

# Re-examining Automatic Keyphrase Extraction Approaches in Scientific Articles

**Su Nam Kim**

CSSE dept.  
University of Melbourne  
snkim@csse.unimelb.edu.au

**Min-Yen Kan**

Computer Science Dept.  
National University of Singapore  
kanmy@comp.nus.edu.sg

## Abstract

We tackle two major issues in automatic keyphrase extraction using scientific articles: *candidate selection* and *feature engineering*. First of all, to develop an efficient candidate selection method, we analyze the nature and variation of keyphrases and then select candidates using regular expressions. Secondly, we re-examine the existing features broadly used for the supervised approach, exploring different ways to enhance their performance. While most other approaches are supervised, we also study the optimal features for unsupervised keyphrase extraction. Our research has shown that effective candidate selection leads to better performance as evaluation accounts for candidate coverage. Our work also attests that many of existing features are also usable in unsupervised extraction.

## 1 Introduction

*Keyphrases* are simplex nouns or noun phrases (NPs) that represent the key ideas of the document. Keyphrases can serve as a representative summary of the document and also serve as high quality index terms. It is thus no surprise that keyphrases have been utilized to acquire critical information as well as to improve the quality of natural language processing (NLP) applications such as document summarizer (D'Avanzo and Magnini, 2005), information retrieval (IR) (Gutwin et al., 1999) and document clustering (Hammouda et al., 2005).

In the past, various attempts have been made to boost automatic keyphrase extraction performance based primarily on statistics (Frank et al., 1999; Turney, 2003; Park et al., 2004; Wan and Xiao, 2008) and a rich set of heuristic features (Barker and Cornacchia, 2000; Medelyan and Witten,

2006; Nguyen and Kan, 2007). In Section 2, we give a more comprehensive overview of previous attempts.

Current keyphrase technology still has much room for improvement. First of all, although several *candidate selection* methods have been proposed for automatic keyphrase extraction in the past (e.g. (Frank et al., 1999; Park et al., 2004; Nguyen and Kan, 2007)), most of them do not effectively deal with various keyphrase forms which results in the ignorance of some keyphrases as candidates. Moreover, no studies thus far have done a detailed investigation of the nature and variation of manually-provided keyphrases. As a consequence, the community lacks a standardized list of candidate forms, which leads to difficulties in direct comparison across techniques during evaluation and hinders re-usability.

Secondly, previous studies have shown the effectiveness of their own features but not many compared their features with other existing features. That leads to a redundancy on study and even becomes a barrier for further improvement. In addition, existing features are specifically designed for supervised approaches with few exceptions. However, this approach involves a large amount of manual task thus it reduces the merit to use for real-world applications. Hence, unsupervised approach is inevitable in order to minimize manual tasks and to encourage utilization. It is a worthy study to attest the reliability and re-usability for the unsupervised approach in order to set up the tentative guideline for applications.

This paper targets to resolve these issues of *candidate selection* and *feature engineering*. In our work on candidate selection, we analyze the nature and variation of keyphrases with the purpose of proposing a candidate selection method which improves the coverage of candidates that occur in various forms. Our second contribution re-examines existing keyphrase extraction features

reported in the literature, in terms of their effectiveness and re-usability. We test and compare the usefulness of each feature for further improvement. In addition, we assess how well these features can be applied in an unsupervised approach.

In the remaining sections, we describe an overview of related work in Section 2, our proposals on candidate selection and feature engineering in Section 4 and 5, our system architecture and data in Section 6. Then, we evaluate our proposals, discuss outcomes and conclude our work in Section 7, 8 and 9, respectively.

## 2 Related Work

The majority of related work has been carried out using statistical approaches, a rich set of symbolic resources and linguistically-motivated heuristics (Frank et al., 1999; Turney, 1999; Barker and Cornacchia, 2000; Matsuo and Ishizuka, 2004; Nguyen and Kan, 2007). Features used can be categorized into three broad groups: (1) document cohesion features (i.e. relationship between document and keyphrases) (Frank et al., 1999; Matsuo and Ishizuka, 2004; Medelyan and Witten, 2006; Nguyen and Kan, 2007), and to lesser, (2) keyphrase cohesion features (i.e. relationship among keyphrases) (Turney, 2003) and (3) term cohesion features (i.e. relationship among components in a keyphrase) (Park et al., 2004).

The simplest system is KEA (Frank et al., 1999; Witten et al., 1999) that uses  $TF*IDF$  (i.e. term frequency \* inverse document frequency) and *distance* (i.e. *first occurrence* in the document).  $TF*IDF$  measures the document cohesion and *distance* implies the importance of the abstract or introduction which indicates the keyphrases have a locality. Turney (2003) added the notion of keyphrase cohesion to KEA features and Nguyen and Kan (2007) added linguistic features such as section information and suffix sequence. The GenEx system (Turney, 1999) employed an inventory of nine syntactic features, such as length in words and frequency of stemming phrase as a set of parametrized heuristic rules. Barker and Cornacchia (2000) introduced a method based on head noun heuristics that took three features: length of candidate, frequency and head noun frequency. To take advantage of domain knowledge, Hulth et al. (2001) used a hierarchically-organized domain-specific thesaurus from Swedish Parliament as a secondary knowledge source. The

Text<sub>extract</sub> (Park et al., 2004) also ranks the candidate keyphrases by its judgment of keyphrases' degree of domain specificity based on subject-specific collocations (Damerau, 1993), in addition to term cohesion using Dice coefficient (Dice, 1945). Recently, Wan and Xiao (2008) extracts automatic keyphrases from single documents, utilizing document clustering information. The assumption behind this work is that the documents with the same or similar topics interact with each other in terms of salience of words. The authors first clustered the documents then used the graph-based ranking algorithm to rank the candidates in a document by making use of mutual influences of other documents in the same cluster.

## 3 Keyphrase Analysis

In previous study, KEA employed the indexing words as candidates whereas others such as (Park et al., 2004; Nguyen and Kan, 2007) generated handcrafted regular expression rules. However, none carefully undertook the analysis of keyphrases. We believe there is more to be learned from the reference keyphrases themselves by doing a fine-grained, careful analysis of their form and composition. Note that we used the articles collected from ACM digital library for both analyzing keyphrases as well as evaluating methods. See Section 6 for data in detail.

Syntactically, keyphrases can be formed by either simplex nouns (e.g. *algorithm*, *keyphrase*, *multi-agent*) or noun phrases (NPs) which can be a sequence of nouns and their auxiliary words such as adjectives and adverbs (e.g. *mobile network*, *fast computing*, *partially observable Markov decision process*) despite few incidences. They can also incorporate a prepositional phrase (PP) (e.g. *quality of service*, *policy of distributed caching*). When keyphrases take the form of an NP with an attached PP (i.e. NPs in *of-PP form*), the preposition *of* is most common, but others such as *for*, *in*, *via* also occur (e.g. *incentive for cooperation*, *inequality in welfare*, *agent security via approximate policy*, *trade in financial instrument based on logical formula*). The patterns above correlate well to part-of-speech (POS) patterns used in modern keyphrase extraction systems.

However, our analysis uncovered additional linguistic patterns and alternations which other studies may have overlooked. In our study we also found that keyphrases also occur as a simple con-

Criteria	Rules
Frequency	<b>(Rule1) Frequency heuristic</b> i.e. frequency $\geq 2$ for simplex words vs. frequency $\geq 1$ for NPs
Length	<b>(Rule2) Length heuristic</b> i.e. up to length 3 for NPs in non- <i>of-PP form</i> vs. up to length 4 for NPs in <i>of-PP form</i> (e.g. <i>synchronous concurrent program</i> vs. <i>model of multiagent interaction</i> )
Alternation	<b>(Rule3) of-PP form alternation</b> (e.g. <i>number of sensor = sensor number</i> , <i>history of past encounter = past encounter history</i> ) <b>(Rule4) Possessive alternation</b> (e.g. <i>agent's goal = goal of agent</i> , <i>security's value = value of security</i> )
Extraction	<b>(Rule5) Noun Phrase</b> = $(NN NNS NNP NNPS JJ JJR JJS)^*(NN NNS NNP NNPS)$ (e.g. <i>complexity</i> , <i>effective algorithm</i> , <i>grid computing</i> , <i>distributed web-service discovery architecture</i> ) <b>(Rule6) Simplex Word/NP IN Simplex Word/NP</b> (e.g. <i>quality of service</i> , <i>sensitivity of VOIP traffic</i> ( <b>VOIP traffic</b> extracted), <i>simplified instantiation of zebroid</i> ( <b>simplified instantiation</b> extracted))

Table 1: Candidate Selection Rules

junctions (e.g. *search and rescue*, *propagation and delivery*), and much more rarely, as conjunctions of more complex NPs (e.g. *history of past encounter and transitivity*). Some keyphrases appear to be more complex (e.g. *pervasive document edit and management system*, *task and resource allocation in agent system*). Similarly, abbreviations and possessive forms figure as common patterns (e.g. *belief desire intention = BDI*, *inverse document frequency = (IDF)*; *Bayes' theorem*, *agent's dominant strategy*).

A critical insight of our work is that keyphrases can be morphologically and semantically altered. Keyphrases that incorporate a PP or have an underlying genitive composition are often easily varied by word order alternation. Previous studies have used the altered keyphrases when forming in *of-PP form*. For example, *quality of service* can be altered to *service quality*, sometimes with little semantic difference. Also, as most morphological variation in English relates to noun number and verb inflection, keyphrases are subject to these rules as well (e.g. *distributed system*  $\neq$  *distributing system*, *dynamical caching*  $\neq$  *dynamical cache*). In addition, possessives tend to alternate with *of-PP form* (e.g. *agent's goal = goal of agent*, *security's value = value of security*).

#### 4 Candidate Selection

We now describe our proposed *candidate selection* process. Candidate selection is a crucial step for automatic keyphrase extraction. This step is correlated to term extraction study since top  $N_{th}$  ranked terms become keyphrases in documents. In previous study, KEA employed the indexing words as candidates whereas others such as (Park et al., 2004; Nguyen and Kan, 2007) generated hand-crafted regular expression rules. However, none carefully undertook the analysis of keyphrases. In

this section, before we present our method, we first describe the detail of keyphrase analysis.

In our keyphrase analysis, we observed that most of *author assigned keyphrase* and/or *reader assigned keyphrase* are syntactically more often simplex words and less often NPs. When keyphrases take an NP form, they tend to be simple form of NPs. i.e. either without a PP or with only a PP or with a conjunction but not much to appear in mixture of those forms. We also noticed that the components of NPs are normally nouns and adjectives but rarely, are adverbs and verbs. As a result, we decided to ignore NPs containing adverbs and verbs in this study as our candidates since they tend to produce more errors and to require more complexity.

Another observation is that keyphrases containing more than three words are rare (i.e. 6% in our data set), validating what Paukkeri et al. (2008) observed. Hence, we apply a *length heuristic*. Our candidate selection rule collects candidates up to length 3, but also of length 4 for NPs in *of-PP form*, since they may have a non-genitive alternation that reduces its length to 3 (e.g. *performance of distributed system = distributed system performance*). In previous studies, words occurring at least twice are selected as candidates. However, during our acquisition of *reader assigned keyphrase*, we observed that readers tend to collect NPs as keyphrases, regardless of their frequency. Due to this, we apply different frequency thresholds for simplex words ( $\geq 2$ ) and NPs ( $\geq 1$ ). Note that 30% of NPs occurred only once in our data.

Finally, we generated regular expression rules to extract candidates, as presented in Table 1. Our candidate extraction rules are based on those in Nguyen and Kan (2007). However, our **Rule6** for NPs in *of-PP form* broadens the coverage of

possible candidates. i.e. with a given NPs in *of-PP form*, not only we collect simplex word(s), but we also extract non-*of-PP form* of NPs from noun phrases governing the PP and the PP. For example, our rule extracts *effective algorithm of grid computing* as well as *effective algorithm* and *grid computing* as candidates while the previous works' rules do not.

## 5 Feature Engineering

With a wider candidate selection criteria, the onus of filtering out irrelevant candidates becomes the responsibility of careful feature engineering. We list 25 features that we have found useful in extracting keyphrases, comprising of 9 existing and 16 novel and/or modified features that we introduce in our work (marked with \*). As one of our goals in *feature engineering* is to assess the suitability of features in the unsupervised setting, we have also indicated which features are suitable only for the supervised setting (**S**) or applicable to both (**S, U**).

### 5.1 Document Cohesion

*Document cohesion* indicates how important the candidates are for the given document. The most popular feature for this cohesion is *TF\*IDF* but some works have also used context words to check the correlation between candidates and the given document. Other features for document cohesion are *distance*, *section information* and so on. We note that listed features other than *TF\*IDF* are related to locality. That is, the intuition behind these features is that keyphrases tend to appear in specific area such as the beginning and the end of documents.

**F1 : TF\*IDF** (*S,U*) *TF\*IDF* indicates document cohesion by looking at the frequency of terms in the documents and is broadly used in previous work (Frank et al., 1999; Witten et al., 1999; Nguyen and Kan, 2007). However, a disadvantage of the feature is in requiring a large corpus to compute useful *IDF*. As an alternative, context words (Matsuo and Ishizuka, 2004) can also be used to measure document cohesion. From our study of keyphrases, we saw that substrings within longer candidates need to be properly counted, and as such our method measures *TF* in substrings as well as in exact matches. For example, *grid computing* is often a substring of other phrases such as *grid computing algorithm* and *efficient grid com-*

*puting algorithm*. We also normalize *TF* with respect to candidate types: i.e. we separately treat simplex words and NPs to compute *TF*. To make our *IDFs* broadly representative, we employed the Google n-gram counts, that were computed over terabytes of data. Given this large, generic source of word count, *IDF* can be incorporated without corpus-dependent processing, hence such features are useful in unsupervised approaches as well. The following list shows variations of *TF\*IDF*, employed as features in our system.

spacing

F1 (a) *TF\*IDF*

F1 \* (b) *TF* including counts of substrings

F1 \* (c) *TF* of substring as a separate feature

F1 \* (d) normalized *TF* by candidate types (i.e. simplex words vs. NPs)

F1 \* (e) normalized *TF* by candidate types as a separate feature

F1 \* (f) *IDF* using Google n-gram

**F2 : First Occurrence (Distance)** (*S,U*) KEA used the first appearance of the word in the document (Frank et al., 1999; Witten et al., 1999). The main idea behind this feature is that keyphrases tend to occur in the beginning of documents, especially in structured reports (e.g., in abstract and introduction sections) and newswire.

**F3 : Section Information** (*S,U*) Nguyen and Kan (2007) used the identity of which specific document section a candidate occurs in. This locality feature attempts to identify key sections. For example, in their study of scientific papers, the authors weighted candidates differently depending on whether they occurred in the abstract, introduction, conclusion, section head, title and/or references.

**F4\* : Additional Section Information** (*S,U*) We first added the *related work or previous work* as one of section information not included in Nguyen and Kan (2007). We also propose and test a number of variations. We used the substrings that occur in section headers and reference titles as keyphrases. We counted the co-occurrence of candidates (i.e. the section *TF*) across all key sections that indicates the correlation among key sections. We assign section-specific weights as individual sections exhibit different propensities for

generating keyphrases. For example, *introduction* contains the majority of keyphrases while the title or section head contains many fewer due to the variation in size.

spacing

F4 \* (a) section, 'related/previous work'

F4 \* (b) counting substring occurring in key sections

F4 \* (c) section *TF* across all key sections

F4 \* (d) weighting key sections according to the portion of keyphrases found

**F5\* : Last Occurrence** (*S,U*) Similar to *distance* in KEA, the position of the last occurrence of a candidate may also imply the importance of keyphrases, as keyphrases tend to appear in the last part of document such as the conclusion and discussion.

## 5.2 Keyphrase Cohesion

The intuition behind using *keyphrase cohesion* is that actual keyphrases are often associated with each other, since they are semantically related to topic of the document. Note that this assumption holds only when the document describes a single, coherent topic – a document that represents a collection may be first need to be segmented into its constituent topics.

**F6\* : Co-occurrence of Another Candidate in Section** (*S,U*) When candidates co-occur in several key sections together, then they are more likely keyphrases. Hence, we used the number of sections that candidates co-occur.

**F7\* : Title overlap** (*S*) In a way, titles also represent the topics of their documents. A large collection of titles in the domain can act as a probabilistic prior of what words could stand as constituent words in keyphrases. In our work, as we examined scientific papers from computer science, we used a collection of titles obtained from the large *CiteSeer*<sup>1</sup> collection to create this feature.

spacing

F7 \* (a) co-occurrence (Boolean) in title collocation

F7 \* (b) co-occurrence (*TF*) in title collection

**F8 : Keyphrase Cohesion** (*S,U*) Turney (2003) integrated keyphrase cohesion into his system by checking the semantic similarity between top *N* ranked candidates against the remainder. In the original work, a large, external web corpus was used to obtain the similarity judgments. As we did not have access to the same web corpus and all candidates/keyphrases were not found in Google *n*-gram corpus, we approximated this feature using a similar notion of contextual similarity. We simulated a latent 2-dimensional matrix (similar to latent semantic analysis) by listing all candidate words in rows and their neighboring words (nouns, verbs, and adjectives only) in columns. The cosine measure is then used to compute the similarity among keyphrases.

## 5.3 Term Cohesion

*Term cohesion* further refines the candidacy judgment, by incorporating an internal analysis of the candidate's constituent words. Term cohesion posits that high values for internal word association measures correlates indicates that the candidate is a keyphrase (Church and Hanks, 1989).

**F9 : Term Cohesion** (*S,U*) Park et al. (2004) used the *Dice coefficient* (Dice, 1945) to measure *term cohesion* particularly for multiword terms. In their work, as NPs are longer than simplex words, they simply discounted simplex word cohesion by 10%. In our work, we vary the measure of *TF* used in *Dice coefficient*, similar to our discussion earlier.

spacing

F9 (a) term cohesion by (Park et al., 2004),

F9 \* (b) normalized *TF* by candidate types (i.e. simplex words vs. NPs),

F9 \* (c) applying different weight by candidate types,

F9 \* (d) normalized *TF* and different weighting by candidate types

## 5.4 Other Features

**F10 : Acronym** (*S*) Nguyen and Kan (2007) accounted for the importance of *acronym* as a feature. We found that this feature is heavily dependent on the data set. Hence, we used it only for N&K to attest our *candidate selection method*.

**F11 : POS sequence** (*S*) Hulth and Megyesi (2006) pointed out that POS sequences of

<sup>1</sup>It contains 1.3M titles from articles, papers and reports.

keyphrases are similar. It showed the distinctive distribution of POS sequences of keyphrases and use them as a feature. Like *acronym*, this is also subject to the data set.

**F12 : Suffix sequence (*S*)** Similar to *acronym*, Nguyen and Kan (2007) also used a candidate’s *suffix sequence* as a feature, to capture the propensity of English to use certain Latin derivational morphology for technical keyphrases. This feature is also a data dependent features thus used in supervised approach only.

**F13 : Length of Keyphrases (*S,U*)** Barker and Cornacchia (2000) showed that candidate length is also a useful feature in extraction as well as in candidate selection, as the majority of keyphrases are one or two terms in length.

## 6 System and Data

To assess the performance of the proposed candidate selection rules and features, we implemented a keyphrase extraction pipe line. We start with raw text of computer science articles converted from *PDF* by *pdftotext*. Then, we partitioned the into section such as title and sections via heuristic rules and applied sentence segmenter<sup>2</sup>, *ParsCit*<sup>3</sup>(Councill et al., 2008) for reference collection, part-of-speech tagger<sup>4</sup> and lemmatizer<sup>5</sup>(Minnen et al., 2001) of the input. After preprocessing, we built both supervised and unsupervised classifiers using Naive Bayes from the WEKA machine learning toolkit(Witten and Frank, 2005), Maximum Entropy<sup>6</sup>, and simple weighting.

In evaluation, we collected 250 papers from four different categories<sup>7</sup> of the ACM digital library. Each paper was 6 to 8 pages on average. In *author assigned keyphrase*, we found many were missing or found as substrings. To remedy this, we collected *reader assigned keyphrase* by hiring senior year undergraduates in computer science, each whom annotated five of the papers with an annotation guideline and on average, took about 15 minutes to annotate each paper. The fi-

<sup>2</sup><http://www.eng.ritsumei.ac.jp/asao/resources/sentseg/>

<sup>3</sup><http://wing.comp.nus.edu.sg/parsCit/>

<sup>4</sup><http://search.cpan.org/dist/Lingua-EN-Tagger/Tagger.pm>

<sup>5</sup><http://www.informatics.susx.ac.uk/research/groups/nlp/carroll/morph.html>

<sup>6</sup><http://maxent.sourceforge.net/index.html>

<sup>7</sup>C2.4 (Distributed Systems), H3.3 (Information Search and Retrieval), I2.11 (Distributed Artificial Intelligence-Multiagent Systems) and J4 (Social and Behavioral Sciences-Economics)

nal statistics of keyphrases is presented in Table 2 where *Combined* represents the total number of keyphrases. The numbers in ( ) denotes the number of keyphrases in *of-PP form*. *Found* means the number of *author assigned keyphrase* and *reader assigned keyphrase* found in the documents.

	Author	Reader	Combined
Total	1252 (53)	3110 (111)	3816 (146)
NPs	904	2537	3027
Average	3.85 (4.01)	12.44 (12.88)	15.26 (15.85)
Found	769	2509	2864

Table 2: Statistics in Keyphrases

## 7 Evaluation

The baseline system for both the supervised and unsupervised approaches is *modified N&K* which uses *TF\*IDF*, *distance*, *section information* and *additional section information* (i.e. *F1-4*). Apart from *baseline*, we also implemented basic KEA and N&K to compare. Note that N&K is considered a supervised approach, as it utilizes features like *acronym*, *POS sequence*, and *suffix sequence*.

Table 3 and 4 shows the performance of our candidate selection method and features with respect to supervised and unsupervised approaches using the current standard evaluation method (i.e. exact matching scheme) over top  $5_{th}$ ,  $10_{th}$ ,  $15_{th}$  candidates.

**BestFeatures** includes *F1c:TF of substring as a separate feature*, *F2:first occurrence*, *F3:section information*, *F4d:weighting key sections*, *F5:last occurrence*, *F6:co-occurrence of another candidate in section*, *F7b:title overlap*, *F9a:term cohesion by (Park et al., 2004)*, *F13:length of keyphrases*. **Best-TF\*IDF** means using all best features but *TF\*IDF*.

In Tables 3 and 4, *C* denotes the classifier technique: unsupervised (*U*) or supervised using Maximum Entropy (*S*)<sup>8</sup>.

A	Method	Feature
+	S	F1a,F2,F3,F4a,F4d,F9a
	U	F1a,F1c,F2,F3,F4a,F4d,F5,F7b,F9a
-	S	F1b,F1c,F1d,F1f,F4b,F4c,F7a,F7b,F9b-d,F13
	U	F1d,F1e,F1f,F4b,F4c,F6,F7a,F9b-d
?	S	F1e,F10,F11,F12
	U	F1b

Table 5: Performance on Each Feature

In Table 5, the performance of each feature is measured using N&K system and the target fea-

<sup>8</sup>Due to the page limits, we present the best performance.

Method	Features	C	Five				Ten				Fifteen			
			Match	Precision	Recall	Fscore	Match	Precision	Recall	Fscore	Match	Precision	Recall	Fscore
All Candidates	KEA	U	0.03	0.64%	0.21%	0.32%	0.09	0.92%	0.60%	0.73%	0.13	0.88%	0.86%	0.87%
		S	0.79	15.84%	5.19%	7.82%	1.39	13.88%	9.09%	10.99%	1.84	12.24%	12.03%	12.13%
	N&K baseline	S	1.32	26.48%	8.67%	13.06%	2.04	20.36%	13.34%	16.12%	2.54	16.93%	16.64%	16.78%
		U	0.92	18.32%	6.00%	9.04%	1.57	15.68%	10.27%	12.41%	2.20	14.64%	14.39%	14.51%
		S	1.15	23.04%	7.55%	11.37%	1.90	18.96%	12.42%	15.01%	2.44	16.24%	15.96%	16.10%
Length<=3 Candidates	KEA	U	0.03	0.64%	0.21%	0.32%	0.09	0.92%	0.60%	0.73%	0.13	0.88%	0.86%	0.87%
		S	0.81	16.16%	5.29%	7.97%	1.40	14.00%	9.17%	11.08%	1.84	12.24%	12.03%	12.13%
	N&K baseline	S	1.40	27.92%	9.15%	13.78%	2.10	21.04%	13.78%	16.65%	2.62	17.49%	17.19%	17.34%
		U	0.92	18.4%	6.03%	9.08%	1.58	15.76%	10.32%	12.47%	2.20	14.64%	14.39%	14.51%
		S	1.18	23.68%	7.76%	11.69%	1.90	19.00%	12.45%	15.04%	2.40	16.00%	15.72%	15.86%
Length<=3 Candidates + Alternation	KEA	U	0.01	0.24%	0.08%	0.12%	0.05	0.52%	0.34%	0.41%	0.07	0.48%	0.47%	0.47%
		S	0.83	16.64%	5.45%	8.21%	1.42	14.24%	9.33%	11.27%	1.87	12.45%	12.24%	12.34%
	N&K baseline	S	1.53	30.64%	10.04%	15.12%	2.31	23.08%	15.12%	18.27%	2.88	19.20%	18.87%	19.03%
		U	0.98	19.68%	6.45%	9.72%	1.72	17.24%	11.29%	13.64%	2.37	15.79%	15.51%	15.65%
		S	1.33	26.56%	8.70%	13.11%	2.09	20.88%	13.68%	16.53%	2.69	17.92%	17.61%	17.76%

Table 3: Performance on Proposed Candidate Selection

Features	C	Five				Ten				Fifteen			
		Match	Prec.	Recall	Fscore	Match	Prec.	Recall	Fscore	Match	Prec.	Recall	Fscore
Best	U	1.14	.228	.747	.113	1.92	.192	.126	.152	<b>2.61</b>	.174	.171	.173
	S	1.56	.312	.102	.154	2.50	.250	.164	.198	<b>3.15</b>	.210	.206	.208
Best w/o TF*IDF	U	1.14	.228	.74	.113	1.92	.192	.126	.152	<b>2.61</b>	.174	.171	.173
	S	1.56	.311	.102	.154	2.46	.246	.161	.194	<b>3.12</b>	.208	.204	.206

Table 4: Performance on Feature Engineering

ture. + indicates an improvement, - indicates a performance decline, and ? indicates no effect or not confirmed due to small changes of performances. Again, *supervised* denotes Maximum Entropy training and *Unsupervised* is our unsupervised approach.

## 8 Discussion

We compared the performances over our candidate selection and feature engineering with simple KEA, N&K and our baseline system. In evaluating candidate selection, we found that longer length candidates play a role to be noises so decreased the overall performance. We also confirmed that candidate alternation offered the flexibility of keyphrases leading higher candidate coverage as well as better performance.

To re-exam features, we analyzed the impact of existing and new features and their variations. First of all, unlike previous study, we found that the performance with and without *TF\*IDF* was not big different which indicates the impact of *TF\*IDF* was minor as long as other features are incorporated. Secondly, counting substrings for *TF* improved performance, while applying term weighting for *TF* and/or *IDF* did not impact on the performance. We estimated the cause that many of keyphrases are substrings of candidates and vice versa. Thirdly, *section information* was also validated to improve performance, as in Nguyen and Kan (2007). Extending this logic, modeling additional section information (*related work*) and *weighting sections* both turned out to be useful features. Other locality features were also

validated as helpful: both *first occurrence* and *last occurrence* are helpful as it implies the locality of the key ideas. In addition, keyphrase co-occurrence with selected sections was proposed in our work and found empirically useful. Term cohesion (Park et al., 2004) is a useful feature although it has a heuristic factor that reduce the weight by 10% for simplex words. Normally, term cohesion is subject to NPs only, hence it needs to be extended to work with multi-word NPs as well. Table 5 summarizes the reflections on each feature.

As unsupervised methods have the appeal of not needing to be trained on expensive hand-annotated data, we also compared the performance of supervised and unsupervised methods. Given the features initially introduced for supervised learning, unsupervised performance is surprisingly high. While supervised classifier produced a matching count of **3.15**, the unsupervised classifier obtains a count of **2.61**. We feel this indicates that the existing features for supervised methods are also suitable for use in unsupervised methods, with slightly reduced performance. In general, we observed that the best features in both supervised and unsupervised methods are the same – *section information* and *candidate length*. In our analysis of the impact of individual features, we observed that most features affect performance in the same way for both supervised and unsupervised approaches, as shown in Table 5. These findings indicate that although these features may be originally designed for use in a supervised approach, they are stable and can be expected to perform similar in

unsupervised approaches.

## 9 Conclusion

We have identified and tackled two core issues in automatic keyphrase extraction: *candidate selection* and *feature engineering*. In the area of *candidate selection*, we observe variations and alternations that were previously unaccounted for. Our selection rules expand the scope of possible keyphrase coverage, while not overly expanding the total number candidates to consider. In our re-examination of *feature engineering*, we compiled a comprehensive feature list from previous works while exploring the use of substrings in devising new features. Moreover, we also attested to each feature's fitness for use in unsupervised approaches, in order to utilize them in real-world applications with minimal cost.

## References

- K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*. 2000.
- R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*. 1997, pp. 10–17.
- K.W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *Proceedings of the 27th Annual Meeting of the Association of Computational Linguistics*. 1989, 76–83.
- I.G. Councill and C.L. Giles and M.Y. Kan. ParsCit: An open-source CRF reference string parsing package. In *Proceedings of the Language Resources and Evaluation Conference*. 2008, 28–30.
- E. D'Avanzo and B. Magnini. A Keyphrase-Based Approach to Summarization: the LAKE System at DUC-2005. In *Proceedings of 5th DUC*. 2005.
- F. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*. 1993, 29, pp.43–447.
- L. Dice. Measures of the amount of ecologic associations between species. *Journal of Ecology*. 1945, 2.
- E. Frank and G.W. Paynter and I. Witten and C. Gutwin and C.G. Nevill-Manning. Domain Specific Keyphrase Extraction. In *Proceedings of the 16th IJCAI*. 1999, pp.668–673.
- C. Gutwin and G.W. Paynter and I.H. Witten and C.G. Nevillmanning and E. Frank. Improving browsing in digital libraries with keyphrase indexes. *Journal of Decision Support Systems*. 1999, 27, pp.81–104.
- K.M. Hammouda and D.N. Matute and M.S. Kamel. CorePhrase: keyphrase extraction for document clustering. In *Proceedings of MLDM*. 2005.
- A. Hulth and J. Karlgren and A. Jonsson and H. Boström and L. Asker. Automatic Keyword Extraction using Domain Knowledge. In *Proceedings of the 2nd CICLing*. 2001.
- A. Hulth and B.B. Megyesi. A study on automatically extracted keywords in text categorization. In *Proceedings of ACL/COLING*. 2006, 537–544.
- M. Jarmasz and C. Barriere. Using semantic similarity over tera-byte corpus, compute the performance of keyphrase extraction. In *Proceedings of CLINE*. 2004.
- D. Lawrie and W.B. Croft and A. Rosenberg. Finding Topic Words for Hierarchical Summarization. In *Proceedings of SIGIR*. 2001, pp. 349–357.
- Y. Matsuo and M. Ishizuka. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. In *International Journal on Artificial Intelligence Tools*. 2004, 13(1), pp. 157–169.
- O. Medelyan and I. Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEED-CS joint conference on Digital libraries*. 2006, pp.296–297.
- G. Minnen and J. Carroll and D. Pearce. Applied morphological processing of English. *NLE*. 2001, 7(3), pp.207–223.
- T. Nguyen and M.Y. Kan. Key phrase Extraction in Scientific Publications. In *Proceeding of International Conference on Asian Digital Libraries*. 2007, pp.317-326.
- Y. Park and R.J. Byrd and B. Boguraev. Automatic Glossary Extraction Beyond Terminology Identification. In *Proceedings of COLING*. 2004, pp.48–55.
- M.S. Paukkeri and I.T. Nieminen and M. Polla and T. Honkela. A Language-Independent Approach to Keyphrase Extraction and Evaluation. In *Proceedings of The 22nd International Conference on Computational Linguistics*. 2008.
- P. Turney. Learning to Extract Keyphrases from Text. In *National Research Council, Institute for Information Technology, Technical Report ERB-1057*. 1999.
- P. Turney. Coherent keyphrase extraction via Web mining. In *Proceedings of the 18th IJCAI*. 2003, pp. 434–439.
- X. Wan and J. Xiao. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of COLING*. 2008.
- I. Witten and G. Paynter and E. Frank and C. Gutwin and G. Nevill-Manning. KEA: Practical Automatic Key phrase Extraction. In *Proceedings of the fourth ACM conference on Digital libraries*. 1999, pp.254–256.
- I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.
- Y. Zhang and N.Zinichir-Heywood and E. Milios. Term based Clustering and Summarization of Web Page Collections. In *Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence*. 2004.