Journal of Biomedical Informatics: X 3 (2019) 100047

Contents lists available at ScienceDirect

Journal of Biomedical Informatics: X

journal homepage: www.journals.elsevier.com/journal-of-biomedical-informatics-x

PMCVec: Distributed phrase representation for biomedical text processing

Zelalem Gero*, Joyce Ho

Emory University, Department of Computer Science, Atlanta, USA

ARTICLE INFO

Keywords:
Phrase embeddings
Biomedical NLP
PubMed abstracts

ABSTRACT

Distributed semantic representation of biomedical text can be beneficial for text classification, named entity recognition, query expansion, human comprehension, and information retrieval. Despite the success of high-quality vector space models such as Word2Vec and GloVe, they only provide unigram word representations and the semantics for multi-word phrases can only be approximated by composition. This is problematic in biomedical text processing where technical phrases for diseases, symptoms, and drugs should be represented as single entities to capture the correct meaning. In this paper, we introduce `PMCVec`, an unsupervised technique that generates important phrases from PubMed abstracts and learns embeddings for single words and multi-word phrases simultaneously. Evaluations performed on benchmark datasets produce significant performance gains both qualitatively and quantitatively.

1. Introduction

The biomedical sciences are pioneers for open-access publication, with the PubMed database alone indexing over 27 million journal articles. Given the rich knowledge contained in these articles, obtaining insights from the publications can be used to address a variety of biomedical problems. The sheer volume of unannotated text dwarfs that of the annotated documents and hence it is imperative to utilize unsupervised machine learning models to capture the semantic meaning of words and phrases from such large corpus which in turn can be used for various downstream biomedical tasks.

For many Natural Language Processing (NLP) tasks based on vector space models, the text is transformed into meaningful vector representations to help improve performance. Recent efforts have introduced new neural network models that can induce semantically meaningful word representations (or embeddings) from large corpora [1,27,36,3]. Dense, low-dimensional vector representation of words are learned such that similar words are close in space. The ability to preserve semantic and syntactic similarities between words been shown to be very useful in a variety of NLP tasks including information retrieval [12], part-of-speech (POS) tagging [9], text summarization [39,46], sentiment analysis [13,24], named entity recognition (NER) [23,42], synonym extraction [18] and relation extraction [19]. Moreover, several biomedical domain word representations have been created from biomedical literature [21,38] and the impact of training word vectors on corpus from various domains for downstream biomedical tasks is explored by [43,33].

Although word embeddings have achieved great success in word-oriented tasks such as NER and POS tagging, they perform poorly on phrases-oriented tasks such as Semantic Role Labeling [8]. The common approach to train state-of-the-art embeddings such as Word2Vec [25], GloVe [36], and FastText [3] is to learn the vector representation for each individual word. Phrase representations are then constructed using compositional approaches of the unigram vectors [45,47,22]. However, the compositional approaches (e.g., sums and products of the word vectors) are often order-insensitive and fail to capture the semantic meaning of the phrase [28]. Unfortunately, in the biomedical domain, many key concepts are often expressed as multi-word phrases [20] and thus are critical for capturing lexical semantics. Furthermore, biomedical phrases may only be weakly compositional, or unlikely to be expressed only based on the meaning of its part. As motivating examples, the phrases ‘Glasgow Coma scale’, ‘open reading frame’, and ‘nuclear magnetic resonance’, may not be well-expressed as a composition of the individual words. Therefore, it is important to build a distributed representation that not only captures single words but multi-word phrases as well.

Learning a distributed phrase and word embeddings have been shown to be effective on a general, non-domain specific corpus [26]. Yet, one of the key challenges is to identify useful phrases. While this task is well-studied, many of the existing works require annotation or extensive computation to achieve good performance [4,10,35,37,44]. A new unsupervised method has been proposed to collect over 700,000 common phrases that may be useful for biomedical NLP from PubMed articles [20]. Unfortunately, including all possible phrases into the

* Corresponding author.
E-mail address: zgero@emory.edu (Z. Gero).

https://doi.org/10.1016/j.yjbinx.2019.100047

Available online 20 July 2019

embedding model significantly impacts the computational complexity and negatively impacts the learned representations.

We propose PMCVec, an unsupervised method that generates useful phrases from the corpus and builds a distributed representation that contains both single words and multi-word phrases by treating both as a single term (or unit). In this paper, we consider a phrase to be a continuous sequence of two or more words with no stopwords or punctuation marks except for a hyphen. We used a standard NLTK[1] stopword list. For example, our method obtains similar representations for the pairs 'hypertension' and 'high blood pressure' as well as 'myocardial infarction' and 'heart attack'. We introduce a new criterion to rank the generated phrases that balance phrase frequency, phrase length, and the frequency of the individual words within the phrase. This step allows us to select only the *k*-most useful phrases, where *k* is a hyperparameter that can be learned as well.

We compared our method against several existing embeddings: two general word embedding models and two biomedical domain word representations. Using five benchmark datasets for biomedical semantic similarity, we show that PMCVec achieves significant improvement over other models. We show that our distributed representation not only captures the semantic meaning of the phrases better than compositional methods, but it also does not significantly degrade the single-word representations.

This paper is organized as follows. First, we describe the various steps in the PMCVec process including preparing the text data; generating, ranking and filtering phrases; and learning the term embeddings. We then describe experimental results on several biomedical term-similarity evaluation datasets. We conclude with a discussion of how our method compares to other similar techniques and what can be done to improve further.

2. Methods

In this section, we present our framework for computing the distributed phrase representations. PMCVec consists of multiple steps: (1) preprocessing the articles, (2) generating phrases from the articles based on chunking, (3) ranking and filtering the phrases, and (4) tagging the phrases and building the distributed phrase representation. Fig. 1 depicts the entire workflow.

2.1. Preprocessing

We used titles and abstracts from all the 27 million documents in PubMed. The National Library of Medicine produces the citation records (in XML format) for PubMed [29]. The XML files are parsed to collect titles and abstracts. These are merged into a single large document. We then cleaned the document by removing terms that consisted only of numbers or special characters. For example, in the sentence "in 29 (69%) patients, the cancer cells showed a strong immunoreactivity for PCNA" the number 29 and (69%) would be removed.

2.2. Phrase generation

The next step in the process is to identify phrases from the corpus. Traditional techniques focus on identifying noun phrases since most meaningful phrases are of this form. These methods use predefined parts of speech (POS) rules or learn those rules from annotated documents to chunk the text [4,44,37]. However, such rule-based methods usually suffer in domain adaptation and will miss out on meaningful non-noun phrases including 'multilocus sequence typing', 'calcitonin gene related peptide', 'electrophoretic mobility shift assay', 'zollinger ellison syndrome', and 'diffusion tensor imaging'. Other generic phrase generation techniques leverage frequency statistics in document collections by extracting all possible n-grams from the text and retaining the most popular concepts [35,10]. However, this result enumerates all the possible n-grams and does not scale well for a large corpus. Instead, we use a conceptually simpler and more generic approach. Potential phrase boundaries are identified using stop words and punctuation [41]. Although this eliminates the possibility of stop words occurring in a phrase, it provides a more systematic methodology for generating variable n-gram phrases without having to specify ahead of time the maximum number of terms and enumerating all the possibilities. Thus, with the last example sentence in Fig. 1, the potential phrases from our chunking process are 'patients', 'cancer cells showed', and 'strong immunoreactivity', and 'PCNA'. Since our interest is to generate meaningful phrases, we remove any single word occurrences.

2.3. Rank and filter

The third step in our workflow is to rank and filter the potential phrases. This is a necessary step as there is no guarantee that all the phrases generated in the previous step are meaningful. Moreover, incorporating all the phrases impacts the learning process in terms of computational and memory complexity, and may degrade the distributed word representations. Thus, it is important to rank the phrases using a metric and filtering out those that do not meet certain criteria. Prior to ranking, we perform an initial filtering step that removes any phrases that do not appear sufficiently in the corpus. While we set the minimum corpus frequency to be 100, this number can be increased to further improve the speed of the ranking process. Thus, in our example in Fig. 1, 'paraffin-embedded bladder cancer section' did not occur frequently enough and was filtered out in this initial stage.

After the initial filtering step, we rank the multi-word phrases to identify meaningful phrases based on their likelihood to occur in PubMed literature as coherent units. Although there are several common phrase ranking criteria [6,17], we found they offered a poor trade-off between phrase frequency, constituent word frequency, and phrase length. Thus, we propose our own ranking criteria "Information Frequency (Info_Freq)" that provides a good balance. As an example, we filter out the phrase "cancer cells showed" in the filtering step of Fig. 1 since it has a low rank according to our criteria. Below, we describe Info_Freq and four of the commonly used phrase ranking metrics as well as discuss the benefits and limitations of each of them.

1. **Raw Frequency**: A measure of the number of times the phrase appears in the entire corpus. With the removal of stop words, most of the phrases that occur very frequently are likely to be good phrases. However, the simple nature of this metric punishes meaningful phrases that do not appear often and predominately favors 2-word phrases. Phrases like 'results suggest' and 'present study' which occur in most documents are ranked high but other important phrases like 'epithelial tissue' and 'acute respiratory failure' do not occur as frequently and subsequently have a low rank.
2. **Point-wise Mutual Information (PMI)[7]:** A measure of how much information is gained about a particular word if you also know the value of a neighboring word. It is defined as:

   $$\mathrm{PMI}(x, y) = \log\frac{p(x, y)}{p(x)p(y)},$$

   where $p(x)$ is the probability of the word x occurring in a document, and $p(x, y)$ is the probability of the co-occurrence of both words x, y occurring in the same document.
   For a three-word phrase, we adapt the above formula as:

   $$\mathrm{PMI}(x, y, z) = \log\frac{p(x, y, z)}{p(x)p(y)p(z)},$$

   PMI is often used to find good collocation pairs as high PMI occurs when the probability of the co-occurrence is either higher or slightly

[1] https://www.nltk.org/.

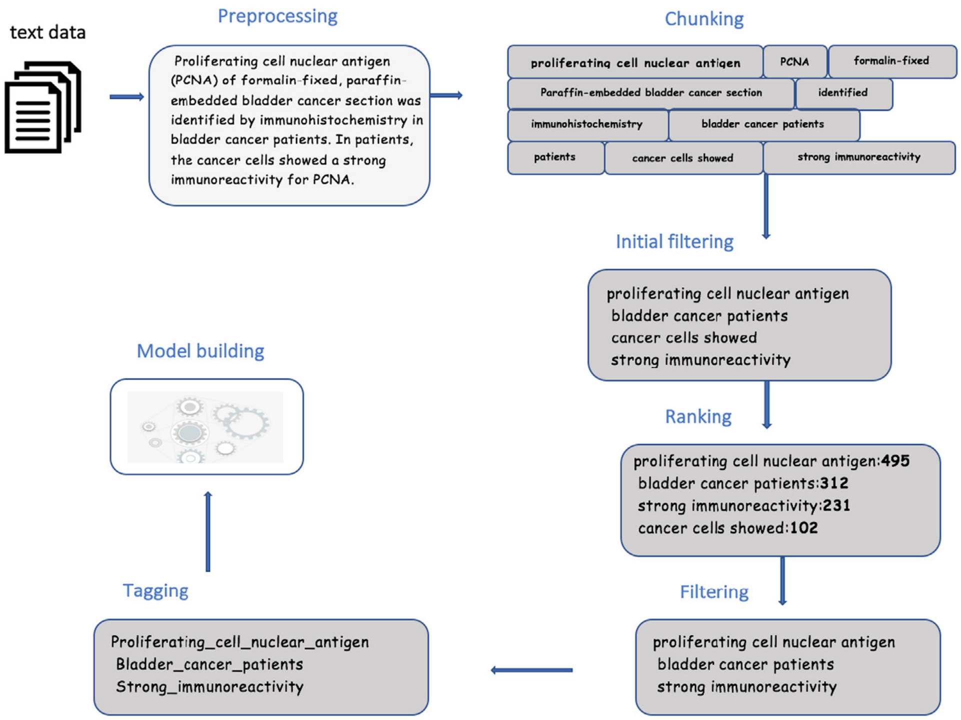

text data
Preprocessing
Proliferating cell nuclear antigen (PCNA) of formalin-fixed, paraffin-embedded bladder cancer section was identified by immunohistochemistry in bladder cancer patients. In patients, the cancer cells showed a strong immunoreactivity for PCNA.
Chunking
proliferating cell nuclear antigen
PCNA
formalin-fixed
Paraffin-embedded bladder cancer section
identified
immunohistochemistry
bladder cancer patients
patients
cancer cells showed
strong immunoreactivity
Initial filtering
proliferating cell nuclear antigen
bladder cancer patients
cancer cells showed
strong immunoreactivity
Ranking
proliferating cell nuclear antigen:495
bladder cancer patients:312
strong immunoreactivity:231
cancer cells showed:102
Filtering
proliferating cell nuclear antigen
bladder cancer patients
strong immunoreactivity
Tagging
Proliferating_cell_nuclear_antigen
Bladder_cancer_patients
Strong_immunoreactivity
Model building

Fig. 1. An illustration of PMCVec's workflow. The text data is preprocessed and chunked to obtain candidate phrases. The phrases are ranked using our proposed Information Frequency criteria, and then filtered. The resulting phrases are tagged to form a single unit and the tagged text is passed into a standard word embedding model. Each term is then represented using a dense vector that maintains semantic similarity and relatedness.

lower than the probabilities of the occurrence of each word. Conversely, phrases that contain frequently occurring words will have small PMI scores even if the phrase is good. As an example, ‘blood cells’ should be an important and meaningful phrase. Unfortunately, the constituent words ‘blood’ and ‘cells’ occur frequently in the corpus. As a result, the phrase is ranked very low.

3. **Jaccard's Coefficient (JC)** [40]**:** A measure of the similarity and diversity of the entire phrase set. It is defined as the frequency of a phrase divided by the total number of phrases that contain at least one term in the phrase:

$$\mathrm{JC}(x, y) = \frac{\mathrm{freq}(x, y)}{\mathrm{freq}(x, y) + \mathrm{freq}(y, *) + \mathrm{freq}(x, *)},$$

where $\mathrm{freq}(x, *)$ denotes the frequency of any phrase that contains the term x but not y. For a three-word phrase, we adapt JC as:

$$\mathrm{JC}(x, y, z) = \frac{\mathrm{freq}(x, y, z)}{\mathrm{freq}(x, y, z) + \mathrm{freq}(x, *) + \mathrm{freq}(y, *) + \mathrm{freq}(z, *)},$$

Although Jaccard index accounts for the diversity of the phrase, longer phrases are punished as there is a higher likelihood of at least one word appearing in a phrase. Thus, longer phrases like ‘reverse transcription polymerase chain reaction’ and ‘cervical squamous cell carcinoma’ will be ranked low even though they are meaningful phrases.

4. **Word2Phrase:** This is a method proposed by [26]. It is a data-driven approach where phrases are formed based on unigram and bigram counts.

$$\mathrm{Word2Phrase}(x, y) = \log\frac{\mathrm{count}(x, y) - \sigma}{\mathrm{count}(x)\mathrm{count}(y)}.$$

σ is used as a discounting coefficient to prevent too many phrases with infrequent words to be formed. This technique is applied in multiple passes to find longer phrases. For example, the phrase “blood cells” occurs 7000 times while “tagging snps” occurs only 350 times but the latter will have a higher score since the constituent words “tagging” and “snps” are infrequent compared to the more frequently occurring words “blood” and “cells” in the first phrase. The discounting coefficient takes off a constant number so that phrases with much less frequency but higher scores due to infrequent constituent words will be penalized more. We provide an empirical example in the supplementary file.

5. **Info_Freq:** Our proposed measure of the association between words in the phrase that accounts for the phrase frequency, the constituent words frequency, and the length of the phrase. For a two word phrase “x,y”, we calculate the info_freq as:

$$\mathrm{Info\_Freq}(x, y) = \log\frac{p(x, y)}{p(x)p(y)} * \log(\mathrm{freq}(x, y)).$$

For a three-word phrase, we adapt the above formula as:

$$\mathrm{Info\_Freq}(x, y, z) = \log\frac{p(x, y, z)}{Info\_Freq(x, y)p(z)} * \log(\mathrm{freq}(x, y, z)).$$

In the above equation, we assume the two-word-phrase (x,y) occurs more frequently than (y,z). Scores are calculated in increasing size of phrase length. All two-word-phrase scores will be calculated before any three-word phrases and so on. For instance, to calculate the info_freq of the phrase “high blood pressure”, we first calculate the score for the shorter phrase “blood pressure” and use this to get the score for the longer phrase. This is applied for phrases with more than three words as well. For the four-word phrase “chronic obstructive pulmonary disease”, we calculate the score for “pulmonary disease”, then for “obstructive pulmonary disease” and finally for “chronic obstructive pulmonary disease”. In the attached supplementary file, we provide detailed examples of how the scores are calculated for longer phrases.

Table 1 shows the top 10 phrases from all 27 million PubMed abstracts based on each of the five above criteria. Both the frequency and JC metrics only contain 2-word phrases. Moreover, the top-ranked phrases by frequency are not medically meaningful. PMI and Word2-Phrase are also biased towards short phrases mostly consisting 2 words. On the other hand, the top 10 phrases using Info_Freq contain a good mix of long and short phrases that are biomedical-relevant terms. We get 2-word, 3-word, 4-word and 5-word phrases using Info_Freq. Since our goal is to minimize the number of phrases to embed while keeping the most important ones, Info_Freq allows us to extract quality phrases

Table 1
The top 10 phrases from 27M PubMed abstracts using five different ranking criteria.

Frequency	PMI	JC	Info_Freq	Word2Phrase
present study	gemtuzumab ozogamicin	stainless steel	polymerase chain reaction	colorectal cancer
risk factor	erector spinae	myasthenia gravis	magnetic resonance imaging	waiting list
significant difference	oculocutaneous albinism	endoplasmic reticulum	vascular endothelial growth factor	virtual screening
cell line	hpv dna testing	anorexia nervosa	chronic obstructive pulmonary disease	tumor necrosis factor
results suggest	enterobius vermicularis	mycophenolate mofetil	coronary artery bypass graft	sodium nitroprusside
control group	cerebrotendinous xanthomatosis	rainbow trout	receiver operating characteristic curve	sensorineural hearing loss
amino acid	labrador retrievers	confidence interval	body mass index	pulmonary arterial hypertension
significantly high	polymyalgia rheumatica	neurofibrillary tangles	reverse transcription polymerase chain reaction	microscopic examination
significantly higher	lymphomatoid papulosis	lupus erythematosus	left anterior descending coronary artery	glucocorticoid receptor
risk factors	planum temporale	vena cava	amino acid	gastric bypass

with different number of words.

2.4. Tag and build embeddings

The final step in our workflow is to tag the selected phrases as a single term and then build the distributed word embeddings. The tagging process reformats the original phrase by joining the constituent words using the '_' symbol. This is to ensure the phrase is considered a single term (or unit) in the embedding process. For example, 'proliferating cell nuclear antigen' is tagged as 'proliferating_cell_nuclear_antigen' in the original corpus.

Once the tagging process is complete, we train a word embedding model on the entire tagged corpus. Under the word embedding model, terms are represented as dense vectors that capture the meaning of the words and retain the semantic and syntactic relationship between words. We use Word2Vec, the most widely used embedding method [27], which trains a shallow neural network to learn the word vectors. Word2Vec consists of two different architectures, the continuous bag of words (CBOW) and Skipgram. In CBOW, each word is trained using its surrounding context words – given this set of context words, what is the word that is most likely to appear? For example, in Fig. 2a, using the context of six words, what is the word that is most likely to appear between them? On the other hand, Skipgram (Fig. 2b) trains the context based on the target word – given the word, what are the other words that are likely to appear? We assessed the impact of the two different architectures (Fig. 2) on the quality of the resulting embeddings. We used an existing work to guide the hyperparameter searches for CBOW and Skipgram to achieve optimal performances on both architectures [5]. While our framework can leverage other word embedding models such as Glove [36] and FastText [3], we achieved the best performance with the Word2Vec model.

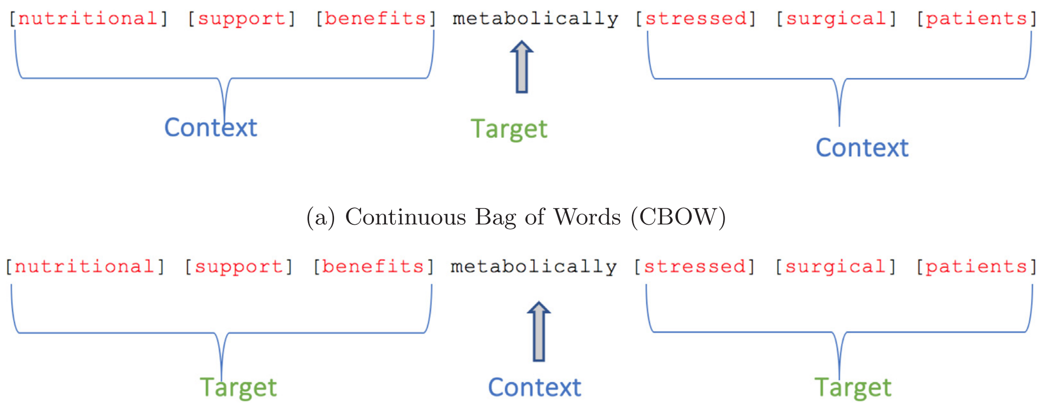
We assessed our model on five different evaluation datasets and performed several experiments to study the impact of the number of phrases, embedding architecture, and phrase generation. We also evaluated PMCVec with several other publicly available word embeddings.

2.5. Evaluation datasets

We evaluated the performance of the final models on five popular medical term similarity and relatedness datasets.

- **Mayo [34]:** This dataset consists of a total of 101 UMLS concept pairs (202 terms): 113 are unigrams, 73 are 2-grams, and 16 are 3-gram or more. 13 medical coding experts rated these 101 pairs for semantic relatedness on an ordinal scale. The relatedness of each term pair was assessed based on a four-point scale: (4.0) practically synonymous, (3.0) related, (2.0) marginally related and (1.0) unrelated.
- **miniMayo:** This is a subset of the 'Mayo' dataset and consists of 30 term pairs on which a higher inter-annotator agreement was achieved. Out of a total of 60 term pairs, 31 are unigrams, 22 are 2-grams and 7 are 3-gram or more.
- **AH [16]:** This is a set of 36 medical concepts extracted from the MeSH repository by Hliaoutakis. The similarity between word pairs was assessed by 8 medical experts. This dataset contains 41 unigram terms, 20 2-gram terms, and 11 terms which are 3-gram or more.
- **UMNSRS [32]:** This is a dataset of 566 UMLS concept pairs that have been ranked by eight medical residents for similarity on a continuous scale. All the 1132 terms are unigrams.
- **UMNSRS_R [32]:** This is a dataset of 587 UMLS concept pairs that have been ranked by eight medical residents for relatedness. All of the 1174 terms are unigrams.

Two of the datasets (UMNSRS and UMNSRS_R) consist of only single-word term pairs only. The other three (Mayo, miniMayo, and AH)


[nutritional] [support] [benefits] metabolically [stressed] [surgical] [patients]
Context
Target
Context
[nutritional] [support] [benefits] metabolically [stressed] [surgical] [patients]
Target
Context
Target

(a) Continuous Bag of Words (CBOW)

(b) Skipgram

Fig. 2. The two different Word2Vec architectures.

contains both single and multi-word term pairs.

2.5.1. Evaluation metric

The comparison on the semantic similarity and relatedness datasets is based on the Spearman rank order correlation coefficient (ρ). The coefficient is computed by comparing the ranking from the model ($\widehat{r_i}$) to the expert judged ranking (r_i):

$$\rho = 1 - \frac{6\sum_i r_i - \widehat{r_i}}{n(n^2-1)}.$$

Since the benchmark models only support single words, we use a compositional approach of vector averaging wherever there are multi-word similarity comparisons. For instance, when comparing the semantic similarity of the two phrases “Kidney Failure” and “Renal Failure”, our model represents both terms as single entities and learns a vector representation for each phrase. The baseline models, however, learn embeddings for each word in the phrase and average those vectors to represent the phrase.

2.6. Impact of phrase generation

Our first experiment assesses the impact of our phrase generation step. A qualitative comparison can be seen in Table 1, which contains the top phrases generated by different phrase generation criteria. In this section, we quantify the performance of the metrics on the evaluation datasets. Table 2 shows the comparative scores based on similarity and relatedness for each metric, with the word2vec hyperparameters selected that achieved the highest score with 18,000 phrases used as this gave the best performance across the board. The full table with exhaustive parameters is attached in the supplementary file for further comparison. We see that Info_freq gets the best scores in the three mixed datasets (both single and multi-word phrases) and performs similarly in the single word datasets too. Moreover, Info_Freq is robust across a wide range of hyperparameter settings for the embedding models.

We also compared the quality of our phrases to PubMed Phrases, a collection of common phrases that were generated for biomedical NLP [20]. Each phrase comes with a precalculated score based on the p value of the hypergeometric test the authors performed on segments of consecutive terms that are likely to appear together in PubMed. To compare the phrase generation method, we tagged the PubMed Phrases in the PubMed abstracts and re-trained a new CBOW model. Longest phrases are tagged first to avoid conflict with substring phrases. Any substring phrases of longer phrases will be tagged only if they appear as stand-alone not as sub-phrase of longer phrase. Fig. 3 shows the average similarity scores using all five test datasets using the PubMed phrases [20] and PMCVec. We include two models for PubMed Phrases, the first is using the top n phrases as scored by the authors and the second (exist in chunk) is also using the authors scores but only tagging phrases if the phrase exists in our preprocessed chunks. The PMCVec-based models consistently outperform the PubMed phrases at all the ranges of phrases. This showcases the effectiveness of our phrase generation technique.

Table 2
Top similarity scores for each phrase selection metric.

Metric	AH	miniMayo	Mayo	UMNSRS	UMNSRS_R
JC	0.59	0.78	0.61	0.6	0.54
pmi	0.62	**0.81**	0.6	**0.62**	**0.55**
Word2Phrase	0.63	0.79	0.55	0.6	**0.55**
Info_Freq	**0.7**	**0.81**	**0.66**	0.59	**0.55**

The best scores for each evaluation dataset are shown in bold.

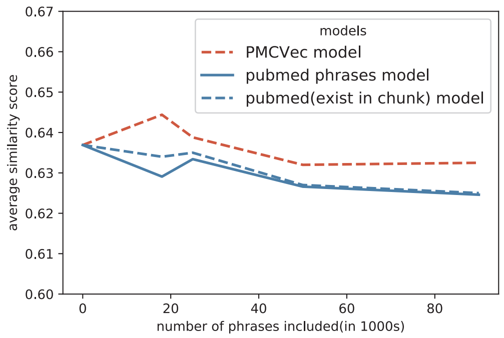


Fig. 3. Comparison with PubMed Phrases.

2.7. Impact of number of phrases and embedding techniques

Our second experiment assesses the quality of the PMCVec-embeddings based on the number of tagged phrases and the two Word2Vec architectures. Fig. 4a depicts how the number of phrases affects the quality of the learned model with respect to the five test datasets (CBOW model is used). For the two datasets (UMNSRS and UMNSRS_R) with only single word pairs, the quality of the embedding monotonically decreases as we include more phrases. As more phrases are tagged, fewer unigrams are available to learn the word embeddings. For the combined test sets (miniMayo, mayo and AH), the quality of the embeddings increases and then decreases or stalls thereafter. Thus, for optimal performance we need to cap the number of phrases so that our model learns quality vectors both for single and multi-word terms.

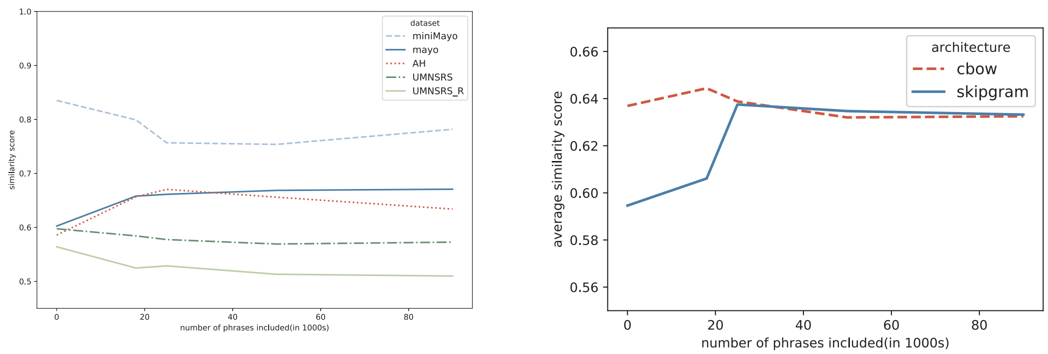
We also assessed the quality of the word vectors using the two different Word2Vec architectures. Fig. 4b shows the average similarity scores on all five datasets for both the CBOW and Skipgram architecture. CBOW is better when there are fewer phrases. As the number of phrases increases, the Skipgram model slightly outperforms CBOW. Based on the figure, the best performance is achieved by CBOW using 18 K tagged phrases. The hyperparameters associated with this model are a negative sample size of 10, sub-sampling of 1e−5, a minimum count of 1, vector dimension of 200, context window size of 10, and a learning rate of 0.025.

3. Results

3.1. Baseline methods comparison

We benchmarked PMCVec with four other word-embedding models, all pre-trained on different corpora. For our model, we used hyperparameters associated with the best performance as described above.

- **Google news** [15] **:** A Word2Vec model that is trained on a general non-biomedical corpus. This is widely used as state-of-the-art embedding model as it is trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million terms.
- **Glove** [14] **:** A GloVE model that is trained on a general non-biomedical corpus. Training is performed on aggregated global word-word co-occurrence statistics from a corpus of Wikipedia and Gigaword 5 (6 Billion tokens). It is a 300-dimensional vector representation for 400k words.
- **BioNLP** [30] **:** A Word2Vec model that is trained on 22,723,473 PubMed abstracts and titles as well as the full-text in 672,589 PubMed Central Open-Access articles. It is a 200-dimensional vector representation for over 3 million words.
- **BioASQ** [2] **:** A Word2Vec model that is trained on a corpus of 10,876,004 English abstracts of biomedical articles from PubMed. The resulting model is 200-dimensional vector representation of



(a) Number of Phrases

(b) Word2Vec Architectures

Fig. 4. The similarity score as a function of the number of tagged phrases and the Word2Vec model architectures.

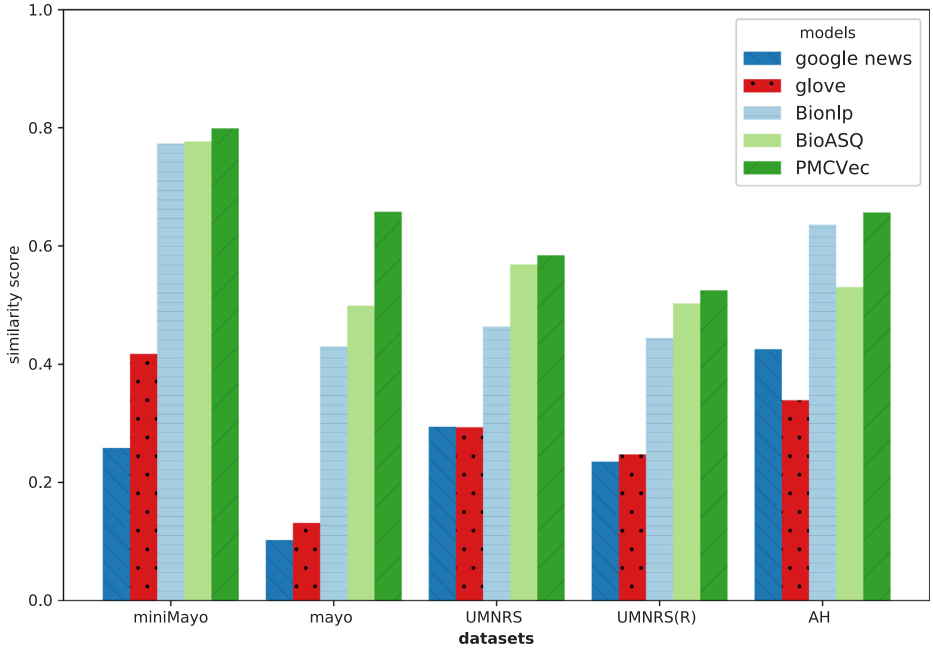


Fig. 5. Comparison of the baseline methods and PMCVec on the five datasets.

1,701,632 distinct words.

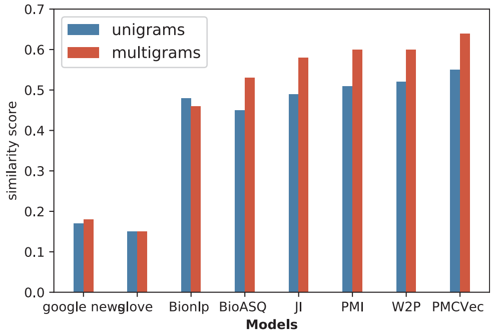
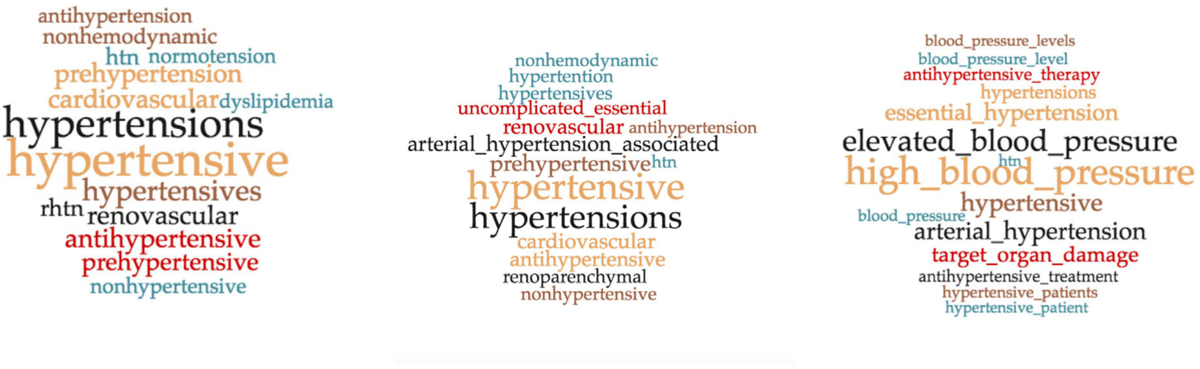


Fig. 6. Comparison of the baseline methods and PMCVec on unigrams and multigrams extracted from the test datasets.

The performance of PMCVec and the baseline models on the five datasets is shown in Fig. 5. The two models trained on general corpora (Google news and Glove) have the lowest scores on all the datasets. On the contrary, the other two baseline models trained on biomedical corpora perform significantly better. This is consistent with prior results outlining the importance of the training corpus [31]. PMCVec outperforms the baseline models on all the datasets. The improvement is noticeable in the Mayo dataset, where the task is harder due to the lower inter-annotator agreement. We also note that our model performs better on both of the single-word pair datasets (UMNRS and UMNRS_R), which shows that incorporating phrases into the embedding process does not significantly compromise the quality of the single word vectors.

To quantify the performance of PMCVec on the single words and multi-words separately, we extract unigrams from the “AH” and “Mayo” datasets. Since “MiniMayo” is a subset of the “Mayo” dataset, all terms are already included in the extracted set. The remaining two datasets (UMNSRS and UMNSRS_R) are all single words and the performance of the models on these datasets are shown in Fig. 5. We depict how all the aforementioned models compare when using only unigrams and multi-word phrases in Fig. 6. We observe that the performance gain from PMCVec is noticeable for both single words and multi-words compared to the baseline methods.

The inclusion of multi-word phrases not only improves the semantic similarity performance but is also qualitatively better. Fig. 7 shows the cluster of terms that are semantically similar to the word ‘hypertension’. In the two scenarios where no phrases are tagged (Fig. 7a) and the PubMed phrases are tagged (Fig. 7b), the closest terms to hypertension



(a) No Phrases (b) PubMed Phrases (c) PMCVec

Fig. 7. Word cloud for semantically similar terms to ‘hypertension’. The size of the term is proportional to how semantically close it is to the word ‘hypertension’ with the largest denoting the most similar.

are the same which are ‘hypertensive’ and ‘hypertensions’ and the third closest are ‘hypertensives’ and ‘prehypertensive’ respectively. Moreover, only two multi-word phrases (‘arterial hypertension associated’ and ‘uncomplicated essential renovascular’) appear when using the PubMed phrases. Using PMCVec, ‘high blood pressure’, ‘elevated blood pressure’, and ‘essential hypertension’ are the closest and all three are semantically similar to hypertension.

Additional examples of similar terms are shown in Table 3 for different disorders, symptoms, and medications. In all 6 cases, PMCVec is able to return relevant multi-word synonyms in the top 5 closest words. ’diabetes mellitus’ is a semantically similar to ’diabetes’ whereas the other two methods contain the top word ‘mellitus’. Similarly for symptoms, ‘joint pains’ is returned for aches whereas the other two embeddings do not have this term. The same holds true for drugs; for ‘aspirin’, single words returns ‘clopidogrel’ and PubMed phrases gets ‘dipyridamoleasprin’ as the most sematically similar term. These are drugs commonly administered with aspirin. With PMCVec, the top term is ‘acetylsalicylic acid’ which is another name for aspirin. In general, the PMCVec-based embeddings produce more accurate vector representations for phrases. Biomedical text is rich with multi-word concepts and terminologies, and as such representing these terms appropriately as single units to learn their vector representations is an important step in biomedical text processing.

3.2. Limitations

Our model focused on obtaining a good distributed term representation by combining multi-word phrases and single-words. Unfortunately, training GloVe and FastText models took considerably more time to train in large dimensions. Due to computational time and

Table 3
Qualitative results with and without phrases in embeddings.

		Single words	PubMed phrases	PMCVec
disorders	**diabetes**	mellitus	mellitus	diabetes_mellitus
		prediabetes	niddm	tdm
		diabetic	diabetic	diabetes_patients
		tdm	tdm	mellitus
		prediabetic	prediabetes	prediabetes
		arrhythmias	arrhythmias	supraventricular_arrhythmia
	arrythmia	tachyarrhythmias	tachyarrhythmias	arrhythmia
		tachyarrhythmia	tachyarrhythmia	tachydysrhythmias
		tachycardia	tachycardiafibrillation	supraventricular_tachyarrhythmia
		arrhythmic	tachycardia	cardiac_arrhythmias
		croupy	tussive	coughing
symptoms	**cough**	coughing	persistent_cough	chronic_cough
		tussigenic	coughs	expectoration
		coughers	tussigenic	rhonchi
		coughs	sneez	tussigenic
		ratbungarus	pains	aching
	aches	bungarusrat	aching	joint_pains
		pains	thighness	pains
		backache	ratbungarus	jointbody
		acetylcholinesterases	bungarusrat	lassitude
		clopidogrel	dipyridamoleaspirin	acetylsalicylic_acid
drugs	**asprin**	aspirins	platet	indobufen
		antiplatelet	esomeprazoles	clopidogrel
		acetylsalicylic	heparinwarfarin	aspirins
		aspirindipyridamole	giibiiia	antiplatelet_drugs
		amoxycillin	amoxycillin	amoxycillin
	amoxicillin	phenoxymethylpenicillin	amoxicillinclavulanate	amoxicillinclavulanic_acid
		amoxicillinclavulanate	augmentin	amoxicillinclavulanate
		cefaclor	cefaclor	phenoxymethylpenicillin
		augmentin	phenoxymethylpenicillin	augmentin

memory limitations, we were not able to train these models with large dimensions and window sizes. The GloVe and FastText models we trained performed much worse than the other two Word2Vec models in smaller dimensions (100-dimension and 200-dimension results are in the supplementary table) which is consistent with the work of Fan et al. [11] on clinical notes.

The method we used for phrase generation did not consider terms and phrases containing only digits or stop words. Even though it is common to remove stop words in the form of subsampling for word embedding generation since they occur much more frequently and inflate the vocabulary size and training time [26], it may not be desired for biomedical phrase generation. We believe that this may result in the exclusion of meaningful phrases. However, incorporating these aspects into the phrase generation process would significantly lengthen the computation time. We plan to experiment in the future to determine the viability of including phrases with digits and stop words.

4. Conclusion

Learning quality vector embeddings that incorporate both single word and multi-word phrases can be quite challenging. Although compositional approaches to combine unigram vectors to obtain a phrase representation has worked well in some domains, this does not capture the meaning of key biomedical concepts. Moreover, incorporating all the existing identified biomedical phrase can negatively impact the quality of the embeddings. To address these issues, we introduced PMCVec, an unsupervised method that bridges the gap in learning quality vector embeddings for multi-word phrases which are a staple in biomedical literature. Our method not only generates useful phrases from the corpus, but it also introduces a new criterion to rank the generated phrases to avoid incorporating all the phrases and achieve a better embedding for both single words and multi-word phrases. We showed that the learned phrase embeddings result in better performance than compositional approaches using several benchmark datasets. As an example, a search result for the term ‘colitis’ should include multi-word expressions like ‘inflammatory bowel disease’. The learning of vectors for both these terms allows easy association of the concepts, which are very similar terms but will not be learned as such with just single-word embeddings. We believe that PMCVec-learned representations will be widely useful for a variety of biomedical NLP tasks.

Data availability

The PubMed dataset used in this study is publicly available for download at https://www.nlm.nih.gov/databases/download/pubmed_medline.html. The resources we used and the final model are available for download at https://github.com/ZelalemGero/PMCvec.

Contributions

Both authors conceived the study and contributed to the design of the proposed model. Z.G. implemented algorithms and conducted the experiments. Z.G. and J.H. performed analysis on experimental results and wrote the manuscript. J.H. provided insightful discussions, reviewed the results and revised the manuscript.

Declaration of Competing Interest

The authors declare no competing interest.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.yjbinx.2019.100047.

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, A neural probabilistic language model, J. Mach. Learn. Res., 0(Feb):1137–1155, 2003.

[2] Bioasq releases continuous space word vectors obtained by applying word2vec to pubmed abstracts. < http://bioasq.lip6.fr/tools/BioASQword2vec/ >.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov, Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

[4] Kuang-hua Chen, Hsin-Hsi Chen, Extracting noun phrases from large-scale texts: a hybrid approach and its automatic evaluation, Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 1994, pp. 234–241.

[5] Billy Chiu, Gamal Crichton, Anna Korhonen, Sampo Pyysalo. How to train good word embeddings for Biomedical NLP, in: Proceedings of the 15th Workshop on Biomedical Natural Language Processing, 2016, pp. 166–174.

[6] Young, Mee ChungJae YunLee, A corpus-based approach to comparative evaluation of statistical term association measures, J. Am. Soc. Inform. Sci. Technol. 52 (4) (2001) 283–296.

[7] Kenneth Ward Church, Patrick Hanks, Word association norms, mutual information, and lexicography, Comput. Linguist. 16 (1) (1990) 22–29.

[8] Ronan Collobert, Jason Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.

[9] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. (2011) 2493–2537.

[10] Paul Deane, A nonparametric method for extraction of candidate phrasal terms, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 605–613.

[11] Yadan Fan, Serguei Pakhomov, Reed McEwan, Wendi Zhao, Elizabeth Lindemann, Rui Zhang, Using word embeddings to expand terminology of dietary supplements on clinical notes, JAMIA Open (2019).

[12] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, Gareth J.F. Jones, Word embedding based generalized language model for information retrieval, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 795–798.

[13] Xavier Glorot, Antoine Bordes, Yoshua Bengio, Domain adaptation for large-scale sentiment classification: a deep learning approach, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 513–520.

[14] Glove: Global vectors for word representation. < https://nlp.stanford.edu/projects/glove/ >.

[15] word2vec: Tool for computing continuous distributed representations of words. < https://code.google.com/archive/p/word2vec/ >.

[16] Angelos Hliaoutakis, Semantic similarity measures in mesh ontology and their application to information retrieval on medline. Master’s thesis, 2005.

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

[18] Abhyuday Jagannatha, Jinying Chen, Yu Hong, Mining and ranking biomedical synonym candidates from wikipedia, Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis, 2015, pp. 142–151.

[19] Zhenchao Jiang, Liuke Jin, Lishuang Li, Meiyue Qin, Qu Chen, Jieqiong Zheng, Degen Huang, A crd-wel system for chemical-disease relations extraction, Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, 2015, pp. 317–326.

[20] Sun Kim, Lana Yeganova, Donald C. Comeau, W. John Wilbur, Zhiyong Lu, Pubmed phrases, an open set of coherent phrases for searching biomedical literature, Scient. Data (2018) 180104.

[21] Aris Kosmopoulous, Ion Androutsopoulos, Georgios Paliouras, Biomedical semantic indexing using dense word vectors in BioASQ, J. Biomed. Semant. (2015).

[22] Rémi Lebret, Ronan Collobert, The sum of its parts: Joint learning of word and phrase representations with autoencoders. arXiv preprint arXiv:1506.05703, 2015.

[23] Shengyu Liu, Buzhou Tang, Qingcai Chen, Xiaolong Wang, Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries, Information (4) (2015) 848–865.

[24] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, Christopher Potts, Learning Word Vectors for Sentiment Analysis, vol. 1, Association for Computational Linguistics, 2011, pp. 142–150.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[27] Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig, Linguistic regularities in continuous space word representations, in: HLT-NAACL, 2013, pp. 746–751.

[28] Jeff Mitchell, Mirella Lapata, Composition in distributional models of semantics, Cognit. Sci. (8) (2010) 1388–1429.

[29] National Library of Medicine. Download medline/pubmed data. < https://www.nlm.nih.gov/databases/download/pubmed_medline.html >.

[30] Biomedical natural language processing: Tools and resources. < http://bio.nlplab.org/ >.

[31] Farhad Nooralahzadeh, Lilja vrelid, Jan Tore Lnning, Evaluation of Domain-specific Word Embeddings using Knowledge Resources, in: Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélne Mazo, Asuncion

Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May 7–12, 2018 2018. European Language Resources Association (ELRA).

[32] Serguei Pakhomov, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, Genevieve B. Melton, Semantic similarity and relatedness between clinical terms: an experimental study, in: AMIA Annual Symposium Proceedings, American Medical Informatics Association, 2010, vol. 2010, p. 572.

[33] Serguei V.S. Pakhomov, Greg Finley, Reed McEwan, Yan Wang, Genevieve B. Melton, Corpus domain effects on distributional semantic modeling of medical terms, Bioinformatics 32 (23) (2016) 3635–3644.

[34] Serguei V.S. Pakhomov, Ted Pedersen, Bridget McInnes, Genevieve B. Melton, Alexander Ruggieri, Christopher G. Chute, Towards a framework for developing semantic relatedness reference standards, J. Biomed. Inform. (2) (2011) 251–265.

[35] Aditya Parameswaran, Hector Garcia-Molina, Anand Rajaraman, Towards the web of concepts: extracting concepts from large datasets, Proceedings of the VLDB Endowment, 2010, pp. 566–577.

[36] Jeffrey Pennington, Richard Socher, Christopher Manning, Glove: global vectors for word representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[37] Vasin Punyakanok, Dan Roth, The use of classifiers in sequential inference, in: Advances in Neural Information Processing Systems, 2001, pp. 995–1001.

[38] Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, Sophia Ananiadou, Distributional semantics resources for biomedical text processing, Proceedings of the 5th International Symposium on Languages in Biology and Medicine, 2013.

[39] Alexander M. Rush, Sumit Chopra, Jason Weston, A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685, 2015.

[40] Gerard Salton, Michael J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Inc, 1986.

[41] Kamal Sarkar, A hybrid approach to extract keyphrases from medical documents. arXiv preprint arXiv:1303.1441, 2013.

[42] Buzhou Tang, Hongxin Cao, Xiaolong Wang, Qingcai Chen, Xu Hua, Evaluating word representation features in biomedical named entity recognition tasks, BioMed Res. Int. (2014).

[43] Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, Hongfang Liu, A comparison of word embeddings for the biomedical natural language processing, J. Biomed. Inform. 87 (2018) 12–20.

[44] Endong Xun, Changning Huang, Ming Zhou, A unified statistical model for the identification of english basenp, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2000, pp. 109–116.

[45] Wenpeng Yin, Hinrich Schütze, An exploration of embeddings for generalized phrases, in: Proceedings of the ACL 2014 Student Research Workshop, 2014, pp. 41–47.

[46] Dani Yogatama, Fei Liu, Noah A. Smith, Extractive summarization by maximizing semantic volume, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1961–1966.

[47] Mo, YuMarkDredze, Learning composition models for phrase embeddings, Trans. Assoc. Comput. Linguist. (2015) 227–242.