Natural Language Engineering **1** (1): 1–24. Printed in the United Kingdom

Keyword Extraction from Emails

S. LAHIRI, R. MIHALCEA
University of Michigan, Ann Arbor, Michigan, USA 48109

and P.-H. LAI
Samsung Research America, Richardson, Texas, USA 75082

(*Received 20 March 2015*)

Abstract

Emails constitute an important genre of online communication. Many of us are often faced with the daunting task of sifting through increasingly large amounts of emails on a daily basis. Keywords extracted from emails can help us combat such information overload by allowing a systematic exploration of the topics contained in emails. Existing literature on keyword extraction has not covered the email genre, and no human-annotated gold standard datasets are currently available. In this paper, we introduce a new dataset for keyword extraction from emails, and evaluate supervised and unsupervised methods for keyword extraction from emails. The results obtained with our supervised keyword extraction system (38.99% F-score) improve over the results obtained with the best performing systems participating in the SemEval 2010 keyword extraction task.

1 Introduction

With 144.8 billion emails sent every day around the world,[1] emails represent an essential mode of digital communication. The market share of emails is tremendous, and largely untapped. Not only are traditional applications of keyword extraction important for emails, but emails present a unique scenario in their own right. According to Wasserman, up to 28% of workers' time is spent checking emails,[2] and most work emails are not important.[3] It is therefore paramount that we be able to somehow sort this enormous pile of emails into a workable collection so that the more important ones are dealt with immediately, whereas others are relegated to future inspection (Laclavík and Maynard 2009).

While all existing email clients include some form of free text search to help users identify relevant threads of conversation, keyword extraction from emails can help us spot salient phrases from emails, thereby automatically tagging/categorizing

[1] http://mashable.com/2012/11/27/email-stats-infographic/
[2] http://mashable.com/2012/08/01/email-workers-time/
[3] http://mashable.com/2012/06/07/most-work-emails-not-important-study/

them into appropriate folders (or “labels”), and in effect, giving us a *faceted search* functionality complementary to the vanilla text search on emails.

In this paper, we address the task of automatic keyword[4] extraction from emails, as a way to automatically annotate email content with salient words or phrases that can help us decide on the importance or relevance of an email or thread. The availability of keywords could facilitate the access to email on mobile devices, and they could be used as a preprocessing step for smart email applications that aim to classify or prioritize emails. Keywords could help visualize emails in the form of “keyword clouds” with larger keywords indicating more salience (Chuang *et al.* 2012), and the clouds can be interactive so that when a user clicks or taps on a certain keyword, all emails pertaining to that keyword get retrieved. Different colors indicate different “keyword topics”, so that all keywords under a certain topic get grouped together in one area of the cloud.

Furthermore, keywords form their own social networks (Grineva *et al.* 2009), and important meta-information about documents can be mined by looking into networks of keywords. Keyword networks also serve as a powerful visualization tool by themselves, so that users who are interested in relationship (or association strength) between two keywords or keyword cliques may benefit from looking at such visualizations.

Keyword extraction is an important problem in natural language processing, where the goal is to identify the most important words and phrases in a document. The keywords can either serve as a short summary of the document, giving users an overview of its contents; or they can indicate the topics that are being discussed.

While it may be argued that keywords are often an impoverished representation of the underlying topic space, and that there are alternative models that capture such spaces (Blei, Ng, and Jordan 2003; Dredze *et al.* 2008; Liu *et al.* 2010), it is important to consider that probabilistic topic models need to be trained on large corpora, and they often suffer from scalability issues. Keyword extraction, on the other hand, can be performed in both supervised and unsupervised fashion, and most keyword extraction methods do not need large training corpora.

Keyword extraction has traditionally been the domain of librarians and book indexers, but more recently the problem has seen a number of novel applications. For instance, keywords have been used to thematically group web sites (Tonella *et al.* 2003), where authors reverse-engineered a graph of webpages by clustering them, and using keywords to label the clusters. Keywords were used as *anchor phrases* to link to Wikipedia articles in (Mihalcea and Csomai 2007), and as summary topics to visualize how topics change over time in online Korean news articles (Lee and Kim 2008). Keywords have been used to target advertisements on webpages (Yih, Goodman, and Carvalho 2006), and as indicators of academic paper content and user interest in a content-based paper recommendation system (Ferrara, Pudota, and Tasso 2011).

With such a large number of applications, it is surprising that keyword extraction

[4] We use the term “keyword” to refer to key words or phrases.

from emails has not received much attention from the research community. In this paper, we introduce a new dataset, consisting of single and thread emails manually annotated with keywords, and describe a number of features that can be used for unsupervised or supervised email keyword extraction. Through several evaluations, we show that we can achieve results that significantly improve over several baselines, and also improve over state-of-the-art systems participating in the SemEval 2010 keyphrase extraction task (Kim *et al.* 2010).

2 Background and Related Work

Keyword extraction usually proceeds in three steps: candidate extraction, ranking/classification, and post-processing. In the candidate extraction step, potentially important phrases are identified and extracted from the documents. In the ranking/classification step, these candidate terms are either ranked according to some ranking function derived from the document structure, or they are classified as to whether they represent key terms or not. In the post-processing step, top k terms from the ranked list (or terms that are classified as keywords) are semantically normalized to yield a set of phrases so that each denotes a single concept.

In practice, there are some good heuristics for candidate extraction. Hulth (2003) noted that base noun phrases often constitute a predominant form of keyphrase. She further leveraged previous studies in observing that specific patterns of part-of-speech tags are beneficial for keyword extraction. Csomai and Mihalcea (2007) experimented with stopword-filtered n-grams and named entities as potential keywords.

The second step – ranking/classification – is trickier, because it is not immediately obvious what ranking function or phrase features to use. In a study by Hasan and Ng (2010), *tfidf* was shown to be a surprisingly robust candidate. While conceptually simple, it beats other more complex ranking strategies. Other important features for keyword classification include *tfidf* (Nguyen and Kan 2007; Jiang, Hu, and Li 2009; Li *et al.* 2010), *first occurrence position* of the phrase (Hulth 2003), *capitalization* (Li *et al.* 2010), *phrase length* (in words) and *is-in-title* (Jiang, Hu, and Li 2009).

Two salient groups of ranking functions deal with the *phraseness* and *informativeness* of a candidate phrase (Tomokiyo and Hurst 2003). Informativeness denotes how much information content a stand-alone phrase carries with it, and is usually determined by the number of occurrences of that phrase in a *background corpus* (Tomokiyo and Hurst 2003). Phraseness, on the other hand, is a measure of how cohesive or *tightly-linked* a phrase is; in other words, whether the constituent words of a phrase come together more often than by chance. Phraseness is estimated based on the co-occurrence frequency of words in a *foreground corpus*. A final ranking of phrases is produced by some linear combination of phraseness and informativeness scores. Tomokiyo and Hurst (2003) proposed the use of *language models* in estimating phraseness and informativeness, whereas Csomai and Mihalcea (2007) used chi-squared test. A different formulation is that of *keyphraseness*, proposed by Mihalcea and Csomai (2007), where the probability of a word or a phrase to be linked to a Wikipedia article, calculated across the entire Wikipedia, is used as an indication of how likely that word or phrase is to be selected as a keyword.

Note that the above-mentioned phrase ranking strategies are mostly *ad hoc*, and they emerged as a way to heuristically assess the purported importance or relevance of a phrase. There is, however, a completely different class of ranking algorithms that look into this problem from a more cognitively appealing standpoint. These algorithms look into the structure of *word co-occurrence networks*, where nodes are word types and edges are word collocations. Mihalcea and Tarau (2004) introduced TextRank and observed that in these networks, important words can be thought of as being *endorsed* by other words, and this leads to an interesting phenomenon. Words that are most important, viz. keywords, emerge as the *most central words* in the resulting network, with high degree and PageRank (Page *et al.* 1998). A stream of studies ensued after the seminal work of TextRank (see Hasan and Ng (2010) for a detailed comparison). While most looked into variants of PageRank, Litvak and Last (2008) experimented with the HITS algorithm (Kleinberg 1999), while Boudin (2013) investigated other indices like degree, betweenness and closeness.

The final important step in keyphrase extraction is *post-filtering*. Extracted phrases are disambiguated and normalized for morpho-syntactic variations and lexical synonymy (Csomai and Mihalcea 2007). Adjacent words are also sometimes collapsed into phrases, for a more readable output.

Benchmark datasets for keyword extraction include ICSI – a collection of meeting transcripts divided into 201 segments (Liu *et al.* 2009), NUS – a set of 211 academic papers (Nguyen and Kan 2007), INSPEC – 2,000 titles and abstracts from journal papers (Hulth 2003), and SemEval – 184 academic papers from SemEval 2010 Keyphrase Extraction Task (Kim *et al.* 2010). Given the preponderance of keyword-annotated datasets in the academic domain, most research in keyword extraction has focused on academic papers.

There are only three previous studies that we are aware of that considered keyword extraction from emails. Turney (2000) reports a study that pioneered email keyword extraction, but his dataset has not been released. Goodman and Carvalho (2005) worked with emails, but since their goal was to extract *implicit search queries* from emails, and not keywords, their dataset is not useful to us. Dredze et al. (2008) extracted summary keywords from emails using latent concept models, and evaluated the extracted keywords in two novel tasks – *automated foldering* (predicting which folder an email should go to), and *recipient prediction.* While Dredze et al.'s study did look into (unsupervised) email keyword extraction, they performed an extrinsic evaluation of their approach rather than intrinsically evaluating on a gold standard dataset.

Also relevant is the work by Laclavík and Maynard (2009), who discuss general strategies for email classification, storage, and integration with other information management systems. They further point out that email communication in a modern organization is mostly *action-oriented*, and that knowledge workers of all kinds interact with their emails on a daily basis. This stands in sharp contrast with keyword extraction in the academic domain,[5] where papers are meant for other researchers who are reasonably familiar with the domain – and therefore use formal and scientific vocabulary. Furthermore, academic communication via papers is *single-way* (author to audience), and *dissemination-oriented*. Also, unlike academic papers, emails mostly discuss topics *at hand*, including urgent ones. Hence, emails stand to benefit from their own keyword extraction system. In fact, Laclavík and Maynard briefly hinted at email keyword extraction as a way to combat the email information overload (cf. Section V).

3 Keyword Extraction Pipeline

Our keyword extraction systems proceed in five stages:

1. Email processing
2. Candidate extraction
3. Pre-processing
4. Ranking/Classification
5. Post-processing

As a first step, we sentence-segment each email manually, followed by tokenization based on whitespace. We ignore email metadata such as filename, ID, date, from and to, subject, and signature fields. This was done to ensure that our systems are only focusing on the email text. We also remove numbers and words consisting of one or two characters.

In the candidate extraction stage, we generate candidate phrases from a document. We experimented with four types of phrase candidates, and found noun phrases and named entities to be the best (Section 4.2).

In the pre-processing stage, we clean up the phrase candidates by removing punctuation, folding to lowercase, and removing numbers and leading and trailing stopwords. We also implemented a pre-processing heuristic, which is a syntactic filter that only considers nouns and adjectives while constructing the word co-occurrence network. This is based on the observation that most keywords consist of nouns and adjectives (along with function words), and therefore a part-of-speech filter at this stage can help eliminate some of the potential noise.

In the fourth and most important stage, we extract keywords from emails using two approaches – unsupervised, and supervised. In the unsupervised (ranking) approach, we (a) rank words using several linguistic and centrality-based features, and then *collapse* the top-ranked adjacent words to form keyphrases; (b) rank *candidate phrases* (noun phrases and named entities) using several phrase features – both linguistic and centrality-based, and then extract the top-ranked phrases as keyphrases. In the supervised (classification) approach, we classify candidate phrases as *keyphrase* vs. *non-keyphrase* using phrase features, and return the ones classified as *keyphrase*. Both approaches are evaluated on our own dataset consisting of 319 keyword-annotated emails.

[5] On which the current state-of-the-art is based (Kim *et al.* 2010).

In the fifth stage, we implement a post-processing heuristic (for word ranking) that constructs longer key phrases starting with the selected keywords, by *collapsing* adjacent words from the top k ranking into phrases.[6] The problem with this collapsing strategy is that the final number of phrases cannot be predicted from the number of input keywords k, and there is no control over the number of collapsed phrases. Other variants of this collapsing strategy that alleviate this problem are possible, but they are found to introduce new complications, e.g., very long keywords or several keywords that are semantically redundant. We therefore use the basic collapsing heuristic described before.

4 Features for Keyword Extraction

We extract two broad classes of features for keyword extraction from emails: word features and phrase features. These features are either used by themselves, in an unsupervised fashion, or together in a supervised setting.

In the following, we describe each feature, along with a short note on its potential utility in keyword extraction. Note that word and centrality features are extracted after removing stopwords.

4.1 Word Features

Word features are used to rank *word types* (i.e., unique words) based on their frequency, positional, and surface properties.

- **Tf**: Raw frequency of a word type in a document.[7] It is an important indicator of the word's saliency.
- **Tf.idf**: Raw frequency of a word type multiplied by its *idf* (inverse document frequency) computed on the British National Corpus (Clear 1993).
- **First position**: Position of the first occurrence of a word in a document. Position is measured by number of word tokens since the beginning of the document. Words appearing towards the beginning of a document often contain introductory information, thereby being important from the perspective of keyword extraction.
- **Last position**: Position of the last occurrence of a word in a document. Position, as before, is measured by number of word tokens since the beginning of the document. Words appearing near the end of a document may contain summary information, thereby becoming important.
- **Normalized first position**: First position feature, normalized by the number of words in the document.
- **Normalized last position**: Last position feature, normalized by the number of words in the document.

[6] To better understand this heuristic, consider the following example: Assume the words "house" and "white" have been returned as top-ranked for the (tiny) document "POTUS spoke in the White House". In this case, the collapsing heuristic will yield "White House" as a keyphrase.

[7] By "document", we mean an email or an email thread.

- **Word length**: Number of characters in a word. Longer words sometimes contain richer information owing to word-compounding, morphology, etc.
- **Is capitalized?**: Whether the word is capitalized. This is a binary feature. Word capitalization is often a strong cue for detecting named entities.
- **Is in subject?**: Whether the word appears in the subject line of an email/thread. This is another binary feature. Words appearing in an email's subject line often contain important information, much like the words in the title line of a general document (Jiang, Hu, and Li 2009).

We also implement word centrality features, which are features defined on word co-occurrence networks (Mihalcea and Tarau 2004). For each email, a word co-occurrence network is constructed by adding all the word types (i.e., unique words) as nodes, and by drawing an edge between the words that occur next to each other. Centrality measures on such co-occurrence networks can yield a powerful set of features for keyword extraction. In this work, we focus on the following centrality features:

- **Degree**: Number of edges incident to a node. Since word types implicitly endorse each other via collocation edges, the more edges that are incident to a word, the more important the word becomes.
- **PageRank**: Stationary probability of a random walk visiting a particular word in the word co-occurrence network. When used with teleportation, such a random walk ends up assigning higher probabilities to more important words in the network (Mihalcea and Tarau 2004).
- **Coreness**: Measure of how "deep" a word is in the co-occurrence network. The "deeper" a word, the more its importance. This feature is inspired by the *core-periphery structure* of small-world networks, and computed using the so-called *k-cores decomposition* (Seidman 1983; Batagelj and Zaveršnik 2003).
- **Neighborhood size (order one)**: Number of immediate neighbors to a node. It is a version of node degree that disregards *self-loops*, which can arise in word co-occurrence networks due to constructions such as "again, again and again". The more neighbors a node has, the higher its importance.

4.2 Phrase Features

In addition to features reflecting the importance of individual words, we also calculate phrase features, which are used to rank/classify entire *phrases*. More precisely, these features are used to classify *(document, phrase)* pairs, as explained in the next section.

Following (Csomai and Mihalcea 2007), we extract four types of candidate keywords: stopword-filtered n-grams (n = 1, 2, 3, 4), stopword-filtered base noun phrases, named entities extracted using the Stanford Named Entity Recognizer (NER) (Finkel, Grenager, and Manning 2005), and named entities extracted using an unsupervised heuristic (sequences of capitalized words that never appear without capitalization). We use the CRFTagger (Phan 2006) for part-of-speech tagging, and

Mark Greenwood's NP chunker for base noun phrase identification.[8] The following features are used to rank these candidate keywords:

- **Phrase Tf**: Raw frequency of a phrase in a document.
- **Phrase Idf**: Inverse document frequency of a phrase, computed as the average of *idf*s of its constituent words. The word *idf*s were computed on the British National Corpus.
- **Phrase Tf.idf**: Raw frequency of a phrase multiplied by its *idf*.
- **Within-document frequency**: Number of sentences a phrase appeared in (for a particular document). The more sentences a phrase appears in, the higher its importance.
- **Mean length of containing sentences**: Mean length of the sentences a phrase appeared in (for a particular document) – in word tokens, word types, and (non-space) characters. These three features encode the importance of the containing sentences. Longer sentences should carry more information.
- **Phrase length**: Length of a phrase calculated as the number of constituent word tokens and non-space characters. These two features encode the fact that longer phrases usually carry more information.
- **Length of the containing document**: Length of the document a phrase appears in – in word tokens, word types, and non-space characters. These three features encode the weight of the containing document.
- **Mean length of constituent words**: Average length of constituent words in non-space characters.
- **First and last containing sentences**: Index of the first and the last sentence a phrase appears in (for a particular document). These two features encode positional information of a phrase.
- **Diameter**: Difference between the indexes of the first and last containing sentences.
- **Wikipedia keyphraseness**: Ratio of the number of Wikipedia documents where a phrase appeared as a keyword, and the number of Wikipedia documents where the phrase appeared (Mihalcea and Csomai 2007). This ratio, when computed for phrases with reasonable document counts, provides an estimate of their importance as well as cohesiveness.
- **POS pattern probability**: Probability of a part-of-speech pattern emerging as a candidate keyword from among all base noun phrase patterns. This feature is inspired by a similar feature used in (Csomai and Mihalcea 2008). We included a second probability – probability of obtaining a candidate keyword pattern from among unique base noun phrase patterns. These two probabilities incorporate syntax information in our model.
- **Is in subject?**: Whether the phrase appears in the subject line of an email/thread. This is a binary feature.

[8] Available from `http://www.dcs.shef.ac.uk/~mark/nlp/software/gate-plugins/chunkerv11.zip`.

Table 1. *Keyphrase ranking obtained with three features.*

Mean neighborhood size		Mean coreness		tfidf	
afternoon	4	afternoon	6	enron	17.63
pen and pencil	3	phone	6	hector	10.62
tomatoes	2	week	6	chris	4.54
goody package	2	enron	6	tomatoes	4.12
hope	2	desk	6	goody package	3.95
york customers	2	rest	6	pen and pencil	3.48
rest	2	hector	4	golf shirt	2.52
week	2	chris	0	afternoon	2.41
golf shirt	2	golf shirt	0	desk	2.00
hector	2	new york	0	phone	1.59
desk	2	goody package	0	york customers	1.57
new york	2	tomatoes	0	new york	0.79
enron	2	pen and pencil	0	care	0.67
care	1	york customers	0	hope	0.55
phone	1	care	0	rest	0.49
chris	1	hope	0	week	0.41

- **Overlap with subject**: If the phrase appears in the subject line, then this feature is the length of the phrase in words, divided by the length of the subject line in words; otherwise, zero.
- **Are all words capitalized?**: This binary feature is a strong cue for detecting named entities.
- **Mean degree, PageRank, coreness, and neighborhood size**: Mean degree, PageRank, coreness, and neighborhood size (order one) of the constituent words of a phrase in the word co-occurrence network. Note that stopwords are not included in the word network. These four features indicate the importance of a phrase in terms of centrality. Higher values denote greater importance.
- **Phrase degree, PageRank, coreness, and neighborhood size**: Degree, PageRank, coreness, and neighborhood size (order one) of a phrase in the *phrase co-occurrence network*. Phrase co-occurrence networks are similar to word co-occurrence networks, except that nodes are candidate phrases instead of words, and edges are defined between candidate phrases that appear in the same sentence. Higher values indicate greater importance.

Note that among the above features, coreness and neighborhood size are novel features in our study, and to the best of our knowledge, they have never been used in keyword extraction. Further, their behavior is different from tfidf. We illustrate an example in Table 1 (ECS080; corporate single email), which shows that both the ranking as well as the value ranges are different for phrase tfidf, phrase mean coreness, and phrase mean neighborhood size. For example, the word "afternoon" has a mean neighborhood size of 4 (i.e., it is connected to 4 other phrases) and a mean coreness of 6, but a low tfidf of 2.41 ("afternoon" appears in several of the emails). Yet another example is the word "chris." It has a low mean neighborhood size of 1 and low mean coreness of 0, but the tfidf is much higher (4.54), showing once again that these features are not redundant in their behavior.

5 Evaluation

5.1 Dataset

To our knowledge, there is no dataset available for keyword extraction from emails. To evaluate our methods, we compiled our own dataset consisting of 212 single emails and 107 email threads (of 4-8 emails each) drawn from the Enron collection (Klimt and Yang 2004).[9]

First, each email and thread within this dataset was manually classified as either "private" or "corporate". The corporate emails discuss issues related to work and office, whereas the personal emails deal with issues related to home, family, and friends. Examples of corporate and personal emails are shown in Table 2.

Next, all the emails and threads in the dataset are annotated for keywords by four independent human judges. The annotators were asked to assign 5-20 keywords to each email/thread, ranked in their order of importance. We requested annotators to select keywords that are up to five words in length. While the definition of a "keyword" can vary depending on the annotator, we provided some guidelines and recommendations for increased consistency, e.g., we recommended the selection of noun phrases, named entities, or any other phrases that best capture the essence of a given email/thread.

Example keywords assigned by the annotators are shown in Table 2. Table 3 shows the keyword statistics of different categories of emails. Overall, threads have more keywords than single emails, and corporate emails have more keywords than personal emails.

We compute inter-annotator agreement by considering one annotator as the ground truth, and the (set union of) remaining annotators as the "system". Further, we consider three forms of agreement:

- **Exact match**: when two phrases match exactly (up to lowercasing and spaces).
- **BOW match**: when two phrases' bags of words (BOW) match exactly after lowercasing (except stopwords).
- **Relaxed match**: when two phrases either match exactly up to lowercasing, or can be made identical by adding a single word to the beginning or end of the shorter phrase (Chuang *et al.* 2012).

Micro-averaged precision, recall, F-score, and Jaccard similarity under these three

[9] An earlier version of this dataset has been described in detail in (Loza *et al.* 2014).

Table 2. *Example of a corporate email and a personal email, along with keyword annotations.*

Corporate Email	**Personal Email**
I am faxing you both the Master Firm Purchase/Sale Agreement executed between Enron Gas Marketing (merged now into ECT) and Aquila Energy Marketing Corporation (merged now into Utilicorp.) with respect to Utilicorp. “signing” the agreement in lieu of giving a guaranty. Actually what Utilicorp. did in this agreement is sign as a co-obligor under the agreement (see Section 16.12 of the agreement). They signed accepting joint and several liability with respect to the obligations. If we can get them to agree to the same language in your master agreement that would effectively be as good or better than getting a guaranty. Anything less (like just sticking their “name” on the signature line) may not get us much or be worthless. All this, of course, is subject to any differences between US and UK law or issues under UK law which I will leave in Edmund’s capable hands. Let me know if I can be of further service...	Hey, Does your email still work. I was wondering if you had a private email that would be appropriate for non business related correspondence?? Just curious. Still would like to find a way to keep in touch better, the messenger thing is ok, but its hard at times with work and all. At least with email we can say alittle more and at least have some uninterrupted time to “talk”. Speaking of talking?.is there any time that is good to call? I would like to hear your voice once in a while! J
Keywords assigned by Annotator 1: Master Firm Purchase/Sale Agreement, Utilicorp, guaranty, obligations, agree to the same language, signing, liability, worthless	Keywords assigned by Annotator 1: email, private email, keep in touch better, hear your voice, talk, non business related correspondence, messenger thing, good time to talk
Keywords assigned by Annotator 2: Master Firm Purchase/Sale Agreement, Enron Gas Marketing, Utilicorp., sign as a co-obligor, language, US and UK law	Keywords assigned by Annotator 2: private email, non business related correspondence, uninterrupted time, keep in touch, ‘talk’
Keywords assigned by Annotator 3: Master Firm Purchase/Sale Agreement, co-obligor, Utilicorp, Enron Gas Marketing, Aquila Energy Marketing Corporation, guaranty	Keywords assigned by Annotator 3: private email, non business related correspondence, messenger thing, uninterrupted time, voice
Keywords assigned by Annotator 4: agreement, Utilicorp, guaranty, Master Firm Purchase/Sale Agreement, signed	Keywords assigned by Annotator 4: keep in touch better, private email, call, messenger thing, hear your voice

settings are shown in Table 4. Note that the best agreement under exact match is only 33.63% F-score, which is not very high, thus indicating the difficulty of the keyword extraction task (cf. (Hasan and Ng 2014)).[10] However, if we consider the BOW match, the best agreement is much higher (51.63% F-score). The same holds

[10] Having said that, 33.63% F-score is close to what one can reasonably expect as an upper bound on the inter-annotator agreement. For example, in the SemEval 2010 Keyphrase Extraction Task, the F-score achieved by readers on author-assigned keyphrases was 33.6% (cf. (Kim *et al.* 2010), Section 4).

Table 3. *Keyword annotation statistics.*

Email Category	Mean #Keyphrases	Standard Deviation
Corporate Single	6.68	1.88
Corporate Thread	7.72	2.36
Personal Single	6.79	2.11
Personal Thread	7.36	2.51
All Single	6.70	1.97
All Thread	7.53	2.47
All Corporate	7.06	2.12
All Personal	6.96	2.26

true for the relaxed match. This shows that annotators – although clearly divergent in their opinion, do in fact tend to select very similar *words* to construct their keyphrases. Results on *pairwise agreement* (Table 5) present the same evidence.

5.2 Evaluation Settings and Metrics

Our experimental results are primarily based on a *combined gold standard*, obtained from the set union of the keywords assigned by the four annotators, with an average of 19.35 keywords per email. Note that we also considered the alternative of creating a gold standard by using the set intersection of the four annotations. This results in zero keywords per email, reflecting the diversity of opinions on the annotations for this task. Instead, we adopt as our intersection gold standard the *union of pairwise intersections* between the annotations, yielding 5.70 keywords per email (on average). This gold standard results in an artificially small dataset that does not accurately reflect the performance of a keyword extraction system. Nonetheless, for the sake of completeness, we also report the results obtained on this intersection set (Section 6.3).

All the keyword extraction experiments are evaluated using micro-averaged precision, recall, F-score, and Jaccard similarity. We used F-score as our primary yardstick for comparing different systems. The evaluations are performed at *phrase-level*, where we count a candidate phrase appearing in the gold standard as an exact match (up to lowercasing and spaces).

6 Results and Discussion

We perform two sets of experiments, one consisting of unsupervised methods that rely on the individual use of the features described in Section 4, and a second set consisting of a supervised framework that combines all the features using machine learning.

Table 4. *Inter-annotator Agreement.*

"Ground Truth" Annotator	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Exact match				
Annotator 1	27.00	40.96	32.55	23.23
Annotator 2	24.06	55.84	33.63	23.03
Annotator 3	21.16	58.27	31.04	20.96
Annotator 4	20.45	57.95	30.23	20.41
BOW match				
Annotator 1	46.37	58.25	51.63	47.17
Annotator 2	34.40	72.97	46.76	40.29
Annotator 3	33.35	74.33	46.04	40.26
Annotator 4	29.66	76.10	42.68	38.78
Relaxed match				
Annotator 1	40.59	61.59	48.93	44.43
Annotator 2	34.62	80.35	48.39	42.69
Annotator 3	30.71	84.56	45.05	39.14
Annotator 4	29.87	84.64	44.15	39.99

6.1 Unsupervised Methods

For unsupervised keyword extraction, we first apply the pre-processing heuristic to select only nouns and adjectives, then rank candidate words according to different features (one feature at a time), followed by a selection of the *top 50%* of the words from the resulting ranked list, and finally use the post-processing heuristic to collapse adjacent words into key phrases. For the binary features (such as *Is in subject?* and *Is the word / Are the words capitalized?*), we take all words/phrases with value 1 instead of *top 50%*.

Table 6 shows the performance values obtained for the word and phrase features, and Table 7 shows the values for binary features. Note that *mean neighborhood size* yields the best F-score, followed by *phrase tfidf* and *phase tf*. Among word features, *word tfidf* performs the best, followed by *word PageRank*. The superiority of *tfidf* in both cases is in line with the findings by Hasan and Ng (2010). For binary features (Table 6), we see that one of them gives the highest precision among all systems (47.11%). However, their recall is very low (2-16%), thereby yielding a relatively low F-score (esp. for *Is in subject?*).

Table 5. *Pairwise Inter-annotator Agreement.*

Annotator-pair	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Exact match				
1 – 2	32.19	24.56	27.86	20.04
1 – 3	35.01	23.27	27.96	19.22
1 – 4	34.61	22.42	27.21	19.58
2 – 3	37.42	32.60	34.85	25.42
2 – 4	36.90	31.32	33.88	24.98
3 – 4	30.58	29.80	30.18	21.01
BOW match				
1 – 2	53.50	37.44	44.05	38.86
1 – 3	56.54	38.07	45.50	40.17
1 – 4	56.66	33.81	42.35	39.68
2 – 3	48.28	46.45	47.34	40.53
2 – 4	51.62	44.00	47.50	43.61
3 – 4	46.90	41.56	44.07	39.43
Relaxed match				
1 – 2	49.32	37.63	42.69	39.50
1 – 3	52.58	34.95	41.99	38.15
1 – 4	51.00	33.03	40.09	37.38
2 – 3	51.35	44.74	47.82	45.94
2 – 4	51.34	43.58	47.14	45.15
3 – 4	45.11	43.96	44.53	40.83

6.2 Supervised Methods

For supervised keyword extraction, we apply the same steps as in the unsupervised methods, but perform the ranking of the candidates using a machine learning algorithm applied in leave-one-out cross-validation fashion using all the phrase features. The supervised system includes a few features that cannot be used for keyphrase ranking, but could be potentially useful for the selection of keywords (e.g., document-specific features such as *document length*).

The supervised framework is formulated as a binary classification task, where each *(document, candidate keyword)* pair is classified as relevant or not. Using a small development dataset of 30 emails, we tried nine different classification algorithms, including KNN, Naive Bayes, SVM SMO, J48 decision tree, PART rule

Table 6. *Performance of unsupervised keyword extraction. Best values in different columns are boldfaced.*

Feature	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Word features				
Tf	20.33	28.96	23.89	13.57
Tf.idf	**23.44**	**31.22**	**26.77**	**15.45**
First position	14.47	16.99	15.63	8.48
Last position	19.39	24.27	21.56	12.08
Word length	17.97	23.64	20.42	11.37
Degree	20.25	28.88	23.81	13.51
PageRank	21.41	28.37	24.41	13.90
Coreness	15.13	18.15	16.50	8.99
Neighborhood size	20.79	29.60	24.42	13.91
Phrase features				
Tf	25.65	33.16	28.92	16.91
Idf	25.63	33.14	28.91	16.90
Tf.idf	26.91	34.79	30.35	17.89
Wikipedia keyphraseness	22.68	29.32	25.58	14.66
Phrase length (words)	22.67	29.30	25.56	14.65
Phrase length (non-space chars)	25.02	32.34	28.21	16.42
Overlap with subject	21.91	28.32	24.71	14.09
Mean length of constituent words	25.27	32.67	28.50	16.62
Mean length of containing sentences in words	19.76	25.55	22.28	12.54
Mean length of containing sentences (unique words)	19.91	25.74	22.45	12.65
Mean length of containing sentences (non-space chars)	20.01	25.87	22.57	12.72
First containing sentence	17.35	22.43	19.56	10.84
Last containing sentence	20.56	26.58	23.18	13.11
Diameter	24.09	31.15	27.17	15.72
Within-document frequency	24.62	31.84	27.77	16.12
Mean degree	22.05	28.50	24.86	14.20
Mean PageRank	21.73	28.10	24.51	13.96
Mean coreness	20.58	26.61	23.21	13.13
Mean neighborhood size	**27.33**	**35.33**	**30.82**	**18.22**
Phrase degree	22.94	29.66	25.87	14.86
Phrase PageRank	23.71	30.66	26.74	15.44
Phrase coreness	21.09	27.26	23.78	13.49
Phrase neighborhood size	22.72	29.37	25.62	14.69

Table 7. *Performance of binary features in unsupervised keyword extraction. Best values in different columns are boldfaced.*

Feature	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Word features (binary)				
Is capitalized?	25.69	**16.97**	**20.44**	**11.38**
Is in subject?	**47.11**	2.66	5.04	2.59
Phrase features (binary)				
Are all words capitalized?	**33.29**	**11.34**	**16.91**	**9.24**
Is in subject?	30.96	2.70	4.96	2.54

Table 8. *Performance of supervised keyword extraction. Best values in different columns are boldfaced. Performance values are micro-averaged in leave-one-out cross-validation.*

Classifier	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
KNN	31.94	**50.03**	**38.99**	**24.22**
Naive Bayes	**45.40**	28.87	35.30	21.43

learner, OneR, Logistic Regression, AdaBoost and LogicBoost. We found that Naive Bayes and KNN performed best, and therefore used these classifiers in our experiments on the entire evaluation dataset. For all the classification experiments, we use Weka (Hall *et al.* 2009). The performance values are micro-averaged.

The results of the leave-one-out cross-validation on the entire dataset of 319 emails are shown in Table 8. Interestingly, the results are comparable to the inter-annotator agreement rates reported in Tables 4 and 5, which is an indication of how accurate our best systems are as compared to human performance.

For an additional analysis, we also determine and report the most discriminative features (by Information Gain), as shown in Table 9. Note that Phrase Tf.idf and Mean neighborhood size appear among the most discriminative features, which is not surprising since these two features are also among the best in the unsupervised approach (Table 6). Note further that Phrase Tf appears to be more discriminative than Phrase Tf.idf, and that the index of the first containing sentence and length of the containing document are also among most discriminative features.

Table 9. *Most discriminative keyword extraction features by Information Gain on the email dataset.*

Feature	Information Gain
Phrase Tf	0.04279
Phrase Tf.idf	0.03804
First containing sentence	0.03545
Length of containing document in word types	0.03287
Length of containing document in non-space characters	0.03284
Length of containing document in word tokens	0.03166
Mean neighborhood size	0.02775
Within-document frequency	0.02655

Table 10. *Results of* **in-domain training**. *Best values in different columns are boldfaced. Performance values are micro-averaged in leave-one-out cross-validation.*

Classifier	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
KNN	32.45	**51.64**	**39.85**	**24.89**
Naive Bayes	**45.69**	31.27	37.13	22.80

6.3 Additional Evaluations

To further analyze the results of our supervised methods, we perform three additional evaluations.

First, we evaluate the effect of *in-domain training*, where for each of the four categories of emails in our dataset – personal single, personal thread, corporate single, and corporate thread – we restrict the training set to other documents in the same category. Table 10 shows the overall results obtained in this evaluation. Although the in-domain constraint results in a net decrease of the training set size, performance values improved because emails are more similar within a category than across categories. The F-score improvement with respect to the open-domain results from Table 8 are relatively small: 0.67 percentage point for KNN and 1.37 percentage point for Naive Bayes. Acknowledging that the size of the data used to train these in-domain systems is smaller than that used to train the open-domain data, the lesson learned from this experiment is that if domain-specific data is available, the same performance can be obtained with a fraction of the data.

Second, we evaluate our supervised methods against a gold standard dataset formed by using the intersection of the *pairwise annotations* produced by the human judges. As noted in Section 5, taking the intersection results in a very small dataset,

Table 11. *Performance of supervised keyword extraction under* **intersection gold standard**. *Best values in different columns are boldfaced. Performance values are micro-averaged in leave-one-out cross-validation.*

Classifier	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
KNN	22.04	**35.87**	27.30	15.81
Naive Bayes	**28.56**	32.52	**30.41**	**17.93**

which is not ideal for measuring the performance of an automatic system. We nonetheless report these results in Table 11, to show the ability of our system to identify these keywords that were agreed upon by both annotators.

Finally, to understand the performance of our keyword extraction methods on different types of emails (e.g., single emails versus threads; personal emails versus corporate emails), we perform separate evaluations of our supervised methods on each of the four different subsets of our dataset. Table 12 shows these comparative results. As seen in the table, personal emails are significantly more difficult to process than corporate emails. The highest F-scores are obtained with the KNN classifier on single emails, which may be due to the fact that there is less variance in the topics covered by the emails in this data (as opposed to threads, where there may be topic shifts).

6.4 Comparison with Existing Systems

To place our results in perspective, using our email dataset we evaluate five previously introduced systems for keyword extraction. We chose two state-of-the-art unsupervised keyword extraction systems – **SingleRank** and **ExpandRank** (Wan and Xiao 2008; Hasan and Ng 2010), two top-performing systems in SemEval 2010 keyphrase extraction task (Kim *et al.* 2010) – **KX_FBK** (Pianta and Tonelli 2010) and **SZTERGAK** (Berend and Farkas 2010; Berend 2011), and **KEA** (Witten *et al.* 1999) – a well-known supervised keyword extractor.[11] Table 13 shows the results obtained by these five systems, in comparison with our two best unsupervised methods, and our two supervised settings. Our Naive Bayes system gives the best precision, which is very encouraging in a subjective task like keyword extraction. Overall, our systems are found to be better than the state-of-the-art, with our KNN system leading to the best F-score (38.99%), which is 12.76% better than the best state-of-the-art system (SZTERGAK) on this dataset. This improvement

[11] We used Kazi Saidul Hasan's C++ implementation of SingleRank and ExpandRank, the publicly available TextPro implementation of KX_FBK (`http://textpro.fbk.eu/`), and the gitHub (Java) implementation of SZTERGAK (`https://github.com/begab/kpe`). KEA source code is available from `https://code.google.com/archive/p/kea-algorithm/downloads`.

Table 12. *Performance of supervised keyword extraction on subsets of our dataset: single emails; threads; personal emails; corporate emails.*

Classifier	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Single emails				
KNN	36.02	53.17	42.94	27.34
Naive Bayes	51.48	24.03	32.76	19.59
Threads				
KNN	26.78	45.46	33.71	20.27
Naive Bayes	40.72	35.92	38.17	23.59
Personal emails				
KNN	30.64	46.48	36.93	22.65
Naive Bayes	48.19	27.05	34.65	20.96
Corporate emails				
KNN	32.89	52.77	40.52	25.41
Naive Bayes	43.66	30.27	35.76	21.77

is significant ($p < 0.00001$) using a two-sample test for equality of proportions with continuity correction. Also, our unsupervised systems performed better than state-of-the-art systems, with best F-score of 30.82%.

6.5 Post-hoc Evaluation of Keyphrases

As a final evaluation, we set up two experiments that allow us to measure the quality of the keywords extracted by our system in an extrinsic way.

First, we perform a post-hoc evaluation, where the keywords produced by our system are manually annotated by a human judge for appropriateness. We set this evaluation as follows: for a given email (single or thread), first the human judge carefully reads the email text to make sure she is familiar with its content; next, the judge is presented with a set of keywords, and her task is to determine which of the keywords reflect important content of the email text.

We take a random sample of 40 single emails (20 personal, 20 corporate), and 20 thread emails (10 personal, 10 corporate), and generate keyphrases using our best system (KNN). The human judge then classifies these keyphrases as appropriate or not, as described above. Table 14 shows the fraction of keywords found to be

Table 13. *Comparison with existing systems. Best values in different columns are boldfaced. Performance values are micro-averaged. Systems marked with ^{O} are ours, ^{U} are unsupervised, and ^{S} are supervised. KX_FBK and SZTERGAK are two of the top performers in* SemEval *2010 keyphrase extraction task.*

System	Precision (%)	Recall (%)	F-score (%)	Jaccard (%)
Phrase Tf.idf^{UO}	26.91	34.79	30.35	17.89
Mean neighborhood size^{UO}	27.33	35.33	30.82	18.22
KNN^{SO}	31.94	**50.03**	**38.99**	**24.22**
Naive Bayes^{SO}	**45.40**	28.87	35.30	21.43
SingleRank^{U}	36.77	19.13	25.16	14.39
ExpandRank^{U}	36.61	19.05	25.06	14.32
KX_FBK^{U}	24.47	25.35	24.90	14.22
SZTERGAK^{S}	41.03	19.27	26.23	15.09
KEA^{S}	26.52	6.91	10.96	5.80

Table 14. *Keyphrase appropriateness in a post-hoc evaluation.*

Email Category	Returned Keyphrases	Appropriate Keyphrases	Percentage
Corporate Single	141	111	78.72
Corporate Thread	73	63	86.30
Personal Single	109	94	86.24
Personal Thread	60	51	85.00

correct for each email type. The results suggest that in such a post-hoc evaluation, a significantly larger fraction (78-86%) of the keywords produced by our system are found to be acceptable by a human judge. This is in line with previous work on keyword extraction (Csomai and Mihalcea 2008), which showed that there can be large gaps between the ad-hoc and post-hoc evaluations of keywords, as humans often have a difficult time generating a comprehensive list of keywords for a given

Table 15. *Email classification results. Standard deviations in parentheses.*

	Text only	Keyphrase only
Accuracy (%)	90.0 (11.83)	85.0 (10.25)
Time (Seconds)	10.07 (1.10)	6.53 (1.13)

text, yet they do agree with the appropriateness of a larger set of keywords when presented to them.

The second experiment consists of an application-based evaluation, where we simulate a potential classification task that a user has to accomplish when presented with a set of emails (e.g., the daily incoming email). Specifically, a human judge is given the task to classify each email in a set as being either “personal” or “corporate.” We compare the scenario where the classification is performed by only reading the extracted keywords, versus reading the entire text, and measure both the correctness of the classification (against our own existing gold standard annotations) as well as the time it takes to perform the task in each scenario.[12]

We perform 10 rounds of simulation, where in each round we select 10 random emails and their corresponding keyphrases. The presentation order of the email texts or email keyphrases is randomized to remove any sequence effect. The classification accuracy of this simulation process – averaged over the 10 rounds – is shown in Table 15 (standard deviations in parentheses). Note that just by reading the keyphrases, the human judge was able to correctly classify the emails 85% of the time, whereas reading the full text leads to 90% – which is only 5% improvement. Note further that the time taken to classify the emails just by reading the keyphrases is 6.53 seconds on average, whereas reading the full text takes at least 10 seconds. This shows that keyword extraction can be very helpful in real life by substantially reducing the time taken to *triage* emails, at comparable accuracy levels.

7 Conclusion

Keyword extraction from emails is largely an open problem, with potentially important benefits given the growing number of emails that we have to handle in our daily communication. In this paper, we described and evaluated methods for unsupervised and supervised keyword extraction from emails. We defined two types of features – word features and phrase features – which we then evaluated on a novel dataset consisting of emails manually annotated with keywords. To the best of our knowledge, our work is the first attempt to extract keywords from emails after the seminal study by Turney (2000). Our unsupervised experiments highlighted the role played by the different features for keyword extraction from emails. We also combined all the features using a supervised framework, and obtained results that improved significantly over the use of individual features. The results obtained with our best system represent a significant improvement over state-of-the-art in general-purpose keyword extraction, which is an encouraging result given the informal nature of emails and their difference from academic abstracts. Moreover,

[12] While we acknowledge that in a real-life setting, the name of the sender is often sufficient to classify an email as either personal or corporate, we use this task as an approximation for a generic email classification task. We believe this approximation is reasonable, given the fact that the human judge performing the task is (1) not provided with the sender name; and (2) is agnostic to the personal and corporate relationships of the actual email owner.

two extrinsic evaluations have further demonstrated the quality of the keywords extracted with our system.

The manually annotated email dataset introduced in this paper is publicly available from `http://lit.eecs.umich.edu`.

8 Acknowledgments

We are grateful to the annotators who made this work possible. This material is based in part upon work supported by Samsung Research America under agreement GN0005468 and by the National Science Foundation under IIS award #1018613. Any opinions, findings, conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of Samsung Research America or the National Science Foundation. We also thank the anonymous reviewers whose insightful comments helped improve the draft substantially.

References

Batagelj, Vladimir, and Zaveršnik, Matjaž. 2003. An O(m) Algorithm for Cores Decomposition of Networks. *CoRR* cs.DS/0310049.

Berend, Gábor, and Farkas, Richárd. 2010. SZTERGAK : Feature Engineering for Keyphrase Extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation.*

Berend, Gábor. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, 1162–1170. Chiang Mai, Thailand: Asian Federation of Natural Language Processing.

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3:993–1022.

Boudin, Florian. 2013. A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing.*

Chuang, Jason, Manning, Christopher D., and Heer, Jeffrey. 2012. "Without the Clutter of Unimportant Words": Descriptive Keyphrases for Text Visualization. *ACM Trans. Computer-Human Interaction*, 19(3):19:1–19:29.

Clear, Jeremy H. 1993. The British National Corpus. In Landow, G. P., and Delany, P., eds., *The digital word.* Cambridge, MA, USA: MIT Press. 163–187.

Csomai, Andras, and Mihalcea, Rada. 2008. Linguistically Motivated Features for Enhanced Back-of-the-Book Indexing. In McKeown, K., Moore, J. D., Teufel, S., Allan, J., and Furui, S., eds., *ACL*, 932–940.

Csomai, András, and Mihalcea, Rada. 2007. Investigations in Unsupervised Back-of-the-Book Indexing. In *FLAIRS Conference*, 211–216.

Dredze, Mark, Wallach, Hanna M., Puller, Danny, and Pereira, Fernando. 2008. Generating Summary Keywords for Emails Using Topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '08).* ACM, New York, NY, USA. 199–206.

Ferrara, Felice, Pudota, Nirmala, and Tasso, Carlo. 2011. A Keyphrase-Based Paper Recommender System. In Agosti, M., Esposito, F., Meghini, C., and Orio, N., eds., *Digital Libraries and Archives*, volume 249 of *Communications in Computer and Information Science.* Springer Berlin Heidelberg. 14–25.

Finkel, Jenny R., Grenager, Trond, and Manning, Christopher. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the Association for Computational Linguistics*, ACL '05, 363–370.

Goodman, Joshua, and Carvalho, Vitor R. 2005. Implicit Queries for Email. In *CEAS*.

Grineva, Maria, Grinev, Maxim, and Lizorkin, Dmitry. 2009. Extracting Key Terms From Noisy and Multi-theme Documents. In *Proceedings of the 18th International World Wide Web Conference*, WWW 2009, 661–670.

Hall, Mark, Frank, Eibe, Holmes, Geoffrey, Pfahringer, Bernhard, Reutemann, Peter, and Witten, Ian H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11(1):10–18.

Hasan, Kazi Saidul, and Ng, Vincent. 2010. Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 365–373.

Hasan, Kazi Saidul, and Ng, Vincent. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1262–1273.

Hulth, Anette. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 conference on Empirical Methods in Natural Language Processing*, EMNLP '03, 216–223.

Jiang, Xin, Hu, Yunhua, and Li, Hang. 2009. A Ranking Approach to Keyphrase Extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 756–757. ACM.

Kim, Su Nam, Medelyan, Olena, Kan, Min-Yen, and Baldwin, Timothy. 2010. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, 21–26. Stroudsburg, PA, USA: Association for Computational Linguistics.

Kleinberg, Jon M. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46(5):604–632.

Klimt, Bryan, and Yang, Yiming. 2004. Introducing the Enron Corpus. In *First Conference on Email and Anti-Spam (CEAS)*.

Laclavík, Michal, and Maynard, Diana. 2009. Motivating Intelligent E-mail in Business: An Investigation into Current Trends for E-mail Processing and Communication Research. In *IEEE Conference on Commerce and Enterprise Computing. CEC '09.*.

Lee, Sungjick, and Kim, Han-joon. 2008. News Keyword Extraction for Topic Tracking. In *Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management - Volume 02*, NCM '08, 554–559. Washington, DC, USA: IEEE Computer Society.

Li, Zhenhui, Zhou, Ding, Juan, Yun-Fang, and Han, Jiawei. 2010. Keyword Extraction for Social Snippets. In *Proceedings of the 19th international conference on World wide web*, WWW '10, 1143–1144.

Litvak, Marina, and Last, Mark. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, 17–24. Stroudsburg, PA, USA: Association for Computational Linguistics.

Liu, Feifan, Pennell, Deana, Liu, Fei, and Liu, Yang. 2009. Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, 620–628.

Liu, Zhiyuan, Huang, Wenyi, Zheng, Yabin, and Sun, Maosong. 2010. Automatic Keyphrase Extraction via Topic Decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 366–376.

Loza, Vanessa, Lahiri, Shibamouli, Mihalcea, Rada, and Lai, Po-Hsiang. 2014. Building a Dataset for Summarization and Keyword Extraction from Emails. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association (ELRA), Reykjavik, Iceland. 26–31.

Mihalcea, Rada, and Csomai, Andras. 2007. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, 233–242.

Mihalcea, Rada, and Tarau, Paul. 2004. TextRank: Bringing Order into Texts. In Lin, D., and Wu, D., eds., *Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2004*.

Nguyen, Thuy Dung, and Kan, Min-Yen. 2007. Keyphrase Extraction in Scientific Publications. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, 317–326.

Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry. 1998. The PageRank Citation Ranking: Bringing Order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, 161–172.

Phan, Xuan-Hieu. 2006. CRFTagger: CRF English POS Tagger.

Pianta, Emanuele, and Tonelli, Sara. 2010. KX: A Flexible System for Keyphrase eXtraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation.*

Seidman, Stephen B. 1983. Network structure and minimum degree. *Social Networks* 5(3):269–287.

Tomokiyo, Takashi, and Hurst, Matthew. 2003. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*, 33–40. Association for Computational Linguistics.

Tonella, Paolo, Ricca, Filippo, Pianta, Emanuele, and Girardi, Christian. 2003. Using Keyword Extraction for Web Site Clustering. In *Web Site Evolution, 2003. Theme: Architecture. Proceedings. Fifth IEEE International Workshop on*, 41–48.

Turney, Peter D. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval* 2(4):303–336.

Wan, Xiaojun, and Xiao, Jianguo. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, 855–860. AAAI Press.

Witten, Ian H., Paynter, Gordon W., Frank, Eibe, Gutwin, Carl, and Nevill-Manning, Craig G. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, 254–255.

Yih, Wen-tau, Goodman, Joshua, and Carvalho, Vitor R. 2006. Finding Advertising Keywords on Web Pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, 213–222. New York, NY, USA: ACM.