2020

Iteratively Linking Words Using Word2Vec and Cosine Similarity

Kevin Ros

Iteratively Linking Words Using Word2Vec and Cosine Similarity

Kevin Ros

5 May 2020

A thesis submitted in partial fulfillment of the requirements for the degree of
Bachelor of Arts
in
Computer Science
at
Vassar College

Advisor: Dr. Nancy Ide
Second Reader: Dr. Jonathan Gordon

Acknowledgements

I would like to thank my advisor, Dr. Nancy Ide, for her guidance, support, and feedback throughout this process. Additionally, I would like to thank my second reader, Dr. Jonathan Gordon, for his helpful comments and feedback. Without both of them, this thesis would not have been possible.

I would also like to thank Charlotte Lambert and my parents for providing countless grammatical, structural, and stylistic suggestions. With their help, my arguments, figures, and equations became much clearer.

Abstract

We propose an algorithm that constructs relationships among any number of seed words. A relationship consists of a set of iteratively-generated paths of similar words, where each path links one seed word to another. The similar words are generated using Word2Vec word embeddings and the cosine similarity measure. By examining the effectiveness of the proposed algorithm in the mental health domain, we find that the algorithm effectively returns meaningful relationships and has the potential to be used for hypothesis generation and information extraction.

1 Introduction

The amount of information published in academic journals is increasing at a breathtaking rate. In 2018 alone, over three million research articles were published in English-speaking journals [21], and shows no sign of slowing down. Not only is it impossible to keep up with every result, but it is becoming increasingly difficult to comprehend the complex interplay and ever-changing relationships among many of the words that constitute the results. Thus, many have turned to extracting relationships among words and concepts from previously published literature with the hope of verifying known relationships, discovering new relationships, and generating hypotheses for further inquiry.

This paper proposes a novel, unsupervised, and domain-independent algorithm for building relationships among any number of given seed words. These relationships are constructed by creating a set of links among seed words, where each link consists of a path of related words from one seed word to another. Links can be thought of as a lexical chain (a sequence of related words or concepts) between two words. The algorithm builds on previous results by using word embeddings generated via Word2Vec [27, 23] and the cosine similarity measure to discover related words used in the construction of the links between the given words. The general idea of the algorithm was inspired by the common literature-based discovery technique of closed discovery [37, 10]. The resulting links and relationships, although interesting in their own right, can be used for hypothesis generation, information extraction, and general guidance for further inquiry.

The proposed algorithm can be applied to numerous domains given a sufficient word embedding space. In this paper, we decided to investigate the algorithm's effectiveness within the context of mental health, looking at relationships between terms such as "anxiety", "depression", and many others via embeddings extracted from the PubMed database [23]. We chose this field partly due to the availability of published articles and embedding models, but mainly due to its impact and importance. A better understanding of an individual's mental health and the relationships between illnesses, treatments, and explanations may lead to a happier, healthier, and safer society as a whole.

This paper is organized as follows. In Section 2, we review relevant literature and highlight the differences between this paper and past approaches. Then, we explore the existing techniques

of literature-based discovery and word embeddings in Sections 3 and 4 respectively, and describe how each topic is relevant to the discovery algorithm described in this paper. In Section 5, we describe the proposed algorithm along with the pseudocode for both discovering and analyzing relationships. We present interesting discovered relationships in Section 6. Finally, we discuss short-comings, extensions, and future work in Section 7, and we conclude with Section 8.

2 Literature Review

In this section, we review relevant literature. Beginning with word embedding techniques, we examine their importance to information extraction in the biomedical domain. Next, we review literature-based discovery and its relation to word embeddings, followed by a brief review of lexical chains and their use in relationship discovery.

2.1 Word Embeddings

There are many techniques for perserving the semantic relationships between words or concepts, such as Latent Semantic Analysis [12], Latent Dirichlet Allocation [4], and more recently, various neural network approaches. One such unsupervised neural network approach, Word2Vec, is a family of model architectures which generate word embeddings from large, unstructured corpora using a shallow neural network [27]. Today, many state-of-the-art results involving word embeddings have been generated via deep learning [5, 28].

In the biomedical domain, word embeddings have been widely used to extract information from text [9, 19, 25, 26, 41]. In the field of psychology, word embeddings have been used to explore word associations in dreams [1], to detect anxiety [33, 35], and to generally monitor mental health [3, 14].

2.2 Literature-Based Discovery

Literature-based discovery was originally developed by Dr. Don Swanson, who proposed a method of extracting undiscovered knowledge and relationships from existing literature [38]. Swanson discovered links between various topics in published literature, such as one between fish oil and Raynaud's syndrome [37], and another between migraines and magnesium [39]. Today, literature-based discovery is widely used in a multitude of fields [15], including technology [16, 24], counter-terrorism [17, 20], and biomedicine [7]. Within the biomedical domain, literature-based discovery has been used to investigate drug development [13, 44], drug repurposing [29, 43, 44], and many other non-explicit relationships.

One framework for literature-based discovery involves using distributional models, which often involve constructing word embeddings [10]. Combining both word embeddings and literature-based discovery have led to substantial results. For example, Latent Semantic Indexing has been shown to improve the effectiveness of literature-based discovery [8]. Additionally, Kibwami and Tutesigensi used the cosine similarity score of TF-IDF category vectors as a basis for literature-based discovery regarding environmental research [22]. Rather, et al., showed that Word2Vec and literature-based discovery can effectively verify and uncover relationships regarding biomedical concepts [30]. The authors, however, only looked at similar words for a single given word. Word embeddings and literature-based discovery have also been used to find "potentially new multimorbidity patterns of psychiatric and somatic diseases" [42]. Pertaining to mental health, Hu and Terrazas used Word2Vec and literature-based discovery to develop a proof-of-concept computer system which leverages extracted knowledge to make recommendations [14].

2.3 Lexical Chains

A lexical chain consists of a sequence of related words or concepts. Lexical chains have long been used in text summarization [2], especially within the biomedical domain [31, 32]. Srinivasan proposed a closed-discovery algorithm (among others) that links two concepts with the help of MeSH profiles [36]. Jha and Jin used a graph-based approach to "glean" across multiple documents to discover relationships between a pair of words, and they evaluated the strength of the discovered relationships via the Kulczynski correlation measure [17]. The authors only discussed relationships between two concepts, and the concepts themselves require prior extraction, along with the construction of a knowledge base to store the relations between concepts. Jiang and Zhai proposed a probabilistic approach based on random walks of word adjacency graphs for discovering meaningful relations between words [18]. We were unable to find any existing approach that used word embeddings along with the cosine similarity score to iteratively generate links between multiple words.

3 Literature-Based Discovery

Literature-based discovery is a form of knowledge extraction from existing literature with the goal of uncovering previously unknown relationships or verifying known relationships. As discussed in Section 2.2, the effectiveness of literature-based discovery to uncover and verify relationships has been demonstrated by its widespread use, both on a single domain and across multiple domains. Regardless of the number of domains, many literature-based discovery models are built upon Swanson and Smalheiser's proposed ABC co-occurrence model [40]. Within the ABC co-occurrence model, there are methods of closed discovery and open discovery [10].

The goal of closed discovery is to explain correlations [10]. Given a start term and an end term, closed discovery attempts to find middle term(s) which link the start term to the end term. Both the start term and end term of the discovery process must be known beforehand. For example, consider two known terms A and C. Additionally, suppose it is discovered that term A is related to term(s) B_i and term(s) B_i is related to term C. Hence, a conclusion may be drawn that term A is related term C. As shown in Figure 1, there can be any number of B_i terms.

On the other hand, the goal of open discovery is to generate new relationships [10]. After beginning with a known start term, related terms are generated until a desired number relationships are found. For example, given any start term A, a list of related terms $B_1, B_2, \ldots B_n$ are generated. Then, from each B_i, a list of $C_{i,1}, C_{i,2}, \ldots C_{i,m}$ are generated. This process is depicted in Figure 2.

Regardless of the discovery approach, there are various methods for finding the intermediate (and end, in the case of open discovery) terms. Three main approaches include co-occurrence models, semantic models, and distributional models [10]. Within a co-occurrence model, relationships are discovered by looking at terms that co-occur with each other. Similarly, a semantic model finds terms that are semantically similar. In both cases, terms are represented as the words themselves. With distributional models, however, the terms are represented as context vectors (generated, for example, by Word2Vec). Here, the relationships are realized through operations in the vector space, such as cosine similarity.

As mentioned in Section 1, we focus on relationships between terms pertaining to mental health that exists within the biomedical domain. Because an open discovery distributional model has no set end term(s), the number of possible relationships could quickly become unmanageable without domain-specific knowledge. Thus, we decided to focus only on a closed discovery distributional model. However, our model differs slightly from the closed discovery model described above by allowing for any number of initial words, where all are treated as both start and end terms. Addi-

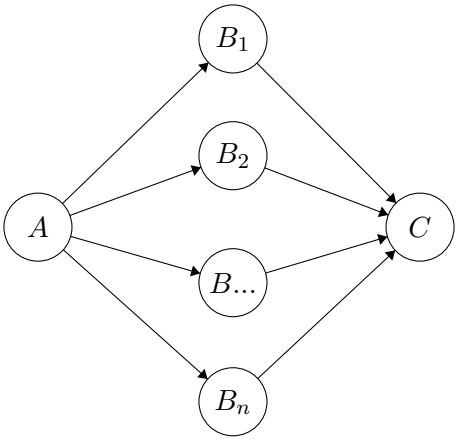

Figure 1: Closed Discovery

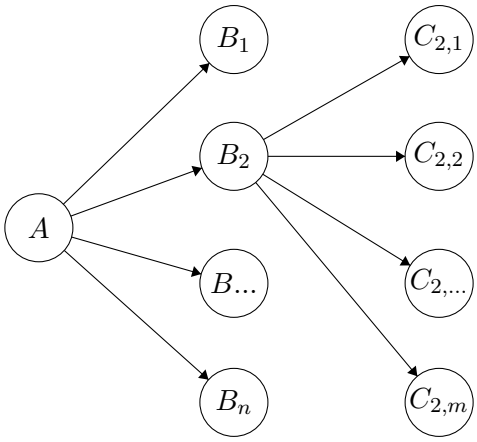

Figure 2: Open Discovery

tionally, we allow for longer paths (more intermediate terms) between the initial terms. The path length is only limited by the number of iterations parameter of our algorithm. This process is formally described in Section 5.

4 Word Embeddings

Embedding words as vectors, when done in a way to preserve semantic relationships between words, allows for more meaningful vector manipulation and more effective relationship discovery. Thus, it is imperative that this process be as efficient and as independent as possible, so that it can be applied to any collection of text, no matter the domain or size. Although there are numerous methods for embedding words as vectors, recent state-of-the-art results have come from machine learning. Here, we leverage embeddings generated via Word2Vec, which was one of the pioneering unsupervised word embedding-generation techniques. In Section 4.1, we discuss the Word2Vec architectures and the structure of the resulting embeddings. Then, in Section 4.2, we explain how Word2Vec was used to generate the PubMed embeddings, that are used in this paper.

4.1 Background on Word2Vec

Word2Vec is a family of model architectures which generate word embeddings from large, unstructured corpora using a shallow, unsupervised neural network [27]. There are two main architectures: Continuous Bag-of-Words (CBOW) and skip-gram. With CBOW, the model predicts a word given a surrounding text. The skip-gram model, on the other hand, predicts the surrounding words of a given word. Because Word2Vec is completely unsupervised, no labeled data is needed during the word embedding generation. Instead, unstructured corpora (such as raw text) are given to the model(s) which, through training, results in word vectors. The resulting vectors are typically between 50 and 300 dimensions, and given sufficient training data, capture the semantic relationships between the words in the original text.

Usually, semantic relationships between words are captured through a similarity measure between their respective vectors. One such measure is cosine similarity, which leverages the cosine of the angle between two vectors. In theory, smaller angles between word vectors indicate a higher similarity between the respective words. Put differently, words with small vector-angle differences have been used in similar contexts throughout the training data. The cosine similarity measures ranges from -1 (words have complete opposite meanings) to 1 (words are exactly the same). Note that the original Word2Vec architectures do not take into account homographs, which limits its ability to accurately embed certain words as well as accurately calculate similarity. This shortcoming is further discussed in Section 7.

4.2 PubMed Embeddings

PubMed is a search engine comprising of over 30 million citations within the biomedical domain [1]. Because the database contains millions of scientific articles and abstracts, it has long been used for information retrieval and extraction tasks. Additionally, due to its large collection of unstructured text, the database lends itself nicely to the Word2Vec models discussed in Section 4.1. While investigating a query-document similarity score, Kim, et al., generated word embeddings using Word2Vec and PubMed article titles and abstracts [23]. More specifically, they used the Word2Vec skip-gram model architecture along with the abstracts and titles from over 25 million PubMed

[1] https://pubmed.ncbi.nlm.nih.gov/

documents to generate word embeddings. In terms of preprocessing, the authors removed all stopwords from the titles and abstracts. No other preprocessing information was given by the authors. Their word embedding model is available online. [2]

We use the aforementioned PubMed embedding model created by Kim, et al., as the foundation for our analysis because, most importantly, the embeddings were generated using a large set of training data, where this data is relevant to the mental health terms being investigated. Additionally, the model is publicly available online, allowing for others to easily download the model and replicate the results.

5 Methods

In the following three subsections, we outline the algorithm used for discovering relationships among words, provide the pseudocode for the algorithm, and present possible methods for analyzing a relationship once discovered.

5.1 Method for Discovering Relationships

Given a set of seed words, the goal is to connect each seed word to one another via links constructed of similar words. The resulting set of links constitute a relationship among the seed words. A link between two seed words is created when the seed words share one of the iteratively-generated similar words. Similar words are found using the cosine similarity measure. When a shared word is discovered, two partial links are created from the shared word back to each seed word. These partial links are then combined, and added onto the overall relationship.

To illustrate this process, let $\{s_1, s_2, \ldots, s_k\}$ be the set of seed words for the model and assume the algorithm is iterated n times with j similar words per word per iteration. The seed words, number of iterations, and number of similar words are all chosen before running the algorithm. Different values and combinations of these parameters and their effects on relationships are explored in Section 6.

Each seed word has its own list that contains the iteratively-generated simialr words. After each iteration, we must retain enough information so that, if there is an overlap between the seed words' similar lists, then we can successfully construct the link between seed words. Thus, every similar word entry added to a seed word's list takes the following form:

$$[word_{similar}, word_{generatedFrom}, iteration, cosSim], \tag{1}$$

where $word_{similar}$ is one of the top j similar words to $word_{generatedFrom}$, $iteration$ is the current iteration of the algorithm, and $cosSim$ is the cosine similarity between $word_{similar}$ and $word_{generatedFrom}$. The latter is used to evaluate the "strength" of the relationship.

The following construction process is repeated for each seed word $s_i \in \{s_1, s_2, \ldots, s_k\}$. For simplicity, we outline the general case for s_i.

First, we begin with the following set:

$$[[s_i, ORIG, 0, 1]], \tag{2}$$

which acts as the base of the list. Note that the current iteration is 0, and $ORIG$ is simply a placeholder to indicate that s_i is a seed word. Then, on the first iteration (iteration 1) of the algorithm, we generate the j most similar words (j highest cosine similarities) to all entries added

[2] https://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/DATASET/

on the previous iteration (iteration 0) and append them to the list with the form described in (1) above. Hence, we have

$$\begin{gathered}[[s_i, ORIG, 0, 1], \\ [w_1, s_i, 1, cosSim(w_1, s_i)], \\ [w_2, s_i, 1, cosSim(w_2, s_i)], \\ \dots \\ [w_j, s_i, 1, cosSim(w_j, s_i)]],\end{gathered} \tag{3}$$

where each $w_1, \dots w_j$ is one of the top j similar words to s_i. Again, note that the current iteration is 1 and that s_i replaced $ORIG$, as each $w_1, \dots w_j$ was generated from s_i. On the next iteration (iteration 2), we again generate the j most similar words to all entries added on the previous iteration (iteration 1) and append them in the correct form to the list. Hence, the list for s_i now looks like

$$\begin{gathered}[[s_i, ORIG, 0, 1], \\ [w_1, s_i, 1, cosSim(w_1, s_i)], \\ [w_2, s_i, 1, cosSim(w_2, s_i)], \\ \dots, \\ [w_j, s_i, 1, cosSim(w_j, s_i)], \\ [w_{1,1}, w_1, 2, cosSim(w_1, w_{1,1})], \\ \dots, \\ [w_{1,j}, w_1, 2, cosSim(w_1, w_{1,j})], \\ [w_{2,1}, w_2, 2, cosSim(w_2, w_{2,1})], \\ \dots, \\ [w_{2,j}, w_2, 2, cosSim(w_2, w_{2,j})], \\ \dots, \\ [w_{j,j}, w_j, 2, cosSim(w_j, w_{j,j})]].\end{gathered} \tag{4}$$

The subscript pair on each generated $w_{m,k}$ above should be read as "$w_{m,k}$ is the m^{th} most similar word to w_k. Note that there are now j^2 entries in list (4) above. This process is repeated until the n^{th} iteration is reached. Upon completion, for each seed word $s_i \in \{s_1, s_2, \dots, s_k\}$, we now have a list containing all information needed to begin constructing the links and relationships. The construction of the seed word lists is described as pseudocode in Algorithm 1.

There exists a relationship if two or more original seed words can be linked via a path consisting of generated similar words. To illustrate this, take the initial set of seed words $\{s_1, s_2, \dots, s_k\}$ and let $S(s_i)$ denote the set of j most similar words to word s_i, and suppose we only look at the first iteration. On the first iteration, we generate set $\{S(s_1), S(s_2), \dots, S(s_k)\}$ containing sets of similar words for each seed word. Next, we check the intersection between every combination of sets in $\{S(s_1), S(s_2), \dots, S(s_k)\}$. Any shared words constitute a relationship. For example, suppose $S(s_1) \cap S(s_2) = \{w\}$. In this case, word w is one of the j most similar words to both s_1 and s_2. Using $\iff$ to denote "is similar to", we can write the relationship as $\{s_1\} \iff \{w\} \iff \{s_2\}$. Or, for example, suppose $\bigcap\{S(s_1), S(s_2), \dots, S(s_k)\} = \{w\}$. This indicates the relationship

$$\begin{aligned} \{w\} &\iff \{s_1\} \\ &\iff \{s_2\} \\ &\dots \\ &\iff \{s_k\} \end{aligned} \tag{5}$$

For multiple iterations, a relationship may look something like

$$\{s_1\} \iff \{x_1\} \iff \{y_2\} \iff \{z_1\} \iff \{s_2\} \tag{6}$$

where $x_1 \in S(s_1)$, $z_1 \in S(s_2)$, and $y_2 \in S(S(s_1)) \cap S(S(s_2))$. As one can see, there are numerous possible paths for relationships to be formed. A seed word may be generated from the similar words of another, or perhaps similar words from different iterations may match. Additionally, the number of words (and combinations) grows exponentially, so only a few iterations are considered in this paper. Further discussion on increasing the number of iterations as well as potentially limiting the generated words to avoid exponential growth can be found in Section 7.

After discovering a shared word between seed word lists, the next step is to effectively "undo" the aforementioned process in order to construct the path of words to each seed word to create the relationship. And because the entries in the seed word lists are of the form described in (1), we can use this information to construct the paths.

Suppose we have the following two different seed word lists:

$$[[s_1, ORIG, 0, 1], [w, s_1, 1, cosSim(w, s_1)], \dots] \tag{7}$$
$$[[s_2, ORIG, 0, 1], [w, s_2, 1, cosSim(w, s_2)], \dots] \tag{8}$$

and we find that w is shared between the two lists. Beginning with list (7) and referring to entry $[w, s_1, 1, cosSim(w, s_1)]$, we must find another entry in (7) generated on the previous iteration that contains s_1 as that first entry. Clearly, $[s_1, ORIG, 0, 1]$ satisfies this, and because we cannot decrement the iteration anymore, we have found the portion of the relationship linking seed word s_1 to w:

$$[s_1, ORIG, 0, 1] \iff [w, s_1, 1, cosSim(w, s_1)] \tag{9}$$

Repeating this for list (8), we find the part of the relationship linking seed word s_2 to w:

$$[s_2, ORIG, 0, 1] \iff [w, s_2, 1, cosSim(w, s_2)] \tag{10}$$

Finally, combining (9) and (10) give us the complete relationship

$$[s_1, ORIG, 0, 1] \iff \{[w, s_1, 1, cosSim(w, s_1)], [w, s_2, 1, cosSim(w, s_2)]\} \iff [s_2, ORIG, 0, 1]$$

Or, reduced to just the words

$$\{s_1\} \iff \{w\} \iff \{s_2\} \tag{11}$$

This process to find shared words between seed word lists (in the form of (4)) and to generate a relationship is detailed in Algorithm 2. Additionally, the process of backtracing through seed word lists to construct a link from a shared word to a seed word is outlined in Algorithm 3.

Although this example only discusses a single iteraiton and two seed words, this process can be repeated for any number of iterations and seed words. Note that the maximum relationship length is limited by the number of iterations performed. Each iteration affects each seed word list independently, thus the length from seed word to shared word grows exactly with the number of iterations. Because a relationship goes from seed word to shared word to seed word, the maximum length of a relationship is bounded by $2i - 1$, where i is the iteration. Note that subtracting one is needed otherwise the shared word would be double-counted. Additionally, note that there is a possibility for duplicate and trivial relationships to be formed. To see how this happens, suppose $S(s_1) \cap S(s_2) = \{w\}$. Then on the next iteration $S(w) \in S(S(s_1)) \cap S(S(s_2))$, which will result in relationships with links containing more than one shared word. This is combated by ignoring any pairs of links containing more than two of the same word.

Once a relationship is discovered, we store it in a dictionary data structure, where the keys are the shared words (i.e., $\{w\}$ from (11) in the above example) and the values are all of the paths from the key to the seed words which are discovered (i.e., (9) and (10) in the above example).

Following the completion of the algorithm, this dictionary should contain all relationships among seed words for the given parameters. Thus, we can now assess the merit of each relationship, look at statistical measures such as word frequency or relationship strength as a basis for verifying and discovering meaningful relations among the seed words.

5.2 Pseudocode

This section presents the pseudocode for generating seed word lists given a set of seed words (Algorithm(1)), discovering shared words between seed words along with creating the relationship (Algorithm(2)), and constructing the links back to seed words after finding a shared word (Algorithm (3)). Note that various optimizations are ignored in the pseudocode for presentation purposes.

Algorithm 1 Generate Seed Word Lists

```
1:  procedure SeedWordLists(L, iteration, numSimilar)        ▷ L is list of seed words
2:      M ← []                                          ▷ To store generated seed word lists
3:      for word in L do                                                ▷ For each seed word
4:          M.append([[word,ORIG,0,1]])
5:      for seedList in M do                                       ▷ For each seed word list
6:          for i in 0,1,...,iteration do                                ▷ For each iteration
7:              for entry in seedList do
8:                  if entry[2] = i then                          ▷ If from previous iteration
9:                      curWord ← entry[0]
10:                     simWords ← most numSimilar similar words to curWord
11:                     for simWord in simWords do          ▷ Add all similar words in correct form
12:                         seedList.append([simWord,curWord,i+1,cosSim(curWord, simWord)])
13:     return M
```

Algorithm 2 Find all relationships given a set of seed word lists

```
1: procedure FindRelationship(SeedWordList)
2:     D ← {}                                          ▷ D is dictionary to store all relationships
3:     i ← 0
4:     for i in 0,1,...,len(seedWordList) − 1 do
5:         for entry in seedList[i] do                  ▷ For each word in seed list
6:             for seedList′ in seedWordList[i + 1 :] do
7:                 for entry′ in seedList′ do
8:                     if entry[0] = entry′[0] then     ▷ Two seed lists share a common word
9:                         link ←backtrace(entry, seedList)
10:                        link′ ←backtrace(entry′, seedList′)
11:                        D[entry].add(link)
12:                        D[entry].add(link′)
13:    return D
```

Algorithm 3 Construct a link from shared word to seed word

```
1: procedure Backtrace(entry, seedList)
2:     i ← entry[2] − 1                                ▷ i is iteration tracker
3:     word ← entry[1]                                 ▷ word is word we search for
4:     link ← [entry]
5:     while i ≥ 0 do
6:         for entry′ in seedList do
7:             if entry′[2] = i and entry′ = word then  ▷ If same word on previous iteration
8:                 link.append(entry′)
9:                 i ← i − 1                           ▷ Now looking at previous iteration
10:                word ← entry′[1]                    ▷ Set word to the word that generated word
11:    return link
```

5.3 Method for Analyzing Relationships

After gathering all possible links between the seed words with the given algorithm parameters, we then proceeded to analyze the relationship in varying ways. One measure is averaging the cosine similarity for each immediate word pair across the entire relationship. For example, given the following relationship:

$$\{s_1\} \iff \{s_1'\} \iff \{w\} \iff \{s_2'\} \iff \{s_2\} \tag{12}$$

we can look at the similarity for the entire relationship:

$$\frac{cosSim(s_1, s_1') + cosSim(s_1', w) + cosSim(w, s_2') + cosSim(s_2', s_2)}{4} \tag{13}$$

or the similarity for the various links (in this case, seed word to shared word) within the relationship:

$$\frac{cosSim(s_1, s_1') + cosSim(s_1', w)}{2}, \frac{cosSim(w, s_2') + cosSim(s_2', s_2)}{2} \tag{14}$$

The above measure (14) is written pair-wise because there exist two links, one from seed word s_1 to shared word w, and one from seed word s_2 to shared word w. Intuitively, the similarity measure can be thought of as the "strength" of the overall relationship or individual link.

Another way to assess the resulting relationships is to look at recurring words within the links of relationships, along with the total number of times each word appears. This measure becomes important as the number of iterations grows because this causes the relationships to grow more and more complex. Relationships and links can also be analyzed on an individual word level, such as by looking at the type of shared words. For example, relationships can be filtered by restricting the shared words to disease or illness names. This is explored in Section 6.

One way to filter relationships is via part-of-speech (POS). For example, the resulting relationships can be filtered to only include links containing verbs or adjectives. The Word2Vec model used in this paper does not take into account multiple tags for the same word, nor does it record the sentences used for training. Therefore, POS tagging must be done after the fact, which limits the ability to accurately tag word. This limitation is further discussed in Section 7.

6 Results

In this section, each relationship is represented as a set of links between seed words and shared words. Each link takes the following form:

$$\{shared\} \iff \{w_n\} \iff \ldots \{w_1\} \iff \{seed\} \tag{15}$$

where *seed* is one of the seed words and $w_1, \ldots, w_n, shared$ are the generated similar words. Because we are interested in connecting seed words, a relationship may look something like this:

$$\begin{aligned} \{shared\} &\iff \{w_n\} \iff \ldots \{w_1\} \iff \{seed_1\} \\ \{shared\} &\iff \{w_k'\} \iff \ldots \{w_1'\} \iff \{seed_2\} \end{aligned} \tag{16}$$

which can also be written as

$$\{seed_1\} \iff \{w_1\} \iff \cdots \iff \{shared\} \iff \cdots \iff \{w_1'\} \iff \{seed_2\}. \tag{17}$$

or as

$$\begin{aligned}\{shared\} &\iff \{w_n\} \iff \cdots \iff \{w_1\} \iff \{seed_1\} \\ &\iff \{w'_k\} \iff \cdots \iff \{w'_1\} \iff \{seed_2\}\end{aligned} \tag{18}$$

The simpler relationships, such as those between two seed words created on a single iteration (i.e., those described in Section 6.1), are written using the method described in (17) above. However, the more complex relationships involving multiple seed words and iterations are written using the method described in (18) above. For some of the links and relationships, we appended the averaged cosine similarity score (when needed) in order to gauge the strength.

6.1 "Anxiety" and "Depression"

First, we investigated the resulting relationships between two seed words, "anxiety" and "depression", and how they are affected by varying the number of similar words. It is important to note that "anxiety" and "depression" are relatively similar to begin with (high cosine similarity score), thus it is expected that they will share many words in their respective similar-word lists.

With a single iteration and five similar words per iteration, we get the following relationship, along with the averaged cosine similarities:

$$\{anxiety\} \iff \{depression\}, (0.9366) \tag{19}$$

$$\{anxiety\} \iff \{depressive\} \iff \{depression\}, (0.9239) \tag{20}$$

$$\{anxiety\} \iff \{mood\} \iff \{depression\}, (0.9039) \tag{21}$$

Note that link (19) indicates that both seed words appear in each others top five similar words. Increasing to ten similar words, we get the same relationship as above, plus the following:

$$\{anxiety\} \iff \{hopelessness\} \iff \{depression\}, (0.8843) \tag{22}$$

$$\{anxiety\} \iff \{somatization\} \iff \{depression\}, (0.8826) \tag{23}$$

$$\{anxiety\} \iff \{psychopathology\} \iff \{depression\}, (0.8799) \tag{24}$$

$$\{anxiety\} \iff \{hyperarousal\} \iff \{depression\}, (0.8714) \tag{25}$$

$$\{anxiety\} \iff \{anhedonia\} \iff \{depression\}, (0.8673) \tag{26}$$

With twenty similar words, we again get the same relationships as above, plus the following:

$$\{anxiety\} \iff \{suicidality\} \iff \{depression\}, (0.8655) \tag{27}$$

$$\{anxiety\} \iff \{ptsd\} \iff \{depression\}, (0.8590) \tag{28}$$

$$\{anxiety\} \iff \{phobia\} \iff \{depression\}, (0.8559) \tag{29}$$

$$\{anxiety\} \iff \{panic\} \iff \{depression\}, (0.8425) \tag{30}$$

$$\{anxiety\} \iff \{dysphoria\} \iff \{depression\}, (0.8406) \tag{31}$$

One immediately noticeable trend is the general decrease of average cosine similarity as the number of similar words increases, which is to be expected given the method of discovery. Looking more closely at the words constituting the above relationships, we find many interesting properties. For example, (21) indicates that both "depression" and "anxiety" are a kind of mood. Similarly, (24) verifies the fact that both "depression" and "anxiety" fall under the umbrella of psychopathology (the study of abnormal cognitions, behaviors, and experiences). More complex relationships are shown in (27) and (28), where both "anxiety" and "depression" are verified to be significantly related with suicide [6] and PTSD [11, 34], respectively.

6.2 Illness from Symptoms

Now suppose we want to see which mental illness is characterized by the following four symptoms: "hallucinations", "delusions", "panic", and "paranoia". Using the symptoms as seed words, and iterating the algorithm once with 150 similar words results in many links. We can filter out trivial links (those that are constructed using a variation of a seed word, basic verbs) as well as links that are unique to a subset of the seed words (as we are interested in a relationship between all four seed words). Note that, rather than writing $\{shared\} \iff \{seed_i\}$ on each line, we combined all of the seed words onto a single line. The similarity score appended onto the end of each relationship is the averaged cosine similarity between the shared word and each seed. Two of the strongest resulting relationship are as follows:

$$\{psychotic\} \iff \{hallucinations, delusions, panic, paranoia\}, (0.8343) \tag{32}$$

$$\{psychosis\} \iff \{hallucinations, delusions, panic, paranoia\}, (0.7986) \tag{33}$$

Therefore, the results seem to suggest that "hallucinations", "delusions", "panic", and "paranoid" are characteristics of "psychosis".

6.3 Multiple Iterations

Next, we look at a relationship constructed using multiple iterations. Consider the following two seed words, "sleep" and "tourette" with three iterations and ten similar words per iteration. This results in only three components of the overall relationship, with shared words "depressive", "panic", and "depressives":

$$\begin{aligned} \{depressive\} &\iff \{insomnia\} \iff \{sleepiness\} \iff \{sleep\} \\ &\iff \{psychotic\} \iff \{catatonia\} \iff \{tourette\} \\ &\iff \{mdd\} \iff \{schizophrenia\} \iff \{tourette\} \\ &\iff \{psychosis\} \iff \{schizophrenia\} \iff \{tourette\} \end{aligned} \tag{34}$$

$$\begin{aligned} \{panic\} &\iff \{insomnia\} \iff \{sleepiness\} \iff \{sleep\} \\ &\iff \{mania\} \iff \{catatonia\} \iff \{tourette\} \\ &\iff \{manic\} \iff \{catatonia\} \iff \{tourette\} \\ &\iff \{psychotic\} \iff \{catatonia\} \iff \{tourette\} \\ &\iff \{tic\} \iff \{gts\} \iff \{tourette\} \\ &\iff \{mdd\} \iff \{schizophrenia\} \iff \{tourette\} \end{aligned} \tag{35}$$

$$\begin{aligned} \{depressives\} &\iff \{insomniacs\} \iff \{sleepiness\} \iff \{sleep\} \\ &\iff \{schizophrenic\} \iff \{schizophrenia\} \iff \{tourette\} \\ &\iff \{schizophrenics\} \iff \{schizophrenia\} \iff \{tourette\} \end{aligned} \tag{36}$$

Note that the relationship is written in the form of (18) above. From (34), (35), and (36), it seems like "sleep" is related to "tourette" through traits such as "panic" and "mania", as well

as “depressive(s)” and “catatonia”. Another way to view this relationship is graphically via the word embeddings. Using the above relationship along with Principle Component Analysis (PCA) to reduce the dimension of the embeddings to two, we get the graph displayed in Figure 3. Seed words “sleep” and “tourette” are depicted as black circles, and all words listed in (34), (35), and (36) are depicted as red stars. All of the similar words not in a link are blue rings. Note the two distinctive clusters, separated by X=0, which correspond to the similar words for each seed word. Due to the loss in dimensionality, the links from one seed word to another (colored red) may get skewed.

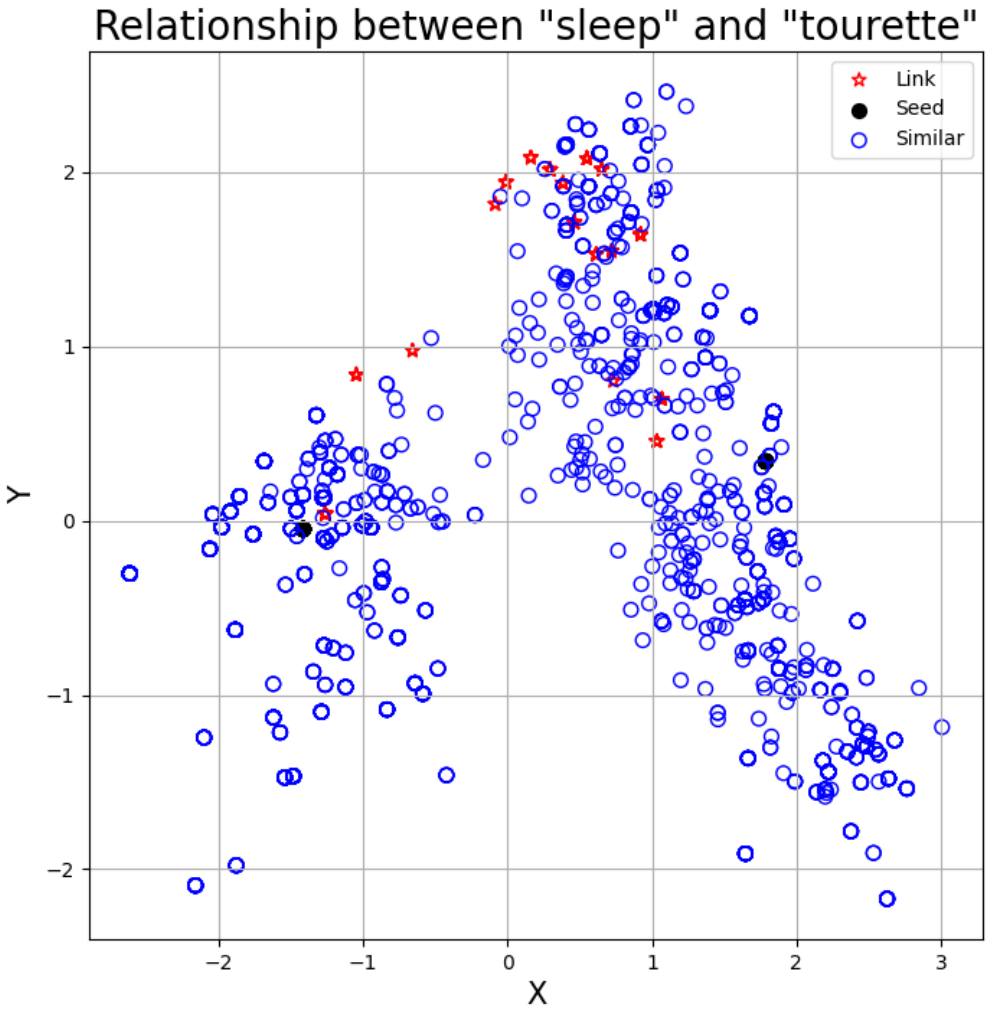

Figure 3: Graphical representation of the relationship between “sleep” and “tourette”. Seed word “sleep” is located at approximately (-1.5,-0.05) and seed word “tourette” is located at approximately (1.8,0.34).

6.4 Another Example of Multiple Iterations

Finally, suppose we use the seed words "violence" and "disorder" with two iterations and twenty similar words per iteration. This gives us the following relationship components (37) and (38), along with the PCA graph depicted in Figure 4.

$$\begin{aligned} \{psychopathology\} &\iff \{maltreatment\} \iff \{violence\} \\ &\iff \{psychosis\} \iff \{disorder\} \\ &\iff \{phobia\} \iff \{disorder\} \end{aligned} \tag{37}$$

$$\begin{aligned} \{suicidality\} &\iff \{maltreatment\} \iff \{violence\} \\ &\iff \{psychosis\} \iff \{disorder\} \\ &\iff \{phobia\} \iff \{disorder\} \\ &\iff \{victimisation\} \iff \{violence\} \\ &\iff \{homelessness\} \iff \{violence\} \end{aligned} \tag{38}$$

Not surprisingly, the links from both "violence" and "disorder" share "psychopathology" (the study of abnormal cognitions), as listed in (37). More interestingly, however, are the words in (38). The links meet on "suicidality", and contain words such as "maltreatment", "victimisation", and "homelessness". Additionally, the graph in Figure 4 shows a clear separation along X=0, and the path from "violence" and "depression" seems a bit more distinct than the path depicted in Figure 3.

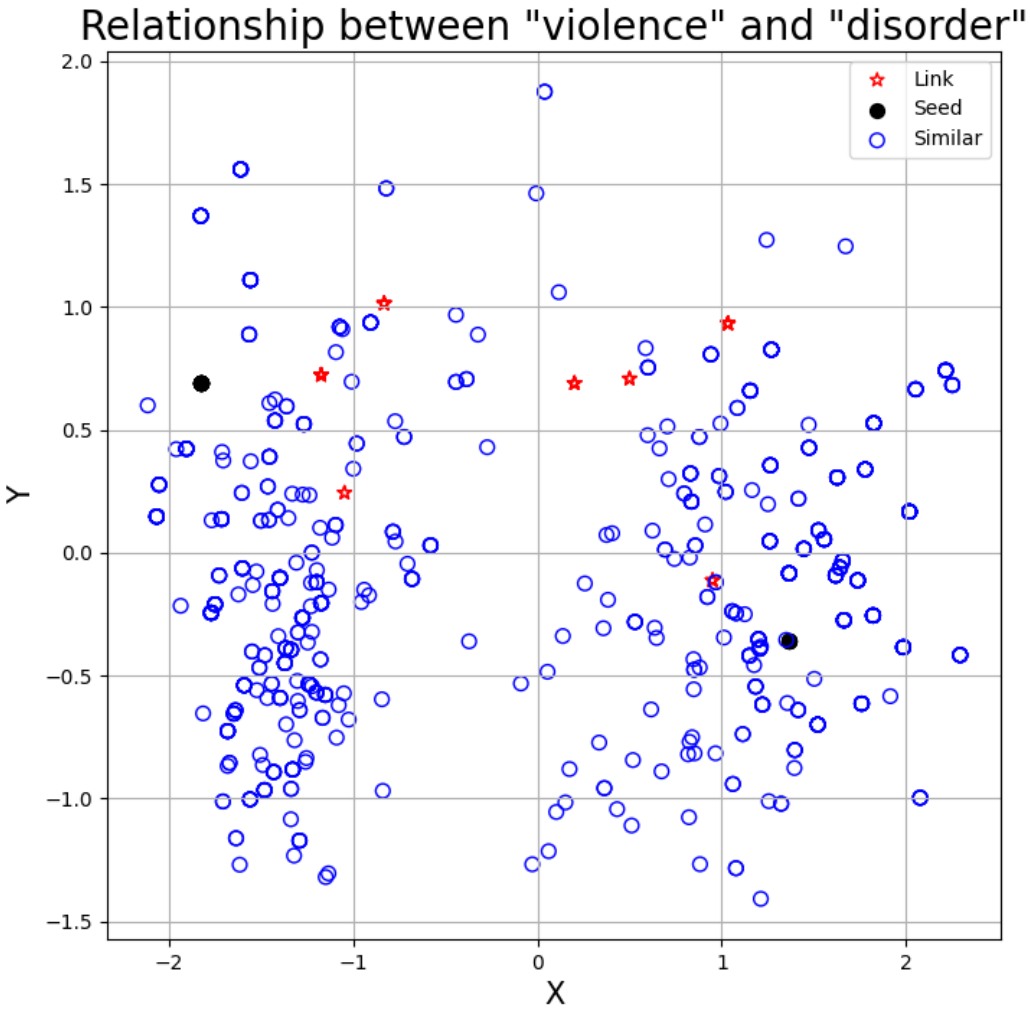

Figure 4: Graphical representation of the relationship between “violence” and “disorder”. Seed word “violence” is located at approximately (-1.83, 0.69) and seed word “disorder” is located at approximately (1.36,-0.36)

7 Limitations, Extensions and Future Work

7.1 Limitations

One limitation is the exponential growth rate of the number of similar words added to each seed word list on each successive iteration. Although this exponential addition potentially leads to more relationships being discovered, it is certainly not efficient nor scalable. Thus, one solution would be to limit the total number of words that can be generated, or reduce the number of similar words on each successive iteration.

Another limitation is the lack of relationships discovered from using seed words with different parts of speech. Generally, words with a certain part of speech will be most similar to words with

the same part of speech, especially in the Word2Vec embedding model used in this paper. Because of this, certain seed words will rarely share a common word, until the seed word lists grow very large (see above paragraph). A potential solution would be to only add similar words with a certain part of speech, but this requires the words to be tagged prior to embedding generation.

7.2 Extensions and Future Work

Regarding extensions of the results, it follows that one may repeat this investigation with different seed words, algorithm parameters, and even on different domains. One could compare resulting relationships between the same seed words across multiple domains, with the hope of quantifying differences between term usage. Additionally, one could also investigate the same relationships using various embedding models, such as embeddings generated using Wikipedia or publications pertaining specifically to mental health. This algorithm could also be applied to text that has already been processed with named-entity recognition or entity linking.

Regarding extensions of the methods, one could integrate part-of-speech filtering, either by using a more informative embedding model or by maintaining the tagged, original sentences. Or, perhaps, one could adapt the algorithm to link n-grams or concepts instead of words. Similar to this, using a more informative embedding model could also help resolve ambiguity regarding homographs in the training corpora. Part-of-speech filtering not only shows more relevant relationships, but also reduces the total number of relationships returned. Another way this can be achieved is though adding domain-specific knowledge, such as disease names, and filtering via this knowledge.

8 Conclusion

In this paper, we proposed a novel, unsupervised, and domain-independent algorithm for constructing relationships among any number of words. The resulting relationships, consisting of various semantically-related links between words, can be used for tasks such as verifying known relationships, generating hypotheses, and discovering new relationships. In the context of mental health, practitioners could use the algorithm and resulting relationships to relate symptoms with illnesses, track changes in diagnostic vocabulary, or investigate relationships between seemingly unrelated diseases. In a broader context, one could easily imagine a geologist using the proposed algorithm to investigate the relationships between epochs and minerals, a pharmacist relating drugs to various side effects, or even a politician linking different policies. Regardless of the domain, the proposed algorithm can be used as the first step in investigating the complex and ever-increasing relationships encountered in scientific literature.

References

[1] Edgar Altszyler, Mariano Sigman, Sidarta Ribeiro, and Diego Fernández Slezak. Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. *arXiv preprint arXiv:1610.01520*, 2016.

[2] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. *Advances in Automatic Text Summarization*, pages 111–121, 1999.

[3] Adrian Benton, Margaret Mitchell, and Dirk Hovy. Multi-task learning for mental health using social media text. *arXiv preprint arXiv:1712.03538*, 2017.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Daniel Eisenberg, Sarah E Gollust, Ezra Golberstein, and Jennifer L Hefner. Prevalence and correlates of depression, anxiety, and suicidality among university students. *American Journal of Orthopsychiatry*, 77(4):534–542, 2007.

[7] Vishrawas Gopalakrishnan, Kishlay Jha, Wei Jin, and Aidong Zhang. A survey on literature based discovery approaches in biomedical domain. *Journal of Biomedical Informatics*, page 103141, 2019.

[8] Michael D Gordon and Susan Dumais. Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science*, 49(8):674–685, 1998.

[9] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[10] Sam Henry and Bridget T McInnes. Literature based discovery: models, methods, and trends. *Journal of Biomedical Informatics*, 74:20–32, 2017.

[11] Stefan G Hofmann, Brett T Litz, and Frank W Weathers. Social anxiety, depression, and PTSD in Vietnam veterans. *Journal of Anxiety Disorders*, 17(5):573–582, 2003.

[12] Thomas Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[13] Dimitar Hristovski, Andrej Kastrin, Borut Peterlin, and Thomas C Rindflesch. Combining semantic relations and DNA microarray data for novel hypotheses generation. In *Linking Literature, Information, and Knowledge for Biology*, pages 53–61. Springer, 2010.

[14] Bo Hu and Boris Villazon Terrazas. Building a mental health knowledge model to facilitate decision support. In *Pacific Rim Knowledge Acquisition Workshop*, pages 198–212. Springer, 2016.

[15] Wendy Hui and Wai Kwong Lau. Application of Literature-Based Discovery in Nonmedical Disciplines: A Survey. In *Proceedings of the 2nd International Conference on Computing and Big Data*, pages 7–11, 2019.

[16] Vitavin Ittipanuvat, Katsuhide Fujita, Ichiro Sakata, and Yuya Kajikawa. Finding linkage between technology and social issue: A Literature Based Discovery approach. *Journal of Engineering and Technology Management*, 32:160–184, 2014.

[17] Kishlay Jha and Wei Jin. Mining hidden knowledge from the counterterrorism dataset using graph-based approach. In *International Conference on Applications of Natural Language to Information Systems*, pages 310–317. Springer, 2016.

[18] Shan Jiang and ChengXiang Zhai. Random walks on adjacency graphs for mining lexical relations from big text data. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 549–554. IEEE, 2014.

[19] Zhenchao Jiang, Lishuang Li, Degen Huang, and Liuke Jin. Training word embeddings for deep learning in biomedical text mining tasks. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 625–628. IEEE, 2015.

[20] Wei Jin and Rohini K Srihari. Knowledge discovery across documents through concept chain queries. In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pages 448–452. IEEE, 2006.

[21] Rob Johnson, Anthony Watkinson, and Michael Mabe. The STM report. *An Overview of Scientific and Scholarly Publishing. 5th edition October*, 2018.

[22] Nathan Kibwami and Apollo Tutesigensi. Using the literature based discovery research method in a context of built environment research. In *Proceedings 30th Annual ARCOM Conference*, volume 1, pages 227–236. ARCOM, 2014.

[23] Sun Kim, W John Wilbur, and Zhiyong Lu. Bridging the gap: a semantic similarity measure between queries and documents. *arXiv preprint arXiv:1608.01972*, 2016.

[24] Ronald N Kostoff. Literature-Related Discovery: Potential treatments for SARS.

[25] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18(1):198, 2017.

[26] Lishuang Li, Jieqiong Zheng, Jia Wan, Degen Huang, and Xiaohui Lin. Biomedical event extraction via long short term memory networks along dynamic extended tree. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 739–742. IEEE, 2016.

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

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[29] Majid Rastegar-Mojarad, Ravikumar Komandur Elayavilli, Dingcheng Li, Rashmi Prasad, and Hongfang Liu. A new method for prioritizing drug repositioning candidates extracted by literature-based discovery. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 669–674. IEEE, 2015.

[30] Nadeem N Rather, Chintan O Patel, and Sharib A Khan. Using deep learning towards biomedical knowledge discovery. *Int. J. Math. Sci. Comput.(IJMSC)*, 3(2):1–10, 2017.

[31] Lawrence Reeve, Hyoil Han, and Ari D Brooks. BioChain: lexical chaining methods for biomedical text summarization. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, pages 180–184, 2006.

[32] Lawrence H Reeve, Hyoil Han, and Ari D Brooks. Biomedical text summarisation using concept chains. *International Journal of Data Mining and Bioinformatics*, 1(4):389–407, 2007.

[33] Asif Salekin, Jeremy W Eberle, Jeffrey J Glenn, Bethany A Teachman, and John A Stankovic. A weakly supervised learning framework for detecting social anxiety and depression. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):1–26, 2018.

[34] Kathryn P Wilder Schaaf, Laura K Artman, Mary Ann Peberdy, William C Walker, Joseph P Ornato, Michelle R Gossip, Jeffrey S Kreutzer, Virginia Commonwealth University ARCTIC Investigators, et al. Anxiety, depression, and PTSD following cardiac arrest: a systematic review of the literature. *Resuscitation*, 84(7):873–877, 2013.

[35] Judy Hanwen Shen and Frank Rudzicz. Detecting anxiety through reddit. In *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality*, pages 58–65, 2017.

[36] Padmini Srinivasan. Text mining: generating hypotheses from MEDLINE. *Journal of the American Society for Information Science and Technology*, 55(5):396–413, 2004.

[37] Don R Swanson. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30(1):7–18, 1986.

[38] Don R Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118, 1986.

[39] Don R Swanson. Migraine and magnesium: eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557, 1988.

[40] Don R Swanson and Neil R Smalheiser. An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence*, 91(2):183–203, 1997.

[41] Buzhou Tang, Hongxin Cao, Xiaolong Wang, Qingcai Chen, and Hua Xu. Evaluating word representation features in biomedical named entity recognition tasks. *BioMed Research International*, 2014, 2014.

[42] Rein Vos, Sil Aarts, Erik van Mulligen, Job Metsemakers, Martin P van Boxtel, Frans Verhey, and Marjan van den Akker. Finding potentially new multimorbidity patterns of psychiatric and somatic diseases: exploring the use of literature-based discovery in primary care research. *Journal of the American Medical Informatics Association*, 21(1):139–145, 2014.

[43] Hsih-Te Yang, Jiun-Huang Ju, Yue-Ting Wong, Ilya Shmulevich, and Jung-Hsien Chiang. Literature-based discovery of new candidates for drug repurposing. *Briefings in Bioinformatics*, 18(3):488–497, 2017.

[44] Rui Zhang, Michael J Cairelli, Marcelo Fiszman, Halil Kilicoglu, Thomas C Rindflesch, Serguei V Pakhomov, and Genevieve B Melton. Exploiting literature-derived knowledge and semantics to identify potential prostate cancer drugs. *Cancer informatics*, 13:CIN–S13889, 2014.