these two features produced better overall results in our experiments.

Keyphrase frequency

The keyphrase frequency of phrase P in document D is the number of times P is manually assigned as a keyphrase in the training document set G, excluding D.

PoS tag pattern

The PoS (Part-of-Speech) tag pattern of a phrase P is the sequence of PoS tags that belong to P. These tags are assigned to each word in P using a Part-of-Speech Tagger.

4.2.3 Training

Unlike the original KEA method, we do not discretize real-valued features (TFIDF and relative position) into numeric ranges; we instead round these values to one decimal place. Experiments with both discretization tables and rounding to one decimal gave similar results, so we decided to use rounding due to its simpler implementation and faster performance.

4.2.4 Ranking

With the two additional features (keyphrase frequency and PoS tag pattern) used in HybridRank, an expression similar to equation (1) can be constructed. The probability that a phrase is a keyphrase using all four features would then be calculated as:

$$P(k \mid T, R, S, F) = \frac{P(T \mid k) \cdot P(R \mid k) \cdot P(S \mid k) \cdot P(F \mid k) \cdot Y}{Y + N}, \tag{4}$$

where $P(S \mid k)$ is the probability that it has PoS tag pattern S and $P(F \mid k)$ the probability that it has keyphrase frequency F. An expression similar to equation (4) is used to calculate the probability that a phrase is *not* a keyphrase ($P(\neg k \mid T, R, S, F)$).

The TFIDF and relative position values are rounded to one decimal place in both the trained model and in the current phrase. Since the keyphrase frequency is a non-negative integer, no rounding is performed. Finally, the PoS tag pattern value has to be an exact string match with the one in the trained model.

4.3 Unsupervised Ranking

The unsupervised ranking component of HybridRank is an implementation of the TextRank algorithm proposed by Mihalcea *et al.* for keyphrase extraction. This section will detail the configuration used in our system for the first stage (graph construction) of the TextRank algorithm. No modifications were made to the ranking stage described in Section 3.2.2.

4.3.1 Graph construction

The parameters we have used for the graph construction in our implementation of TextRank presented the best results in our experiments. The following configuration was used:

- The graph is unweighted and undirected.
- Two types of vertices are added to the graph: words and phrases.
- Maximum phrase size is 4 words; they can only be composed of nouns and adjectives.
- The words added to the graph and those in the phrases cannot be any of the 539 predetermined stopwords.
- The relation between words and phrases is the co-occurrence, i.e. the maximum distance

(in words) between two text units. The value used for co-occurrence is 2.

4.4 Merging

The merging component is the core of HybridRank. Once the two keyphrase lists are generated by KEA and TextRank, they are combined into a single list using a merging algorithm. The overall list is the result of the collaboration between a supervised and an unsupervised approach for keyphrase extraction.

The two main stages in the merging component are *keyphrase list merging* and *post-processing*. We illustrate the procedure with an example for easier understanding.

4.4.1 Keyphrase list merging

The first step performed in the merging stage is to add missing keyphrases to each keyphrase list, which results in two lists of the same size and with the same keyphrases, but in different order. In other words, keyphrases that appear in the KEA list which are not in the TextRank list are appended to the TextRank list, and vice-versa. Missing keyphrases are added to each list in the same order of their original list; their corresponding scores are marked with a flag to indicate that these phrases were not in that list before.

Next, a reordering of the two lists is done by giving more priority to those phrases that appear in both lists. Assuming that the two lists are already sorted, the reordering is done by applying the following algorithm to each list L:

```
$reorderedK$ = {}
$existentK$ = {}
$inexistent K$ = {}
for each phrase $P$ in $L$ $P$ do
    if exists in both lists then
        $existentK$ .append( $P$ )
    else
        $inexistent K$ .append( $P$ )
    end if
end for
$reorderedK$ .append( $existentK$ )
```

```
$reorderedK$ .append( $inexistent K$ )
$L \leftarrow reorderedK$
```

The previous algorithm partitions each list into two sections, leaving phrases that appear on both lists on top, and phrases that only appear in one list on the bottom. It is worth pointing out that the original order of the phrases is maintained in each partition.

Finally, the two keyphrase lists are merged into a single list based on the order in which each phrase appears in both lists. Given phrase P with position i in the KEA list and with position j in the TextRank list, three different merging methods can be used to assign an overall position k to P :

- **Average:** $k = (i + j)/2$
- **Min:** $k = Min(i, j)$
- **Max:** $k = Max(i, j)$

Once the new HybridRank position k has been calculated for every phrase in the text, the phrases are sorted according to this new position. If two phrases have the same value for k , as it often occurs, then a tie-breaker is used. The tie-breakers have the following precedence: KEA score, TextRank score, TFIDF value, and finally alphabetical order.

4.4.2 Post-processing

In the final stage of HybridRank, a post-processing filter is applied on the final list of keyphrases. First, any phrase that is a subphrase of a higher-ranking phrase is removed from the list. For example, if the phrase *bass diffusion* has a higher ranking than the phrase *bass*, then the latter is eliminated.

Second, any phrase that exists in a predetermined *stop-phrase* list is removed. The stop-phrase list is a list of words and phrases that will rarely or never be keyphrases by themselves. We have identified 28 stop-phrases, which consist of frequent nouns and noun phrases found in the training documents that were never assigned as author keyphrases. These phrases are different to stopwords in the way that when combined with other words they may become keyphrases. Stopwords, on the other hand, are removed in a previous stage because they will rarely or never be part of a keyphrase. For example, the words *research* and *method* are stop-phrases and not stopwords, because they are too general to be keyphrases, unless combined with other word(s), such as in *photonics research* or *kernel method*.

5. Experiments

5.1 The Corpora

Two different document collections were used for our experiments: the *IEEE Xplore* collection (1,606 documents) and the *Hulth 2003* collection (2,000 documents). The documents consist of abstracts in English from journal and conference papers of various disciplines with their corresponding, manually-assigned keyphrases. Of the total number of abstracts, 1,822 were used for training (to construct the trained model), 917 for testing, and 867 for validation (to evaluate different parameters in the methods used and select the values with the best performance); this assignation was made by random sampling.

Some statistics relevant to the analysis of our experiments were extracted from the collections used. The statistics show that – in general – only 51% of the manually-assigned keyphrases are actually contained in the abstract text in their stemmed forms. With this knowledge, it can be deducted that the precision of any keyphrase extraction method will rarely surpass this percentage on these corpora, which presents a difficulty for adequate evaluation. For the purpose of carrying out a fairer evaluation, a *utopian subset* was selected from the testing set. Each of this subset's abstracts must contain at least one of the manually-assigned keyphrases in the text. Additionally, an average of 7 keyphrases were manually assigned for each abstract by either authors or other human annotators, which correspond to roughly 6% of the total number of words per abstract.

5.2 Experimental Setup

For evaluating the performance of HybridRank, we have performed experiments on the utopian subset using two other keyphrase extraction methods: KEA and TextRank. HybridRank has been separated into three different merging methods, which we evaluate individually: *average*, *min* and *max*.

To further break down our evaluation, we have performed experiments using the original procedures stated in the KEA and TextRank papers, and compared their performance with our modified versions. Additionally, we separated the evaluation of the KEA and HybridRank methods by using two different feature sets for the Naïve Bayes classifier: the *Base Feature Set (New)* and the *PoS Tag Feature Set*.

Base Feature Set (New)
Only the TFIDF, relative position and keyphrase frequency are taken into account when calculating equation 4 in Section 4.2.4.

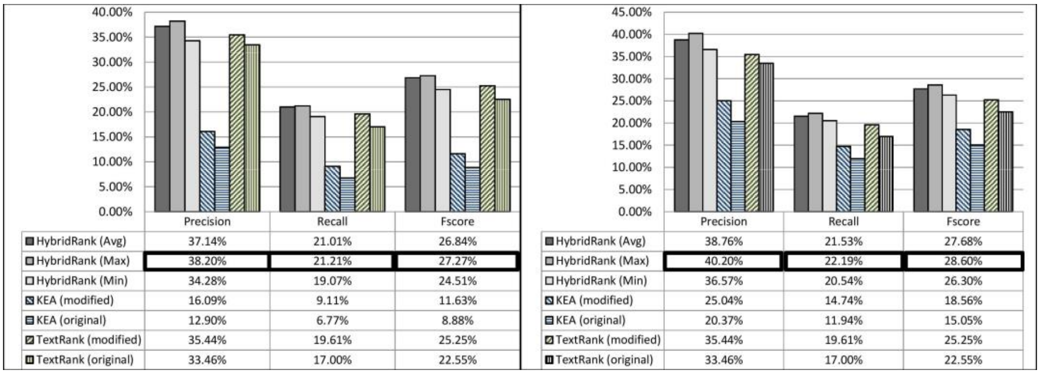
PoS Tag Feature Set
Only the TFIDF, relative position and PoS tag pattern are taken into account when calculating equation 4 in Section 4.2.4.

The three measures used in our evaluation were the precision, recall and F-score. We compare the output keyphrases of each method with those in the manually-assigned list; the keyphrases in each list are previously stemmed. The number of keyphrases extracted per abstract corresponds to 6. This way of selecting the number of output keyphrases presented the best results.

5.3 Evaluation and Discussion

The results for the Hulth 2003 dataset are shown in Figure 1. For this dataset, HybridRank obtained the highest precision, recall and F-score when using the Max merging method. This best performance was obtained when applying either the Base Feature Set (New) or the PoS Tag Feature Set on KEA. It can also be observed in Figure 1 that our modified versions of both KEA and TextRank performed better than the original ones.

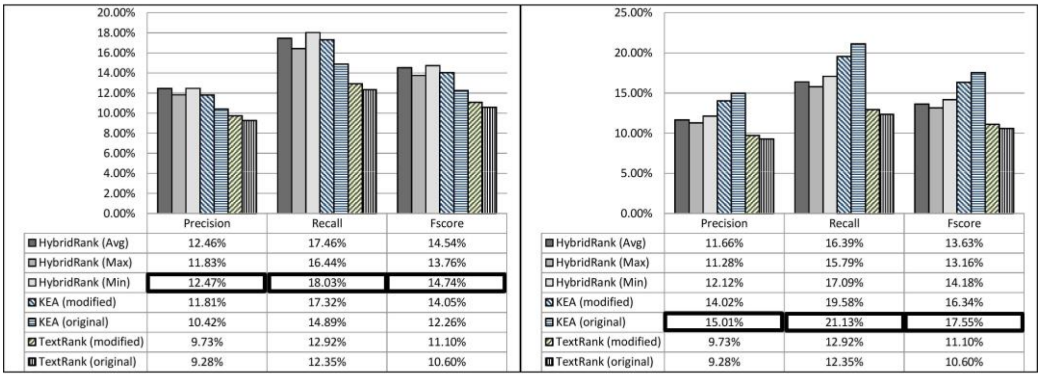
Figure 2 displays the results for the IEEE Xplore dataset. In this dataset, when applying the Base Feature Set (New) on KEA and using the Min merging method, HybridRank performed better than the other methods. However, when applying the PoS Tag Feature Set, the original KEA method outperformed the others. This is probably due to the fact that the IEEE Xplore dataset has a greater variety of subjects than the Hulth 2003 dataset. This wide range of subjects causes the Keyphrase Frequency attribute – applied on the Base Feature Set (New) – to become less meaningful (Frank *et al.*, 1999), thus allowing the PoS Tag Feature Set to predict a phrase's class (keyphrase or non-keyphrase) with higher accuracy. Overall, our method performed better than either KEA or TextRank in all of the cases.



	Precision	Recall	Fscore
HybridRank (Avg)	37.14%	21.01%	26.84%
HybridRank (Max)	38.20%	21.21%	27.27%
HybridRank (Min)	34.28%	19.07%	24.51%
KEA (modified)	16.09%	9.11%	11.63%
KEA (original)	12.90%	6.77%	8.88%
TextRank (modified)	35.44%	19.61%	25.25%
TextRank (original)	33.46%	17.00%	22.55%

	Precision	Recall	Fscore
HybridRank (Avg)	38.76%	21.53%	27.68%
HybridRank (Max)	40.20%	22.19%	28.60%
HybridRank (Min)	36.57%	20.54%	26.30%
KEA (modified)	25.04%	14.74%	18.56%
KEA (original)	20.37%	11.94%	15.05%
TextRank (modified)	35.44%	19.61%	25.25%
TextRank (original)	33.46%	17.00%	22.55%

Figure 1. Precision, recall and F-score on the Hulth 2003 dataset. The left corresponds to the Base Feature Set (New), the right to the PoS Tag Feature Set.



	Precision	Recall	Fscore
HybridRank (Avg)	12.46%	17.46%	14.54%
HybridRank (Max)	11.83%	16.44%	13.76%
HybridRank (Min)	12.47%	18.03%	14.74%
KEA (modified)	11.81%	17.32%	14.05%
KEA (original)	10.42%	14.89%	12.26%
TextRank (modified)	9.73%	12.92%	11.10%
TextRank (original)	9.28%	12.35%	10.60%

	Precision	Recall	Fscore
HybridRank (Avg)	11.66%	16.39%	13.63%
HybridRank (Max)	11.28%	15.79%	13.16%
HybridRank (Min)	12.12%	17.09%	14.18%
KEA (modified)	14.02%	19.58%	16.34%
KEA (original)	15.01%	21.13%	17.55%
TextRank (modified)	9.73%	12.92%	11.10%
TextRank (original)	9.28%	12.35%	10.60%

Figure 2. Precision, recall and F-score on the IEEE Xplore dataset. The left corresponds to the Base Feature Set (New), the right to the PoS Tag Feature Set.

6. Conclusions and Future Work

In this paper, we have described and evaluated a hybrid keyphrase extraction method: HybridRank. Our results show that collaboration between a supervised and an unsupervised approach can produce high-quality keyphrase lists for short articles. We have compared the performance of HybridRank with two other well-known keyphrase extraction methods – KEA and TextRank – and showed that HybridRank obtained a higher precision, recall and F-score when applied on the Hulth 2003 dataset.

On our second dataset (IEEE Xplore), the original KEA algorithm performed better than HybridRank and TextRank when using PoS Tag Patterns because this dataset contains a wide range of domains, affecting the performance of the Naïve Bayes classifier when using the Base Feature Set (New). Our method, however, outperformed in all cases either the supervised (KEA) or unsupervised (TextRank) approaches. Furthermore, doing some modifications to KEA and TextRank improved their performance in most cases as compared to the original methods proposed by their authors.

We can conclude that HybridRank performs the best when the unsupervised component outperforms the supervised component. Additionally, merging KEA's and TextRank's keyphrases with the Min or Max methods produced better results than using the Average.

Among our planned future work is adopting a weighting mechanism to both components, so as to have biased merging, either towards the supervised component or towards the unsupervised one. Another approach we have considered is to implement different (and newer) methods for the supervised and unsupervised components (see Section 2), so as to maximize the overall performance of the HybridRank system.

References

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107--117, 1998.

Eibe Frank and Gordon W. Paynter and Ian H. Witten. Domain-specific keyphrase extraction. *IJCAI*, 1999.

Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, :216--223, 2003.

R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. *Proceedings of EMNLP*, :404--411, 2004.

T.D. Nguyen and M.Y. Kan. Keyphrase extraction in scientific publications. *Proceedings of ICADL2007*, 2007.

Peter D. Turney. Coherent Keyphrase Extraction via Web Mining. *Proceedings of the Eighteenth Research Council*, 1999.

Peter D. Turney. Learning Algorithms for Keyphrase Extraction. *Inf. Retr.*, 2(4):303--336, 2000.

X. Wan and J. Xiao. CollabRank: towards a collaborative approach to single-document keyphrase extraction. *Proceedings of the 22nd International Conference on Computational Linguistics*, 1:969--976, 2008.

X. Wan and J. Yang and J. Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. *Annual Meeting-Association for Computational Linguistics*, 45(1):552, 2007.

Ian H. Witten and Gordon W. Paynter and Eibe Frank and Carl Gutwin and Craig G. Nevill-Manning. KEA: practical automatic keyphrase extraction. *DL '99: Proceedings of the fourth ACM conference on Digital libraries*, :254--255, 1999.

J. D. Cohen. Highlights: Language- and domain-independent automatic indexing terms for abstracting. *JASIS*, 46(3):162--174, 1995.

T. Jo. Neural based approach to keyword extraction from documents. In *Computational Science and Its Applications--ICCSA 2003*, pages 456--461. Springer, 2003.

T. Jo, M. Lee, and T. M. Gatton. Keyword extraction from documents using a neural network model. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, volume 2, pages 194--197. IEEE, 2006.

H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309--317, 1957.

Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157--169, 2004.

G. Salton, C.-S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, 26(1):33--44, 1975.

K. Sarkar, M. Nasipuri, and S. Ghose. A new approach to keyphrase extraction using neural networks. *arXiv preprint arXiv:1004.3274*, 2010.

W. tau Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213--222. ACM, 2006.

J. Wang, H. Peng, and J. song Hu. Automatic keyphrases extraction from document using neural network. *In Advances in Machine Learning and Cybernetics*, pages 633--641. Springer, 2006.

C. Zhang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 2008.