NamedKeys: Unsupervised Keyphrase Extraction for Biomedical Documents

Zelalem Gero
zgero@emory.edu
Emory University
Atlanta, USA

Joyce C. Ho
joyce.c.ho@emory.edu
Emory University
Atlanta, USA

ABSTRACT

A vast amount of biomedical literature is generated and digitized every year. As a result is a growing need to develop methods for discovering, accessing, and sharing knowledge from medical literature. Keyphrase extraction is the task of summarizing a text by identifying the key concepts. The keyphrases can be single-word or multi-word linguistic units which can concisely represent a document. Although a variety of models have been proposed for automated keyphrase extraction, the performance is poor in comparison with other natural language processing tasks. The problem is even more daunting for biomedical domain where the text is filled with highly domain-specific terminologies. We propose a new method, NamedKeys, to automatically identify meaningful and informative keyphrases from biomedical text. NamedKeys integrates named entity recognition, phrase embedding, phrase quality scoring, ranking, and clustering to extract author-assigned keywords from biomedical documents. Performance evaluation on PubMed abstracts demonstrates that NamedKeys achieves significant improvements over existing state-of-the-art keyphrase extraction models. Furthermore, we propose the first benchmark dataset for keyphrase extraction from biomedical text.

CCS CONCEPTS

• **Information Extraction**; • **Health care information systems**; • **Health informatics** ; • **Summarization**;

KEYWORDS

Keyphrase extraction, Document summarization, concept extraction, Phrase embedding

ACM Reference Format:
Zelalem Gero and Joyce C. Ho. 2019. NamedKeys: Unsupervised Keyphrase Extraction for Biomedical Documents. In *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB '19), September 7–10, 2019, Niagara Falls, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3307339.3342147

ACM-BCB '19, September 7–10, 2019, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-6666-3/19/09…$15.00
https://doi.org/10.1145/3307339.3342147

1 INTRODUCTION

There has been an exponential growth in biomedical literature with over 28 million articles indexed by PubMed [1]. Thus, information extraction is a key component in automated text processing as it facilitates the acquisition of structured information. Keyphrase extraction, the identification of single-world or multi-word linguistic units that concisely represent a document, is a crucial aspect of information extraction. Keyphrases help readers rapidly understand, organize, access, and share information of a document by providing a short summary of the document. Extracting keyphrases from documents is of paramount importance for natural language processing (NLP) tasks such as text summarization [2, 37], text classification [11], topic detection [22, 44], recommendation systems[30, 38], citation summarization [34] and information visualization [9]. Scientific publishers use keyphrases to identify potential reviewers for submitted articles, recommend articles to readers, and suggest missing citations to authors [1].

A variety of models have been introduced for keyphrase extraction due to its widespread use [18]. Existing keyphrase extraction systems are either supervised or unsupervised. Supervised methods train classifiers on labeled examples and require large domain-specific annotations. Unfortunately, such labeled data is typically unavailable in the biomedical domain as the annotation process is labor intensive and usually necessitates significant domain expertise. Unsupervised methods, on the other hand, rely on word co-occurrence statistics from large external corpora such as Wikipedia and WordNet. Large external corpora are good statistical approximations for general domain keyphrase extraction but lack good representation in domain specific settings like biomedical text where the vocabulary can be significantly different. Moreover, many of the unsupervised approaches focus on word level co-occurrence and prefer keyphrases containing highly ranked words. These biases results towards keyphrases with more number of words. Recent unsupervised approaches such as graph-based methods and topic-based methods offer better keyphrase generation yet can suffer from a lack of diversity of the extracted keyphrases or generate phrases that may not be meaningful. Despite these efforts, the task remains challenging and the performance of current systems remains poor in comparison to other NLP tasks [24].

The challenge of keyphrase extraction is even more daunting in the biomedical domain where the text contains highly domain-specific terminologies. Given the vast amount of biomedical literature generated and digitized every year, there is a growing need to develop methods for discovering, accessing, and sharing knowledge from medical literature [35]. Significant portions of research articles published in medical journals do not have author-assigned

[1] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

keyphrases. Even when they come with keyphrases, the number of author-assigned keyphrases available with the articles is too limited to represent the topical content of the articles. This makes an automatic keyphrase extraction process highly desirable. Despite this need, keyphrase extraction for biomedical text has been largely ignored by the research community. Only two works [23, 36] have been introduced and either have been demonstrated on a small set of documents or necessitated a hand-curated list of keyphrases.

A similar line of work related to keyphrase extraction is the use of Medical Subject Headings (MeSH) by a prominent medical literature database, MEDLINE. MeSH is a controlled set of terms manually assigned by human indexers in the National Library of Medicine. Even though MeSH terms make it easier to search for a document and cluster similar documents, generating MeSH terms for every document is expensive and time-consuming; new articles are not immediately indexed until 2 or 3 months later and approximately ten dollars per article is spent for the manual indexing [28]. Moreover, MeSH terms are 'controlled language' meaning users searching for a document need to use the exact terminology (or a term in the MeSH vocabulary) which are not easy for end users to define compared to more natural languages of author keywords which are not bound by any vocabulary. Neveol et al. [32] found that about 75% of keywords authors considered as important to describe the content of their own articles could not be matched to MeSH terms. Hence, there is a need to develop an automated keyphrase generation system that can extract author-assigned keywords from biomedical documents.

To address the challenges of keyphrase generation from biomedical literature, we propose `NamedKeys`, a novel method to produce meaningful and informative keyphrases to summarize biomedical text. Our method uses biomedical-specific Named Entity Recognition (NER) to identify words and phrases that are important in the text which are mostly named entities(NE). We also introduce a data-driven phrase-based embedding process to extract the most descriptive phrases from a given text. We propose a new phrase scoring criteria to identify meaningful phrases. Finally, we employ a ranking and clustering approach to identify diverse keyphrases that best reflect the document. Our experimental results on 3049 PubMed abstracts illustrate the power of `NamedKeys`.

Our approach attains better precision and recall scores compared to other state-of-the-art keyphrase generation algorithms, including up to a 35% F1 score improvement over the next best model. We summarize our major contributions in this work.

- **Improved generation of candidate keyphrases**: Since the quality of selected top keyphrases is dependent on the candidate keyphrases generated, we employ NER and a new phrase generation mechanism to identify possible candidate keyphrases.
- **Better document and keyphrase representation**: While word embedding methods are popular representations for text, biomedical literature contains many multi-word phrases which may not be reflected in compositions of single word vectors. We propose a data-driven mechanism for identifying common phrases in a large corpus and learn joint single word and multi-word representations that better capture the semantic meaning of the phrases. Documents are represented as inverse document frequency (IDF) weighted averages of the learned word and phrases vectors.
- **New score to measure phrase quality**: We propose a new metric to measure the quality of a multi-word phrase. Our metric accounts for the frequency of the phrase itself and the frequency of the constituent words.
- **Diverse and representative extracted keyphrases**: We propose to rank the candidate keyphrases based on the semantic similarity using our phrase embeddings and the new phrase quality score. The selection of top keyphrases is diversified using clustering and preventing semantically similar terms from being selected.
- **Creation of a new benchmark dataset**: In the general domain, there are several publicly available datasets for keyphrases extraction. Unfortunately, no such dataset exists for biomedical literature. We created the *first* benchmark from PubMed Central Open Access articles and author-submitted keyphrases for reproducibility and to inspire future research.

The next section presents a brief description of the previous works related to the work presented in this paper. The proposed keyphrase extraction method has been discussed in section 3. Section 4 and 5 present experimental results and conclusion respectively.

2 RELATED WORK

Keyphrases extraction methods can be categorized into unsupervised and supervised approaches [18]. Supervised approaches require training data that contain a collection of documents with their labeled keyphrases. Many of such approaches pose the problem of keyphrase extraction as a binary classification problem [40, 41, 45]. These methods use various learning algorithms to train a classifier on datasets annotated with positive and negative keyphrases. Support vector machines [20, 26], multi-layer perceptron [21] and maximum entropy [21, 46] are a few of the commonly used learning algorithms. Others pose the problem as a ranking problem where the ranker learns to rank between two candidate keyphrases [20]. These supervised approaches use different statistical and structural features from within the document and outside resources to build their models. Term frequency-inverse document frequency, the number of times a phrase occurs in a document, where the phrase first occurs in a document, the number of words preceding the phrases first occurrence and the part-of-speech of word(s) in a phrase are commonly used features [18].

Unsupervised approaches do not require labeled data and hence are preferred for domains with limited training data. The first common unsupervised approach is graph-based methods. These methods build a graph from the input document and rank its nodes per their importance using a graph-based ranking method. All the candidate keyphrases will be vertices and the connection between each candidate is represented by edges. Based on the relatedness between the candidates, the weights in each vertex is determined. Different relatedness measures are employed to measure the relationship between candidates. TextRank [29], a prototypical graph-based algorithm, starts by assigning arbitrary values to each node in the graph, and iterates until convergence below a given threshold is

achieved. After running the algorithm, a score is associated with each vertex, which represents the importance of the vertex within the graph. Vertices with more connections and connections from other important nodes are ranked higher.

The second common unsupervised approach is topic-based methods. Such approaches cluster candidate keyphrases into topics in the document so that all the topics in the input document are represented by the selected keyphrases. This helps to assure all topics in the input document are represented in the final keyphrases and the selected keyphrases are relevant to one or more topic(s) in the document. [25] clusters semantically similar candidates using Wikipedia and co-occurrence-based statistics and selects candidates close to the centroid of each cluster. The assumption is that each cluster will cover a topic from the input document and choosing candidates from each topic ensures exhaustively representing all topics. [24] modifies [25] to weigh each candidate based on the probability of the topics it belongs to ensure more important topics get more weight and hence more number of selected keyphrases.

Yet another unsupervised approach focuses on statistical features that do not require external dictionaries or corpora. Methods which use this approach rely on features extracted from the documents in the current corpus such as the position of the first occurrence of a candidate, word frequency, casing, and how often a candidate word appears in different sentences [8, 12]. Recently, due to the popularity of word embedding methods, keyphrase extraction methods are using document semantic similarity measured in word vector cosine similarity to weigh the relatedness between candidates [27, 43].

Keyphrase extraction in biomedical domain has been experimented by few researchers. In [23], Li et al. extract noun phrases from medical literature as keyphrase candidates and assign weights to extracted noun phrases for a medical document based on how important they are to that document and how domain-specific they are in the medical domain using WordNet lexical database and Specialist Lexicon. Even though this work is a pioneer in the extraction of keyphrases from medical documents, the use of a very small test set of 60 documents is not large enough for conclusive results. In [36], Sarkar presents a hybrid approach to keyphrase extraction from medical documents. The approach is an amalgamation of two methods: the first one assigns weights to candidate keyphrases based on combination of features such as position, term frequency, inverse document frequency and the second one assign weights to candidate keyphrases using some knowledge about their similarities to the structure and characteristics of keyphrases available in the memory (stored list of keyphrases). This approach necessitates the availability of hand-curated keyphrases in memory to learn from making it harder to use in an unsupervised setting.

3 METHODOLOGY

Extracting keyphrases from text can be considered as selecting phrases that capture the gist of the document and are also semantically and syntactically correct. In [39] these two measures are referred to as informativeness and phraseness. Informativeness measures how good a phrase is in capturing the main theme of the document while phraseness measures the likelihood of a sequence of words to be a meaningful phrase. We propose `NamedKeys` – a new keyphrase extraction algorithm, that produces informative and meaningful keyphrases for biomedical text. Our model, illustrated in Figure 1, consists of the following steps: (1) new candidate keyphrase generation mechanisms to construct an extensive keyphrase candidate set; (2) a new phrase-embedding representation for the document and the phrases to better measure the informativeness of a given phrase; (3) a new "phraseness" metric to assign a normalized score for every phrase generated from the corpus; and (4) a ranking and clustering module that ranks the candidate phrases and clusters the keyphrases to ensure that the extracted keyphrases are diverse. The details for each of the four steps are discussed in the following subsections.

3.1 Candidate Keyphrase Generation

We propose two new mechanisms for generating keyphrase candidates. The first process uses named entity recognition (NER) to extract information such as disease names, medication, symptoms, and chemicals. Instead of constructing multi-word phrases based on important consecutively occurring words, we start with phrases as a single unit of representation. We observed that many keyphrases in biomedical literature capture concepts such as chemicals, diseases, cell types, proteins, and gene named entities. Biomedical-specific NER has been shown to help identify, problems and symptoms a patient has exhibited, tests that have been run, and treatments that have been administered [3]. Biomedical named entities are important concepts that improve our understanding of medical text and our ability to analyze them; identifying, for example, problems and symptoms a patient has exhibited, tests that have been run, and treatments that have been administered [3]. Unfortunately, biomedical named entities may not occur frequently enough in the text, and thus are often not suggested by existing keyphrase extraction tools. Thus, we used SciSpacy [31], a specialized NLP library for processing biomedical texts, to detect all the named entities in the text. SciSpacy contains different modules for chemicals and disease named entities; cell types; proteins and gene named entities; and cell lines, DNA, RNA and cancer named entities.

The second mechanism finds phrases that are not named entities but are still potentially meaningful. Rather than rely only on the NER, we propose a generic approach to extract more candidate phrases. `NamedKeys` chunks the text by identifying potential keyphrase boundaries using stopwords and punctuation [36]. From the generated chunks, those which belong to the following parts of speech will be retained: 'JJ', 'JJR', 'JJS', 'NN', 'NNS', 'NNP', 'NNPS'. We perform the parts of speech selection using Genia Tagger[2]; a biomedical tool for text processing. Although stop words and punctuations will never occur in any proposed keyphrase, it provides a systematic methodology for generating variable n-graph keyphrases. The detected named entities from the NER process and from this chunking process are combined to construct the candidate keyphrase pool.

Figure 2 illustrates a comparison of `NamedKeys` against baseline keyphrase generation models. The baseline algorithms used for comparative analysis are detailed in Section 4.3. Our method extracts six of the seven named entities correctly as keyphrases while the other models extract a fewer number of keyphrases.

[2]http://www.nactem.ac.uk/GENIA/tagger/

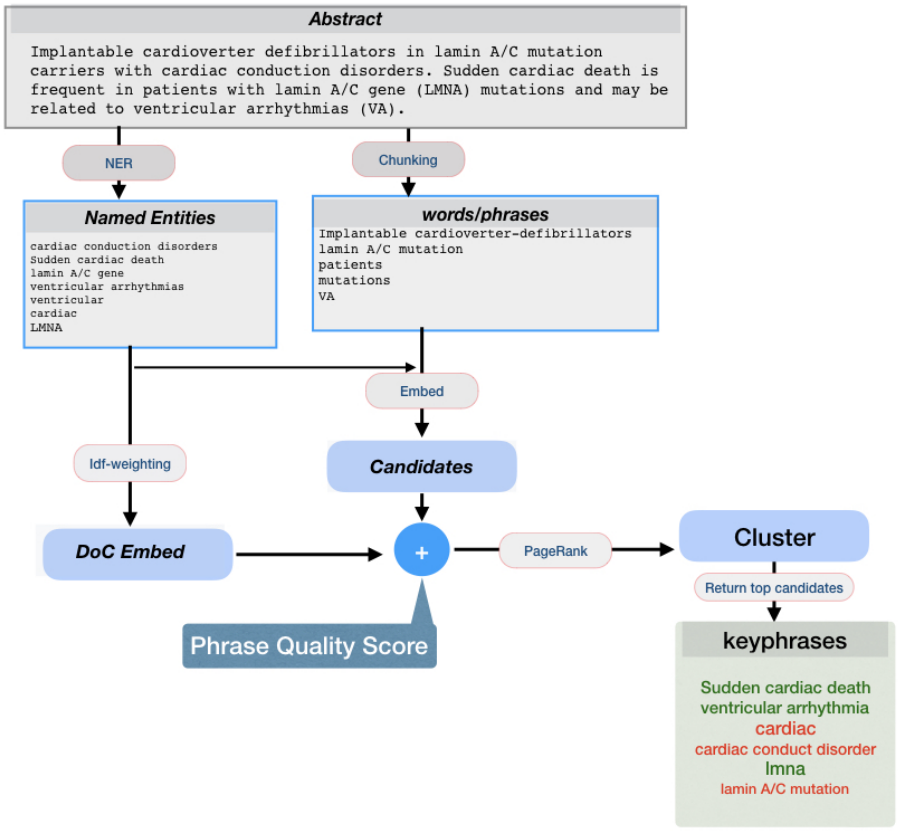

Figure 1: An illustration of our workflow. The abstract is preprocessed to extract Named Entities (NEs) and noun phrase chunks. Inverse document frequency (idf) scores are calculated for each NE and this will be used to calculate the document embedding by performing idf-weighted vector averaging of the NEs. NEs and the noun phrases will be candidate keyphrases and their similarity to the document embedding and phrase quality will be calculated. The two scores will be used to build the PageRank algorithm. Finally, we cluster the candidates based on their semantic similarity to each other and top candidates from each cluster will be returned.

3.2 Phrase Embedding

The next step of `NamedKeys` focuses on identifying the candidate keyphrases with high informativeness measures. We propose the use of word embeddings to help rank the candidate phrases based on closeness to the document. Word embeddings are dense-low dimensional vector representations of words such that related words are close in vector space. Each dimension in the vector represents a feature of a word, and the vector can, in theory, capture both semantic and syntactic features of the word. Word2Vec [17], Glove [16], and FastText [4] are commonly used approaches to train word vectors. Unfortunately, many of the common word embedding approaches and pre-trained vectors focus on unigrams, while key concepts in biomedical literature are often expressed as multi-word phrases [33].

In this work, we develop a phrase-based embedding model to capture the semantic and syntactic relation between terms (or n-grams). We use a data-driven approach of extracting a commonly occurring sequence of words and learn embeddings for the extracted phrases along with the single words. [16] showed that the presence of unigram words intermixed with multi-word phrases improves the performance of embedding models. To avoid pre-specification of the number of words for a phrase, we used a similar idea as the second mechanism in the keyphrase candidate generation step. We identify potential phrase boundaries using stopwords and punctuations (excluding the hyphen). A sequence of words that occur more than a pre-defined threshold (100 is used for our experiments) are considered potential phrases. Phrases are merged into a single word (e.g., prostate cancer becomes prostate_cancer) in the order they originally appeared in the text. The multi-word phrases are trained with all the single words in the corpus. We use the Word2Vec tool to train our phrase embedding model on over 27 million PubMed abstracts which took 5hrs 34 minutes on a Macbook Pro machine with Intel Dual Core i7@2.2Ghz CPU and 16GB RAM. Thus, our embeddings can capture the semantic relation between related concepts like "Hypertension" and "High Blood Pressure".

Once the vector representations are available for all the terms in a document as well as the candidate keyphrases, `NamedKeys`

Oral mucosal manifestations in some genodermatoses: correlation with cutaneous lesions. The clinical picture of several genetic skin diseases may include the presence of oral mucosal lesions. These manifestations, however, have not been granted much attention in most dermatological publications. In this article, we fully review the oral mucosal lesions of tuberous sclerosis, dyskeratosis congenita, lipoidoproteinosis, Cowden disease, Darier's disease and pachyonychya congenita and compare these with their respective cutaneous lesions. Some dental aspects are discussed as well. This unifying approach may allow a better understanding of these oral lesions, avoiding obscure nomenclature and classification.

Algorithms	tuberous sclerosis	dyskeratosis congenita	lipoidoproteinosis	Cowden disease	Darier's disease	pachyonychya congenita	oral lesions
TopicRank[7]	✓	✓	✓	✓			
SingleRank[41]		✓		✓		✓	✓
TextRank[28]		✓				✓	✓
PositionRank[13]		✓		✓		✓	✓
Tfldf							✓
YAKE[8]			✓				
KPminer[12]			✓				
MultiPartiteRank[6]		✓	✓	✓			
NamedKeys	✓	✓	✓	✓	✓	✓	

Figure 2: Named entities correctly extracted as keyphrases by various methods

measures the informativeness of the candidates compared to the candidate vectors of the document. Given the importance of the named entities in the biomedical documents, we represent the document, D using the IDF-weighted sum of the named entities:

$$D = \frac{\sum_{i=1}^{n} IDF_i * w_i}{\sum_{i=1}^{n} IDF_i},$$

where w_i is the corresponding vector representation of the named entity and IDF_i is the inverse document frequency of the named entity. IDF of a given named entity is calculated as the inverse of the number of documents the term appears in. This is used for attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination. Then to calculate the informativeness of a given candidate phrase, we compute the cosine similarity between the document and the keyphrase representation, w_k:

$$Similarity = \frac{D^T w_k}{||D||||w_k||} \quad (1)$$

3.3 Phrase Quality

A candidate keyphrase can have a high cosine similarity to the document and can still not be a syntactically meaningful phrase. As an example, 'ventricular arrhythmias vary" could have a high cosine similarity to a document discussing "ventricular arrhythmias" but should not be ranked high since the phrase is not syntactically sound. A better phrase would be just "ventricular arrhythmias". Although there are several common phrase ranking criteria [10, 19], we found they offered a poor trade-off between phrase frequency, constituent word frequency, and phrase length. For example, pointwise mutual information (PMI) is often used to find good collocation pairs as it calculates the probability of co-occurrence relative to the probabilities of the occurrence of each word. Conversely, phrases that contain frequently occurring words will have small PMI scores

The predictive value of endorectal 3 Tesla multiparametric magnetic resonance imaging for extraprostatic extension in patients with low, intermediate and high risk prostate cancer.We determined the positive and negative predictive values of multiparametric magnetic resonance imaging for extraprostatic extension at radical prostatectomy for different prostate cancer risk groups. We evaluated a cohort of 183 patients who underwent 3 Tesla multiparametric magnetic resonance imaging, including T2-weighted, diffusion weighted magnetic resonance imaging and dynamic contrast enhanced sequences, with an endorectal coil before radical prostatectomy. Pathological stage at radical prostatectomy was used as standard reference for extraprostatic extension. The cohort was classified into low, intermediate and high risk groups according to the D'Amico criteria. We recorded prevalence of extraprostatic extension at radical prostatectomy and determined sensitivity, specificity, positive predictive value and negative predictive value of multiparametric magnetic resonance imaging for extraprostatic extension in each group. Univariate and multivariate analyses were performed to identify predictors of extraprostatic extension at radical prostatectomy. The overall prevalence of extraprostatic extension at radical prostatectomy was 49.7% ranging from 24.7% to 77.1% between low and high risk categories. Overall staging accuracy of multiparametric magnetic resonance imaging for extraprostatic extension was 73.8%, with sensitivity, specificity, positive predictive value and negative predictive value of 58.2%, 89.1%, 84.1% and 68.3%, respectively. Positive predictive value of multiparametric magnetic resonance imaging for extraprostatic extension was best in the high risk cohort with 88.8%. Negative predictive value was highest in the low risk cohort with 87.7%. With an odds ratio of 10.3 multiparametric magnetic resonance imaging is by far the best preoperative predictor of extraprostatic extension at radical prostatectomy. For adequate patient counseling, knowledge of predictive values of multiparametric magnetic resonance imaging for extraprostatic extension is of utmost importance. High negative predictive value, important for decisions on nerve sparing strategies at radical prostatectomy, is only reached in low risk subjects.

Keyphrases **correctly** extracted by NamedKeys:

prostate cancer	0.81
radical prostatectomy	0.84
magnetic resonance imaging	0.92
positive predictive value	0.93
negative predictive value	0.93
multivariate analyses	0.91
endorectal coil	0.89
extraprostatic extension	0.75

Common phrases **incorrectly** extracted by baseline methods:

different prostate cancer risk groups	0
high risk prostate cancer	0.32
high risk cohort	0.17
high risk categories	0.12
tesla multiparametric magnetic resonance imaging	0
low risk subjects	0.1
low risk cohort	0.1
multiparametric magnetic resonance imaging	0.25
multiparametric magnetic	0
determined sensitivity	0

Figure 3: The "phraseness" score of extracted keyphrases by `NamedKeys` and baselines. The phrases in red are from NER while the others are from chunking.

even if the phrase is good. To measure the phraseness of a candidate keyphrase, we propose "Information Frequency (Info_Freq)", a new ranking metric based on the phrase frequency and constituent words frequency in the corpus [15]. This criteria achieves the state-of-the-art performance when the resulting distributed word representations are evaluated on five benchmark datasets for biomedical semantic similarity. Our metric adds a multiplier to the PMI index that captures the overall frequency of the phrase:

$$\text{Info\_Freq}(x, y) = \log \frac{p(x, y)}{p(x)p(y)} * \log(\text{freq}(x, y)) \quad (2)$$

where $p(x, y)$ is the probability of the two words occurring together, $p(x)$ is the probability of the first word in the text and $p(y)$ is the probability of the second word in the text. Thus, Info_Freq measures the phraseness of a sequence of words by considering how often the phrase and the constituent words occur. Info_Freq is calculated for all the candidate keyphrases and the scores are normalized to lie between 0 and 1.

Figure 3 provides an illustrative example of a PubMed abstract with the Info_Freq scores for phrases generated by `NamedKeys` and common baselines. The baseline algorithms used for comparative analysis are detailed in Section 4.3. From the figure, we can observe that semantically meaningful phrases have high scores and are highly likely to be assigned as keyphrases by the authors. On the other hand, most of the phrases on the bottom right of the figure are not semantically meaningful with low phrase quality scores and hence are not considered appropriate keyphrases by the authors.

3.4 Candidate Clustering and Ranking

The final two steps of `NamedKeys` are to rank the candidates using the informativeness and meaningfulness measures, as well as cluster the candidate keyphrases to avoid redundancy of keyphrases. While phrase embedding and phrase quality capture the general informativeness and meaningfulness of a given candidate phrase respectively, we use local co-occurrence of two candidates to capture the local relationship between the phrases within the context of the given document using a weighted PageRank algorithm [18]. The document is represented as a weighted undirected graph, where vertices correspond to the words/phrases and the edges represent the co-occurrence relations between two terms. Two vertices are connected if they occur in the same sentences. For a graph with V vertices and E edges, the score for vertex v_i is calculated as:

$$S(v_i) = (1-d)W_i + d \sum_{j \subseteq in(v_i)} \frac{sim(v_i, v_j)}{|out(v_j)|} S(v_j), \quad (3)$$

where W_i is the importance weight of vertex i measured as the average of its similarity to the document vector and its phrase quality, *sim* is the cosine similarity between keyphrases v_i, v_j and *out* is the number of outgoing edges of keyphrase v_j, and d is a damping factor. Thus, high scores reflect phrases that capture the gist of the document and are also semantically and syntactically correct.

Unfortunately, two common problems with keyphrase extraction algorithms are overgeneration and redundancy of keyphrases. Overgeneration errors occur when a system correctly predicts a candidate as a keyphrase because it contains a word that appears frequently in the associated document, but at the same time erroneously outputs other candidates as keyphrases because they contain the same word. Redundancy errors occur when a system correctly identifies a candidate as a keyphrase, but at the same time outputs a semantically equivalent candidate (e.g., it's alias) as a keyphrase. A recent study performed error analysis on the various algorithms and showed that 52-64% of the errors were due to overgeneration and redundancy of keyphrases [18]. For existing algorithms that rely on frequency, it can be difficult to avoid overgeneration errors as rejecting a non-keyphrase containing a word with a high term frequency might negatively impact the precision of the algorithm. On the other hand, redundancy errors occur due to the inability to detect that two candidates are semantically equivalent.

To overcome the overgeneration and redundancy error, we propose clustering the candidate keyphrases based on their semantic similarity. The cluster analysis achieves two purposes: identify keyphrases that are semantically similar and diversify the generated document keyphrases. The clustering algorithm uses the cosine similarity scores for all pairs of the candidate keyphrases to identify keyphrases that are similar. The importance of the cluster is then calculated based on the average distance of each of its candidates to the document. The cluster importance weights are then normalized to sum up to 1, and are used to determine the composition of the extracted keyphrases. For example, a cluster with a weight of 0.5 will provide approximately 50% of the final generated keyphrases, while a cluster with 0.1 weight will contribute 10%. To further avoid redundancy, keyphrases will not be selected if they are too similar (e.g., $sim(v_i, v_j) \geq \alpha$ where $\alpha = 0.75$ in our experiments). While any clustering algorithm based on distances can be used, we used the Affinity Propagation clustering algorithm[14] with a damping factor of 0.85 and Euclidean affinity. One benefit for Affinity Propagation is that the number of clusters is automatically learned from the data.

Figure 4 provides an example text with the clustered candidates. We can observe that each cluster focuses on a particular topic, which aids in diversifying the final selected keyphrases instead of taking the top-ranked candidates only. Moreover, we can see the potential benefit of not selecting keyphrases that are too similar from cluster 1. A keyphrase algorithm that selects both "photosensitizer methylene blue" and "photosensitizer methylene" as important keyphrases will suffer from overgeneration error. By introducing the threshold (α) to avoid selecting too similar phrases, we can reduce the possibility of overgeneration errors.

4 EXPERIMENTS

4.1 Dataset

To the best of our knowledge, keyphrase extraction for biomedical text has not been studied except for the two works mentioned in the related works. As a result, there is no benchmark dataset for this problem. We created the *first* dataset using the PubMed Central Open Access Subset articles. This dataset was constructed by selecting all the abstracts which contain at least 5 author-provided keyphrases. Five is chosen as the minimum number of keyphrases since most evaluation benchmarks are done at a minimum of five keyphrase extraction. Since the focus of this work is keyphrase extraction, we propose that the author-provided keyphrases serve as appropriate summarizations of their articles. Thus, we did not consider abstracts where there are no author-provided keyphrases or abstracts where one or more keyphrases are not in the abstracts.

While the PubMed Central Open Access Subset contains over 27 million articles at the time of download, only 3049 articles had a title, abstract, and at least five author-provided keyphrases found in the abstract. A value of 0.85 is used for the damping factor d as this gave the best score on a separate training set of 2000 abstracts. This training set is different from this benchmark test set as we used abstracts with less than five author provided keyphrases.

In addition to creating the first dataset, we have made it publicly available [3]. We hope that by releasing our dataset, we can facilitate reproducibility and foster a new community to solve this important problem. In our benchmark dataset, we provide the following fields for each article:

- title: the title of the article,
- abstractText:the abstract of the article,
- keyphrases: a list of keyphrases provided by the authors

4.2 Evaluation Metric

The keyphrase extraction algorithms are evaluated using exact match. In other words, the extracted keyphrase must have the same exact word or words. All the algorithms are evaluated using precision, recall, and F1 which are standard evaluation metrics for

[3] www.github.com/zelalemgero/namedkeys/testset

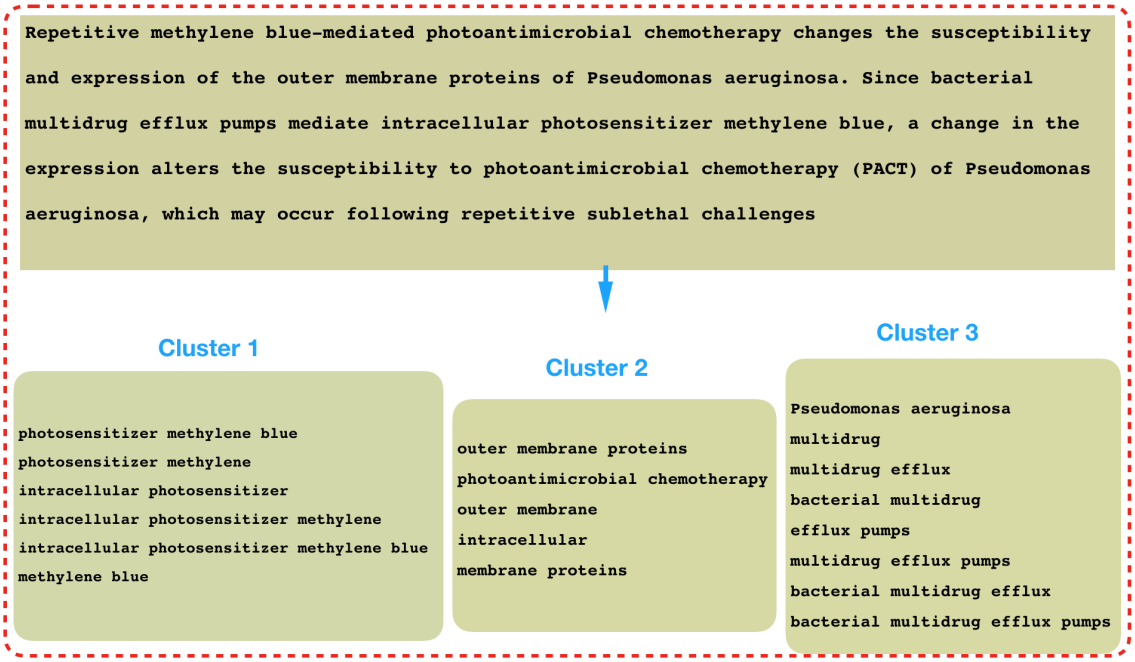

Figure 4: The impact of clustering candidate keyphrases based on their semantic similarity to each other.

keyphrase generation [5]. The three measures are defined as:

$$Precision = \frac{the\ number\ of\ correctly\ matched}{total\ number\ of\ extracted} = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{the\ number\ of\ correctly\ matched}{total\ number\ of\ ground\ truth} = \frac{TP}{TP + TN} \quad (5)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (6)$$

4.3 Baseline Methods

We compare `NamedKeys` with the state-of-the-art keyphrase extraction approaches implemented by Boudin et al. [5]. Graph-based approaches commonly use PageRank algorithm to determine the importance of candidates by using incoming and outgoing vertices to/from each candidate. Candidates with connections to other important candidates will have higher rank while candidates with fewer connections or connections to less important vertices will be ranked lower. Statistical-based approaches rely on features extracted from the document such as the position of first occurrence of a candidate, word frequency, casing, and how often a candidate word appears in different sentences. These approaches commonly use external corpus like Wikipedia to construct the features and perform well in a general domain while graph-based methods have the benefit of performing well in any domain since they do not depend on specific corpus features. We could not find the implementations of two baselines [27, 43] reported performing well in the general domain. The results we got by implementing the algorithms were worse than the other baselines used here. Hence, we did not report the scores from those baselines. For the graph-based approaches the following are used as baselines:

- **MultiPartiteRank** [6]: An approach that encodes the topical information within a multipartite graph structure and exploits their mutually reinforcing relationship to improve candidate ranking.
- **PositionRank**[13]: An algorithm that incorporates information from all positions of a word's occurrences into a biased PageRank algorithm.
- **SingleRank**[42]: A method that encodes the mutual influences of multiple documents within a cluster context.
- **TextRank**[29]: A model that accounts for the local context of a text unit (vertex) and the information recursively drawn from the entire text (graph).
- **TopicRank**[7]: A graph-based method that relies on a topical representation of the document.

For the statistical-based approaches the following are used as baselines:

- **TFIDF**: Term frequency-inverse document frequency, a common weighting technique in information retrieval and text mining.
- **KP Miner**[12]: A model that makes use of the first position a candidate phrase appears and the TFIDF measure as a weight.
- **YAKE**[8]: A feature-based system for multi-lingual keyword extraction from single documents.

4.4 Evaluation Results

First, we evaluated the algorithms based on the number of phrases extracted. Figure 5 shows the F1 scores for `NamedKeys` and the other

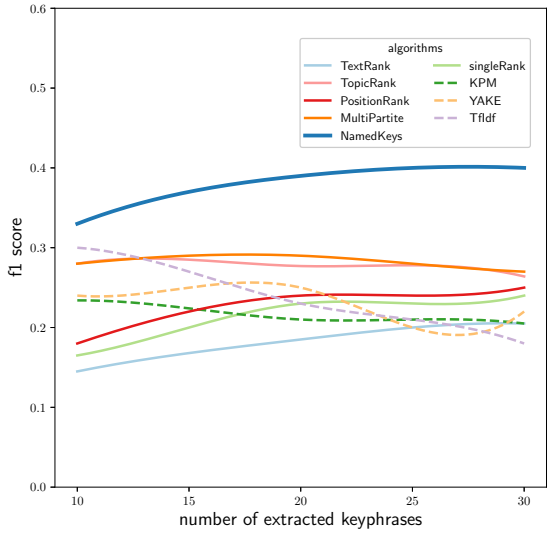

Figure 5: A comparison of the F1 score comparison across the various keyphrase generation models.

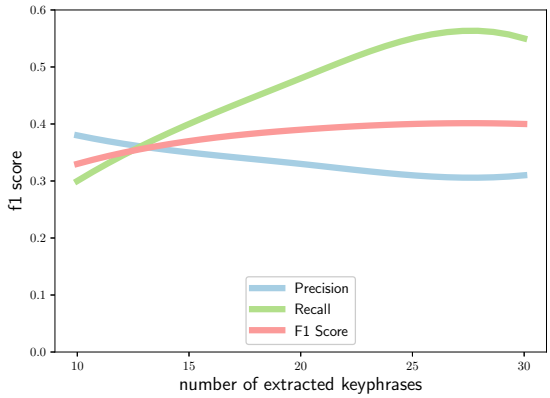

Figure 6: The precision, recall and F1 score of `NamedKeys` as a function of the number of extracted keyphrases.

baseline approaches based on the extracted number of phrases between ten and thirty in increments of five. `NamedKeys` consistently achieves the highest score with a performance gain of up to 35% from the next best method. We also observe that almost all of the algorithms achieve the highest F1 score when evaluated at 30. This intuitively makes sense as the algorithms can achieve higher recall without much loss in precision as the number of extracted phrases increases. From the figure, the three statistical approaches (KP Miner, YAKE and TFIDF) achieve the worst F1 scores overall. This can be attributed to the fact that such approaches mainly focus on the number of times the candidate occurs and the position of

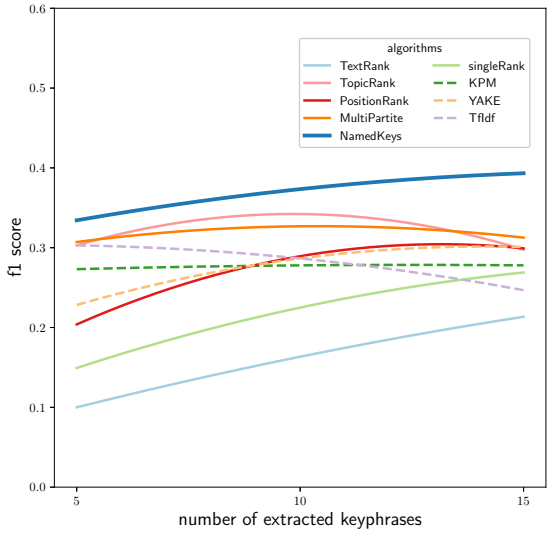

Figure 7: Comparison of F1 scores on short abstracts (ones with ten or lower assigned keyphrases).

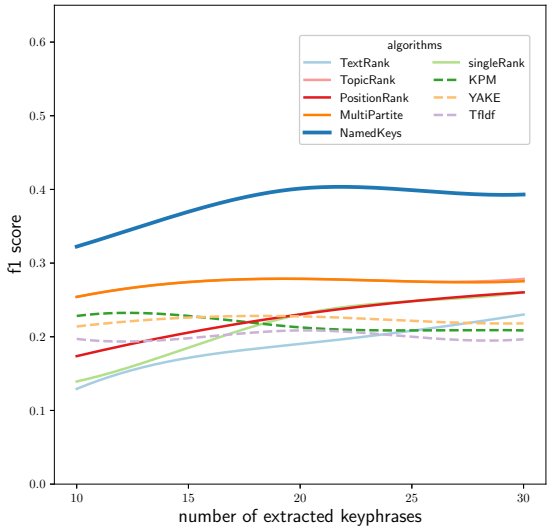

Figure 8: Comparison of F1 scores on long abstracts (ones with eleven or more assigned keyphrases).

first occurrence. These heuristics are not typically important in biomedical documents as phrases can be very important without having to occur multiple times.

We further investigate the impact of the number of extracted keyphrases on NamedKeys. Figure 6 shows the precision, recall, and F1. As more number of keyphrases are extracted, the precision plateaus while the recall goes up. This shows that the candidate phrases have good quality and we can improve the F1 score by extracting more keyphrases without affecting the accuracy.

The number of keyphrases submitted by authors in the PubMed articles varies widely ranging from as low as two to as high as thirty. A fewer number of keyphrases are usually submitted for shorter abstracts while longer abstracts have more keyphrases. To evaluate the performance of NamedKeys and the baseline methods with respect to the size of abstracts, we categorized the test set into short and long abstracts. Short abstracts are the ones with ten or less assigned keyphrases (710 abstracts) while long abstracts have more than ten keyphrases (2339 abstracts). Figure 7 shows the performance of the methods on short abstracts, where we only evaluate the algorithms at five, ten, and fifteen keyphrases. Even though NamedKeys still achieves the best average score, the performance gains are not as large as the full test set (shown in Figure 5). This is due to the fact that there are fewer possible keyphrases when the abstract text is short. Figure 8 summarizes the performance of the various models on long abstracts. We note that as the size of the text gets larger, the baseline methods fail to extract relevant keyphrases. However, NamedKeys maintains a consistent performance across both long and short abstracts.

We also performed an ablation experiment to quantify the performance gains for the various NamedKeys modules. We explored four different settings:

- No NER: This is without using named entity recognition to generate candidate phrases. Here we use only phrases that are extracted by chunking on punctuations and stopwords. These phrases are represented using phrased embedding.
- NER + Embed: This used named entity recognition to generate candidate phrases. Only named entities identified by NER are used without chunked phrases. Named entities will be embedded as single units to learn their vector representations.
- NER + Embed + chunk: Here we included named entities generated by NER and phrase chunks generated by our chunking method. All the generated phrases will be embedded as a unit by phrase embedding.
- NER + Embed + chunk + phraseQuality: This is the final complete method including all the proposed modules. The added phraseQuality component makes sure the candidates have phraseness scores above the threshold set to be considered candidates.

The results of the ablation experiment are shown in Table 1. Without named entities, the performance is quite poor. Adding NER and the phrase embeddings further improved the precision and recall at all the different number of extracted keyphrases. The inclusion of phrase chunking helped improve precision slightly, with minimal impact to recall. The addition of the phrase quality measure has a major impact in improving the recall, especially as the number of extracted keyphrases improves. Although phrase quality negatively affects precision, the gain from recall makes up for the overall improvement in the F1 score.

Method		@10	@15	@20	@25	@30
No NER	**P**	0.24	0.20	0.17	0.15	0.13
	R	0.17	0.20	0.24	0.26	0.30
	F1	0.19	0.20	0.20	0.19	0.18
NER + Embed	**P**	0.39	0.37	0.37	0.36	0.36
	R	0.22	0.29	0.36	0.39	0.42
	F1	0.28	0.32	0.36	0.37	0.39
NER + Embed+ chunk	**P**	**0.39**	**0.37**	**0.38**	**0.37**	**0.37**
	R	0.22	0.29	0.36	0.38	0.42
	F1	0.28	0.32	0.37	0.37	0.39
NER + Embed + chunk + phraseQuality	**P**	0.38	0.35	0.33	0.31	0.31
	R	**0.30**	**0.40**	**0.48**	**0.55**	**0.55**
	F1	**0.34**	**0.37**	**0.39**	**0.40**	**0.40**

Table 1: The effect of various modules of NamedKeys.

We also explored the impact of the clustering algorithm on the model performance. While the previous experimental results use the Affinity Propagation, we also experimented with spectral clustering and agglomerative clustering with a different number of clusters. Table 2 shows the results of the various clustering techniques as a function of the number of clusters evaluated using 25 extracted keyphrases. As a baseline comparison, Affinity Propagation achieved an F1 score of 0.4 for the same number of extracted keyphrases. Thus, the clustering technique has minimal impact on the final results.

no. of clusters	Spectral Clustering	Agglomerative Clustering
1/4n	0.40	0.41
1/3n	0.40	0.40
1/2n	0.39	0.40
2/3n	0.41	0.39
4/5n	0.40	0.40

Table 2: Comparison of F1@20 for different clustering techniques with various settings for the number of clusters.

5 CONCLUSION

In this paper, we introduced NamedKeys – a method composed of four modules: Named Entity recognition, Phrase embedding, Phrase quality score and similarity-based clustering for keyphrase extraction from biomedical documents. To evaluate the proposed method, we created a new publicly available benchmark dataset from PubMed Central Open Access articles. Our unsupervised approach results in performances much better than all the eight statistical and graph-based baselines at various numbers of keyphrases extracted.

One potential limitation of this work is the relatively small size of the benchmark dataset. To create the benchmark dataset, we relied on author-provided keyphrases which are submitted during article publication. Most of the PubMed Central Open Access articles lack author-provided keyphrases or only just a few phrases are submitted. As part of our future work, we plan to expand the benchmark dataset by including keyphrases assigned by domain experts.

REFERENCES

[1] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853* (2017).

[2] Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242* (2017).

[3] Willie Boag, Elena Sergeeva, Saurabh Kulshreshtha, Peter Szolovits, Anna Rumshisky, and Tristan Naumann. 2018. CliNER 2.0: Accessible and Accurate Clinical Concept Extraction. *arXiv preprint arXiv:1803.02245* (2018).

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[5] Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan, 69–73. http://aclweb.org/anthology/C16-2015

[6] Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. *arXiv preprint arXiv:1803.08721* (2018).

[7] Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*. 543–551.

[8] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. YAKE! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*. Springer, 806–810.

[9] Jason Chuang, Christopher D Manning, and Jeffrey Heer. 2012. "Without the clutter of unimportant words": Descriptive keyphrases for text visualization. *ACM Transactions on Computer-Human Interaction (TOCHI)* 19, 3 (2012), 19.

[10] Young Mee Chung and Jae Yun Lee. 2001. A corpus-based approach to comparative evaluation of statistical term association measures. *Journal of the American Society for Information Science and Technology* 52, 4 (Jan. 2001), 283–296.

[11] Frans Coenen, Paul Leng, Robert Sanderson, and Yanbo J Wang. 2007. Statistical identification of key phrases for text classification. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 838–853.

[12] Samhaa R El-Beltagy and Ahmed Rafea. 2010. Kp-miner: Participation in semeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*. 190–193.

[13] Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1105–1115.

[14] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[15] Zelalem Gero and Joyce C. Ho. 2019. PMCVec: Distributed phrase representation for biomedical text processing. *Journal of biomedical Informatics, in press* (2019).

[16] Glove vec [n. d.]. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/.

[17] Google [n. d.]. word2vec: Tool for computing continuous distributed representations of words. https://code.google.com/archive/p/word2vec/.

[18] Kazi Saidul Hasan and Vincent Ng. 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)*, Vol. 1. 1262–1273.

[19] Aminul Islam, Evangelos E Milios, and Vlado Keselj. 2012. Comparing word relatedness measures based on Google n-grams. In *Proceedings of COLING 2012: Posters*. 495–506.

[20] Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 756–757.

[21] Su Nam Kim and Min-Yen Kan. 2009. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*. Association for Computational Linguistics, 9–16.

[22] G Hemantha Kumar, Seyedmahmoud Talebi, and K Manoj. 2017. Users' Topic Detection from Tweets based on Keyword Extraction. *International Journal of Computer Applications* 975 (2017), 8887.

[23] Quanzhi Li and Yi-Fang Brook Wu. 2006. Identifying important concepts from medical documents. *Journal of biomedical informatics* 39, 6 (2006), 668–679.

[24] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 366–376.

[25] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 257–266.

[26] Patrice Lopez and Laurent Romary. 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics, 248–251.

[27] Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 634–639.

[28] Yuqing Mao and Zhiyong Lu. 2017. MeSH Now: automatic MeSH indexing at PubMed scale via learning to rank. *Journal of biomedical semantics* 8, 1 (2017), 15.

[29] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

[30] Naw Naw and Ei Ei Hlaing. 2013. Relevant words extraction method for recommendation system. *Bulletin of Electrical Engineering and Informatics* 2, 3 (2013), 169–176.

[31] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. *arXiv preprint arXiv:1902.07669* (2019).

[32] Aurélie Névéol, Rezarta Islamaj Doğan, and Zhiyong Lu. 2010. Author keywords in biomedical journal articles. In *AMIA annual symposium proceedings*, Vol. 2010. American Medical Informatics Association, 537.

[33] Aditya Parameswaran, Hector Garcia-Molina, and Anand Rajaraman. 2010. Towards the web of concepts: Extracting concepts from large datasets. *Proceedings of the VLDB Endowment* 0, 1-2 (2010), 566–577.

[34] Vahed Qazvinian, Dragomir R Radev, and Arzucan Ozgur. 2010. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010)*. 895–903.

[35] Wullianallur Raghupathi and Viju Raghupathi. 2014. Big data analytics in healthcare: promise and potential. *Health information science and systems* 2, 1 (2014), 3.

[36] Kamal Sarkar. 2013. A hybrid approach to extract keyphrases from medical documents. *arXiv preprint arXiv:1303.1441* (2013).

[37] Kamal Sarkar. 2014. A keyphrase-based approach to text summarization for English and bengali documents. *International Journal of Technology Diffusion (IJTD)* 5, 2 (2014), 28–38.

[38] Stamatina Thomaidou and Michalis Vazirgiannis. 2011. Multiword keyword recommendation system for online advertising. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 423–427.

[39] Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*.

[40] Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval* 2, 4 (2000), 303–336.

[41] Peter D Turney. 2002. Learning to extract keyphrases from text. *arXiv preprint cs/0212013* (2002).

[42] Xiaojun Wan and Jianguo Xiao. 2008. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 969–976.

[43] Rui Wang, Wei Liu, and Chris McDonald. 2014. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Software Engineering Research Conference*, Vol. 39.

[44] Christian Wartena and Rogier Brussee. 2008. Topic detection by clustering keywords. In *2008 19th International Workshop on Database and Expert Systems Applications*. IEEE, 54–58.

[45] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI Global, 129–152.

[46] Wen-tau Yih, Joshua Goodman, and Vitor R Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*. ACM, 213–222.