# Investigating Word Embedding Techniques for Extracting Disease, Gene, and Chemical Relationships from Biomedical Texts

By

Sushumna S Pradeep

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science
at
Dalhousie University
Halifax, Nova Scotia
August 2024

I dedicate this work to Anantha Padmanabha Swamy and my family.

## Table of Contents

## List of Tables

## List of Figures

## Abstract

This thesis investigates word embedding models, including PubMedBERT, BioBERT, SkipGram, CBOW, and GloVe, in the context of Literature-Based Discovery (LBD) within biomedical research, with a specific focus on cancer-related entities. Firstly, I study the effectiveness of word embedding models in identifying current functional relationships (e.g., interaction) between genes, diseases, and chemicals, as recorded in the medical literature. As a reference, I use curated functional relationships from the Comparative Toxicogenomics Database (CTD). The goal is to evaluate each word embedding model, highlighting their strengths and weaknesses in identifying functional relationships in particular, and in biomedical text mining in general.

Next, I study the ability of word embedding models in discovering previously unknown functional relationships from the medical literature. I create word embeddings from the medical literature up until 2022, and check whether they can identify functional relationships that were not in CTD at that time (i.e., functional relationships found in CTD version 2024 but not part of CTD version 2022; time-slicing). If this is successful, it means that word embedding models can conduct LBD; they can identify previously unknown functional relationships[1] from the medical literature.

We created word embeddings using models such as CBOW, SkipGram, GloVe, BioBERT, and PubMedBERT based on PubMed abstracts up to 2022. After generating the embeddings, we measured functional relatedness using cosine similarity for curated pairs from the CTD dataset. To evaluate the performance of these models, we calculated precision and recall by comparing the curated CTD pairs with the instance vector pairs of instances from CTD, using cosine similarity thresholds of 0.6, 0.7, and 0.8. Once these values were obtained, heatmaps were plotted to compare model performance and identify which model produced the best results.

The findings reveal that PubMedBERT and BioBERT significantly outperform traditional models like CBOW, SkipGram, and GloVe both on precision and recall; especially at a cosine similarity threshold of 0.7, which has been identified as an optimal balance between accuracy and comprehensive data retrieval.

The results also show that the word embeddings created from PubMed abstracts up to 2022 are able to capture functional relationships in newly curated pairs from the CTD dataset. Specifically, the dataset included 157 disease-chemical pairs, 138 disease-gene pairs, and 191 chemical-gene pairs. Using the generated word embeddings, the model successfully captured relatedness in 42 disease-chemical pairs, 58 disease-gene pairs, and 83 chemical-gene pairs.


[1] At last, functional relationships that were not yet part of the curated CTD at that time.

## List of Abbreviations Used

**BERT**: Bidirectional Encoder Representations from Transformers

**BioBERT**: Bidirectional Encoder Representations from Transformers for Biomedical Text Mining

**CBOW**: Continuous Bag of Words

**CHI**: Calinski-Harabasz Index

**CLS**: Classification

**CNN**: Convolutional Neural Network

**CSF-1**: Colony Stimulating Factor 1

**CSV**: Comma-Separated Values

**CTD**: Comparative Toxicogenomics Database

**GFP**: Green Fluorescent Protein

**GloVe**: Global Vectors

**GloVe/42B**: Global Vectors for Word Representation trained on a corpus of 42 billion tokens

**GloVe/840B**: Global Vectors for Word Representation trained on a corpus of 840 billion tokens

**JSON**: JavaScript Object Notation

**L1**: Manhattan Distance

**L2**: Euclidean Distance

**LBD**: Literature-Based Discovery

**LLM**: Large Language Model

**MLM**: Masked Language Model

**MN**: Medical Subject Headings Number

**NaN**: Not a Number

**NASARI**: Normed Association and Similarity Analysis Representation for Information

**NER**: Named Entity Recognition

**NIH**: National Institutes of Health

**NLM**: National Library of Medicine

**NLTK**: Natural Language Toolkit

**NLP**: Natural Language Processing

**NNLM**: Neural Network Language Model

**np**: Numpy

**NSP**: Next Sentence Prediction

**pd**: pandas

**PMC**: PubMed Central

**PMID**: PubMed ID

**PubMedBERT**: Bidirectional Encoder Representations from Transformers for PubMed articles

**ReLU**: Rectified Linear Unit

**RNN**: Recurrent Neural Network

**SEP**: Separator

**SG**: SkipGram

**t-SNE**: t-Distributed Stochastic Neighbor Embedding

**UI**: Unique Identifier

**W**: Window Size

**V**: Vector Size

**W2V**: Word2Vec

# Acknowledgements

First and foremost, I thank God for his blessings and guidance.

I extend my gratitude to my professor, Dr. Raza Abidi, for his unwavering support and guidance throughout this journey. I am also immensely grateful to my co-supervisor, Dr. William Van Woensel, for his invaluable insights. Additionally, my thanks go to Dr. Samina Abidi for her generous funding support.

A very special thanks to Dr. Ali Daowd for his encouragement and assistance.

I would also like to extend my special thanks to Dr. Michael McAllister for his timely support and contributions to complete my research work in time for which I'm forever indebted to.

I want to express my heartfelt thanks to my parents, Dr. Manjula S. N. and G. S. Srinivasa Pradeep, for their love, encouragement, and relentless push. Their blessings at every step of my life have been my guiding light.

To my family and friends, both back home and here, your support has been invaluable. I am also deeply grateful to my grandparents for their blessings, which have been a constant source of strength. Special mention goes to the Nanjunda Sharma and Savitramma family for their love and constant encouragement.

I would also like to extend a special mention to my friends, Keerthana, Naveen and Kritika for being there for me at every step of my thesis, through thick and thin. They were my stress busters, always ready to lift my spirits and keep me going. To all my friends here Lav, Arpit,Neha and back home  thank you for your love, support, and encouragement. Each one of you has played a significant role in helping me achieve this milestone

# 1. Introduction

## 1.1 Motivation

The exploding volume of biomedical literature presents both opportunities and challenges, especially in complex fields like cancer research [99]. The complexity of cancer research, combined with the sheer scale of available data, necessitates the use of advanced analytical tools to efficiently extract and analyze meaningful information. The Literature-Based Discovery (LBD) field involves identifying hidden or previously unknown connections within vast bodies of scientific literature [7]. LBD includes techniques such as co-occurrence analysis which involves identifying terms or concepts that frequently co-occur in the same document or across a set of document [97], natural language processing and data mining, to bridge knowledge gaps by linking concepts that do not co-occur directly but are related through intermediary terms, enabling the generation of new hypotheses [8].

Word embeddings, a natural language processing (NLP) technique, convert words and phrases into dense, continuous vector representations that encapsulate semantic relationships based on context and co-occurrence patterns. While they have been successfully applied in NLP tasks, their potential in LBD [27], especially in identifying functional relationships across diverse biomedical concepts, such as gene-disease or chemical-disease associations, remains underexplored.

Functional relatedness, which refers to the association of different biomedical concepts based on their roles, activities, or interactions within biological systems, is particularly important in cancer research. For instance, identifying how genes, chemicals and diseases interact within the context of cancer, can lead to the discovery of new therapeutic targets and treatment strategies. This study aims to evaluate the effectiveness of state-of-the-art word embedding techniques, including GloVe, BERT, PubMedBERT, BioBERT and Word2Vec, in supporting LBD. In particular, we focus on their ability to capture current and previously unknown functional relations across multiple categories in cancer research.

## 1.2 Research Objectives

The primary objective of this study is to explore and evaluate the use of word embedding techniques in capturing functional relatedness in an LBD setting within the biomedical domain, particularly focusing on cancer-related research. Specifically, this study aims to:

- **Assessment of Functional Relatedness Across Key Biomedical Categories**: The research aims to evaluate the functional relatedness between biomedical categories, including disease-gene, gene-chemical, and disease-chemical pairs.
- **Comparative Analysis of Word Embedding Models**: A comparative study will be conducted to analyze the performance of various word embedding models such as GloVe, BERT variations, and Word2Vec. The focus will be on their efficacy in creating embeddings that accurately capture functional relatedness.
- **Exploring Functional Relatedness with Word Embeddings in a Literature-Based Discovery Context:** This objective involves utilizing word embedding techniques to identify previously unknown functional relations; i.e., functional relations within the medical literature up until 2022 that were not yet part of the curated CTD version 2022. The goal is to assess whether these embeddings can perform LBD.

## 1.3 Solution Approach

To address the research objectives, a systematic solution approach was developed, incorporating advanced text processing and word embedding techniquesThe dataset, which consisted of PubMed abstracts, was preprocessed using PubTator [18][19] to extract relevant entities, including diseases, chemicals, and genes. Word embedding models, including Word2Vec, GloVe, BioBERT, and PubMedBERT, were then trained on this processed corpus to capture functional relationships among these entities [20]. The word embeddings' effectiveness was quantified by calculating cosine similarity for gene-

chemical, chemical-disease, and disease-gene relations, using the CTD as a reference [10]. To evaluate which model best captured functional relatedness, the curated CTD pairs were validated against the instance vector pairs of all CTD entries, using cosine similarity thresholds of 0.6, 0.7, and 0.8. Precision and recall were calculated, and heatmaps were used to visualize the performance of each model across the different thresholds to identify which model performed best at each level.

For BERT-based models, word embeddings were saved from different layers, including the summation of the last four hidden layers and the individual embeddings of each of the last four layers. Cosine similarity was calculated for the embeddings across these layers, and the average similarity between the layers was calculated. The results were plotted on a bar graph to determine which layers performed best for evaluating functional relatedness.

Additionally, the study explored whether the word embeddings could capture functional relatedness within the newly introduced 2024 CTD dataset. Only the new pairs present in CTD 2024, but not in previous versions, were filtered. Cosine similarity was then calculated for these new pairs using the previously generated word embeddings.

## 1.4 Contribution

This research makes several significant contributions to the field of biomedical text mining and knowledge discovery:

- **Evaluation of Word Embedding Models**: This thesis systematically evaluates different word embedding models, focusing on their performance in biomedical data analysis. The comparison helps in selecting the appropriate models for specific biomedical natural language processing tasks.
- **Identification of Functional Relationships**: The use of word embeddings in this research has enabled the identification of functional relationships between genes, diseases, and chemicals. These findings contribute to bridging gaps in biomedical knowledge and facilitate further exploration in fields such as cancer research.

- **Contribution to LBD**: This research contributes to the field of LBD by demonstrating how word embedding models can be used to process large-scale biomedical literature and identify previously unrecognized connections between biological entities.
- **Discovery of New Biomedical Associations**: The use of the latest CTD dataset has allowed the identification of new biomedical relationships. This capability supports a deeper understanding of biological interactions and advances medical research and diagnostics.

Through these contributions, the research advances the understanding and application of word embedding techniques in the biomedical domain, particularly in cancer research, enhancing the ability to capture and analyze the functional relationships among diverse biomedical concepts.

## 1.5 Organization

This thesis is organized into five chapters, each addressing different aspects of the research. Chapter 1 provides an introduction, outlining the research problem, objectives, and significance of the study, and introduces key concepts of word embeddings and their applications in the biomedical domain, particularly in cancer research. Chapter 2 covers the background and literature survey, reviewing existing studies on word embeddings, their use in biomedical text mining, and their applications in cancer research, including models such as Word2Vec, GloVe, BioBERT, and PubMedBERT. Chapter 3 details the methodology and experiments, including the development of a text processing pipeline, the training of various word embedding models, and the methods used to quantify functional relatedness and validate the results using the CTD. Chapter 4 presents the results of the functional relatedness quantification and validation, comparing the performance of different word embedding models and discussing the findings in the context of known associations between biomedical concepts. Chapter 5 concludes the thesis with a summary of key findings, their contributions to the field, and potential directions for future research, including exploring other embedding models and broader biomedical applications

# 2. Background and Related works

This chapter covers the necessary background and literature related to the research presented in this thesis. It starts with an introduction to key concepts such as Literature-Based Discovery (LBD), Medical Subject Headings (MeSH), word embeddings, and the principle of functional relatedness. The discussion then extends to the various word embedding models utilized in this study, including detailed descriptions of each model's architecture and relevance to the biomedical domain. Additionally, the chapter outlines the methodologies applied to assess the performance of these models and the metrics, such as cosine similarity, employed to quantify the functional relatedness among biomedical entities. This foundation sets the stage for understanding how advanced computational techniques are applied to facilitate discoveries in biomedical research.

## 2.1 Literature-Based Discovery (LBD)

LBD is a method used to uncover hidden or previously unknown associations within vast corpora of scientific literature. This approach helps bridge gaps in scientific knowledge by capturing overall features and semantics, allowing researchers to identify novel connections that may not be immediately apparent [83][84]. The main purpose of LBD is to generate new ideas or hypothesis by linking concepts from different fields, potentially leading to new discoveries. It is especially useful in fields where vast amounts of scientific data are available but remain disconnected [96].

Techniques used in LBD include co-occurrence analysis, which identifies how often terms appear together in texts, and semantic similarity measures, which assess the relatedness of concepts based on their meanings [97]. Natural Language Processing (NLP) methods, including concept extraction and entity recognition, are also used to automatically detect important terms like genes, diseases, or chemicals [98]. By applying these techniques, we can uncover meaningful connections between terms and concepts across different studies, supporting interdisciplinary research and helping to advance scientific knowledge.

## 2.2 Medical Subject Headings (MeSH)

MeSH is a comprehensive controlled vocabulary used for indexing, cataloging, and searching for biomedical and health-related information. It covers a wide range of domains in the life sciences, including anatomy, diseases, drugs, and procedures, facilitating a systematic organization of biomedical literature [21]. It is developed and maintained by the National Library of Medicine (NLM) at the U.S. National Institutes of Health (NIH). It is used in various NLM databases, including PubMed, to index articles for easier retrieval and research [22].

Key Features of MeSH:

- **Hierarchy:** The terms are organized in a hierarchical structure that allows broad as well as specific searches. For example, the term "cardiovascular diseases" is a broader term under which specific diseases like "Heart Failure" are listed.
- **Descriptors:** Descriptors are utilized to index articles, facilitating the accurate retrieval of literature. Each descriptor is paired with a definition, a scope note, and occasionally historical context.
- **Qualifiers:** Also known as subheadings, these qualifiers describe specific aspects of a topic, such as "diagnosis," "genetics," and "epidemiology." They can be combined with descriptors to enable more refined and precise searching.
- **Supplementary Concept Records:** For chemicals, drugs, and other substances not included as descriptors, MeSH provides supplementary concept records with information similar to that of descriptors.
- **Entry Terms:** Synonyms, alternate forms, and related expressions that point to the preferred descriptor facilitate the indexing and retrieval process, ensuring that literature is accessible under consistent terms.

## 2.3 Word Embeddings

Word embeddings transform words into continuous vector representations in a high-dimensional space, effectively capturing linguistic features such as morphology, semantics, and context [1]. These techniques are particularly valuable in biomedical research, enabling the quantification of functional relatedness between concepts such as genes, diseases, and chemicals. By embedding these entities in a shared vector space, word embeddings facilitate the identification of semantic similarities and functional relationships, crucial for uncovering hidden connections within extensive biomedical datasets [3]. There are various methods to for generating word embeddings, each with its unique approach to capturing word relationships like Word2Vec which predicts either a word from its context (CBOW) or its context from the word (SkipGram)[38] , Fasttext which is an extention of Word2Vec that incorporates subword information [25] , GloVe creates word vectors based on the frequency of word co-occurrences in a corpus [2] and BERT generates contextual embeddings by analyzing both the left and right context of a word and many more [10].

The key features of word embeddings are:

- **Understanding Context:** Word embeddings help machines understand context in words. For instance, the same word could have different meanings in different situations, like "apple" in "apple fruit" versus "Apple company." Word embeddings are capable of capturing these nuances [39].
- **Capturing Semantics:** Words with similar meanings are placed closer together in the vector space. For example, "king" and "queen" would be near each other because they are semantically related, meaning they share a similar meaning or role, such as being royalty [3].
- **Processing Efficiency**: Representing words as vectors allows for more efficient computation [64].

There are various applications of word embeddings such as,

- **Information Retrieval:** Search engines use word embeddings to understand queries and documents, improving search results beyond simple keyword matching [42].
- **Named Entity Recognition:** Identifying proper nouns like names of people, organizations, or locations in texts is easier with word embeddings that recognize semantic roles [77].
- **Sentiment Analysis:** By understanding the context and semantics, machines can identify if a text expresses positive or negative sentiments [87].
- **Machine Translation:** Translation of words/sentences from one language to another.
- **Chatbots and AI Assistants:** To respond appropriately, these applications need to understand user queries, which word embeddings enable by capturing the meanings and intentions behind words.

Given the essential role word embeddings play in these applications, it's important to understand how they are generated and utilized.

**Practical Application of Word Embeddings:**

To understand the application and effectiveness of word embeddings in capturing semantic relationships between words, let us consider two sentences from different documents: “The Minister speaks to the media in Illinois” and ‘The president greets the press in Chicago” [24]. We can obtain word embeddings for these statements using the Word2vec model. A sample of word embeddings for sentence 1 can be seen in Table 1, and word embeddings for sentence 2 are provided in Table 2.

*Table 1: Word Embeddings for sentence "The Minister speaks to the media in Illinois”*

| | |
|---|---|
| Minister | [-0.13671875 -0.015197754 0.06640625 -0.16601562 -0.10986328 ……] |
| speaks | [-0.036865234 0.20605469 0.030395508 -0.20410156 0.010009766 ……] |
| the | [0.080078125 0.10498047 0.049804688 0.053466797 -0.06738281 ……] |

| | |
|---|---|
| media | [0.09765625 -0.009277344 -0.26757812 -0.28125 0.016113281 ……..] |
| in | [0.0703125 0.08691406 0.087890625 0.0625 0.06933594 -0.10888 .....] |
| Illinois | [-0.0028076172 -0.05517578 0.01373291 0.45898438 0.056884766 ……] |

*Table 2: Word Embeddings for sentence "The president greets the press in Chicago"*

| | |
|---|---|
| The | [0.080078125 0.10498047 0.049804688 0.053466797 -0.06738281…….] |
| President | [-0.0134887695 -0.12011719 0.14453125 0.028930664 -0.02319336…...] |
| greets | [-0.060791016 0.46289062 0.16503906 -0.26757812 -0.39453125 ……..] |
| press | [-0.021972656 0.16992188 -0.19726562 -0.24414062 0.036621094 ……] |
| in | [0.0703125 0.08691406 0.087890625 0.0625 0.06933594 -0.10888 .....] |
| Chicago | [-0.13476562 0.18164062 0.09326172 0.4140625 -0.13085938 …….] |

These vectors are generated by training the model on a text corpus, enabling it to learn a 300-dimensional space. Since it's difficult to derive insights directly from these numerical values, we reduce the dimensionality of the data for better visualization. The reduced-dimensionality vectors, as shown in Table 3, allow us to plot and analyze the data more effectively.

*Table 3: Comparison of dimensionality-reduced vector values*

| Word | X | Y | Word | X | Y |
|---|---|---|---|---|---|
| Minister | -15.343611 | -119.114159 | President | -3.853197 | -132.21804 |
| speaks | 32.972813 | -147.397461 | greets | 50.354527 | -145.97364 |
| media | 5.780852 | -161.051666 | press | 10.367796 | -149.97613 |
| Illinois | -209.49865 | 416.482422 | Chicago | -198.14723 | 416.511383 |

Looking at these vectors we can say that similar words are closer to each other.

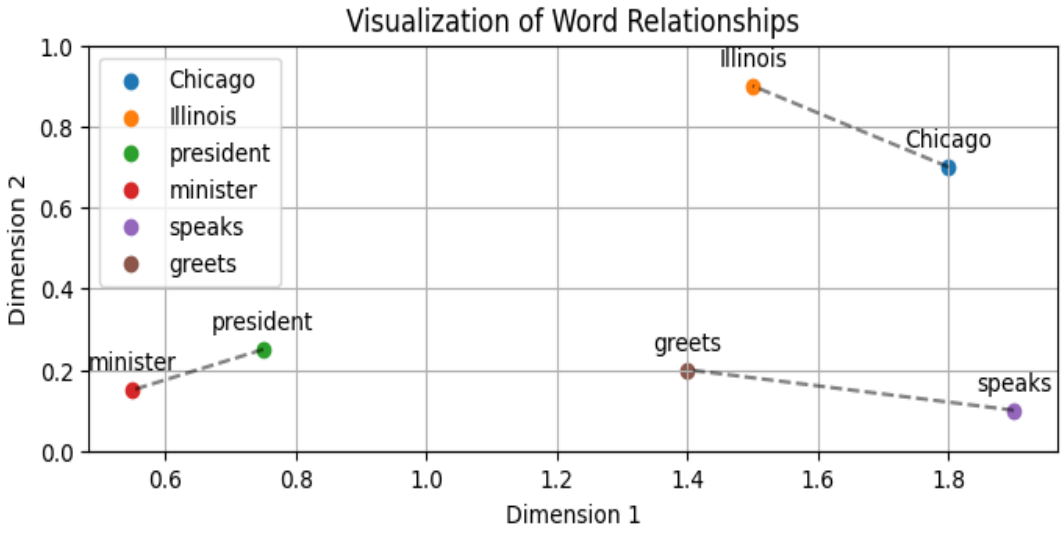


*Figure 1: t-SNE Visualization of Word Embeddings for Sample Terms [24]*

The Figure 1 is a visualization of these reduced-dimension vectors which confirms that similar words tend to be closer to each other in a vector space. This graph is a clear example of how word embeddings can be used to visualize and understand the semantic relationships between different terms in a corpus. The words "Illinois" and "Chicago" are closely plotted, reflecting their geographical relation. In contrast, "minister," "president," "speaks," "press," "media," and "greets" cluster together, indicative of their contextual linkage to themes of communication and leadership [24].

These semantic relationships are captured by various types of word embeddings, each employing a distinct approach. Word embeddings can be broadly categorized as follows:

a) Traditional Word Embeddings

- i) GloVe
- ii) Word2vec:
  - 1) SkipGram
  - 2) CBOW

b) Contextualized Word Embeddings

- i) BERT:
  - 1) BioBERT

2) PubMedBERT

## 2.4 Traditional Word Embeddings

Traditional word embeddings, also known as non-contextual embeddings, are created by analyzing large amounts of text to understand how words commonly appear together. These word embeddings represent the meaning of words based on their general usage across different texts [2]. However, they are context-independent, meaning the representation of a word stays the same no matter where or how it is used in a sentence. We have mainly used two types of traditional word embeddings.

### 2.4.1 GloVe

GloVe, which stands for Global Vectors is a word embedding model which was introduced by Pennington, Socher, and Manning in 2014 [2]. This embedding model constructs an explicit co-occurrence matrix using the entire corpus. The explicit co-occurrence matrix is nothing but how frequently each pair of words appears together within a specified context throughout the entire corpus. The matrix is "explicit" as it computes and stores the co-occurrence statistics before the training of word embeddings begins and captures co-occurrence information through prediction tasks without pre-computing a global co-occurrence structure as explained by Pennington and colleagues (2014) [2].

Let us consider the vocabulary size of the corpus as V, thus the co-occurrence is a square matrix of size V * V that contains how often each word pair appears together in the context of the corpus. The matrix is usually a large, sparse matrix in which both rows and columns represent unique words in the corpus. Each entry in the matrix is the frequency or probability of occurrence of two words within a certain distance this is defined by the context window size, and it is across the entire corpus. In this square matrix, rows represent individual words whereas, columns correspond to the context words. Each entry at position (i, j) in the matrix represents how strong the bond association between the word indicated by row i, and the context word indicated by column j. The main goal of GloVe is to modify

the vectors accordingly so that the similarity calculated between the context, and target pair words will closely align with the natural logarithm of their actual co-occurrence frequency in the corpus [2]. GloVe uses a loss function that calculates the difference between the predicted similarity of the words (based on the vectors) and the actual frequency of their co-occurrence.

GloVe uses gradient descent, a mathematical technique that iteratively minimizes the difference between predicted and actual word similarities, improving the accuracy of word vectors. By gradually adjusting the vectors, GloVe aligns the predicted similarities with the natural logarithm of their actual co-occurrence frequencies, effectively capturing both the semantic and syntactic relationships between words [2].

Here are the steps involved in extracting and utilizing GloVe embeddings for biomedical terms such as "bladder cancer" from pre-trained word vectors:

**Loading GloVe Embeddings:** We load the pre-trained GloVe embeddings are loaded from a file. The file contains pre-trained word vectors, with each word associated with a high-dimensional vector representation.

**Tokenization:** The next step involves tokenizing the input sentence into individual words using nltk's word tokenizer. In this case, we're specifically interested in extracting embeddings for the term "bladder cancer**".** The sentence "Recent studies have shown that patients whose bladder cancer exhibit overexpression of RB protein..." is tokenized into individual words. The words ['bladder', 'cancer'] are separated from the rest of the sentence to retrieve their embeddings [2].

**Filtering for Valid GloVe Words:** Once the text is tokenized, the system checks if each word is present in the pre-trained GloVe embeddings [2]. Words not found in the GloVe vocabulary are filtered out. Both "bladder**"** and "cancer" are present in the GloVe embeddings, so they are retained for embedding generation.

**Generating Embeddings:** For each valid word (in this case, "bladder" and "cancer"), the corresponding GloVe embedding is retrieved from the pre-trained embeddings.

- The GloVe embedding for "bladder" is retrieved, such as [0.1, -0.2, 0.5, ...].
- The GloVe embedding for "cancer" is retrieved, such as [0.05, -0.1, 0.3, ...].

**Averaging the Embeddings (Mean Pooling):** When a term consists of multiple words (e.g., "bladder cancer"), the embeddings for each word are averaged to form a single vector representing the entire term [2]. This ensures that the final embedding reflects the meaning of the combined words.
The embeddings for "bladder" and "cancer" are averaged to generate a single vector representing the term "bladder cancer." For example:

- "bladder" = [0.1, -0.2, 0.5]
- "cancer" = [0.05, -0.1, 0.3]

The averaged embedding would be:

$$\left[\frac{0.1 + 0.05}{2}, \frac{-0.2 + -0.1}{2}, \frac{0.5 + 0.3}{2}\right]$$

Resulting in:
[0.075, -0.15, 0.4].

**Storing the Embeddings:** Once the embeddings are generated, they are stored in a file containing only the word and its corresponding word embedding. Each entry in the file includes the phrase (e.g., "bladder cancer") and the numerical vector representing its embedding

**Pre-trained Models for GloVe**

- **glove.840B.300d:** This model, trained on a corpus of 840 billion tokens from Common Crawl, uses 300-dimensional vectors. It offers a broad vocabulary and rich semantic nuances due to its extensive training dataset [2].
- **glove.42B.300d:** Also trained on Common Crawl, but with 42 billion tokens, this 300-dimensional model provides a comprehensive understanding of language,

though with a slightly less diverse vocabulary compared to the 840 billion token model [2].

### 2.4.2 Word2vec

Word2vec is a word embedding technique introduced by Mikolov et al. in 2013 [27]. It transforms words into vector representations using neural networks. Word2vec is unique as it learns these word vectors by predicting words based on their neighboring words in a sentence, or vice versa [28]. These word embeddings capture both syntactic (grammatical) and semantic (meaning-based) relationships between words by analyzing the context in which they appear [10].

However, Word2vec is context-independent, meaning that each word is assigned a single, static vector, regardless of its context in different sentences. For example, the word "bank" would have the same vector representation in both "river bank" and "bank account," even though the meanings are different [26]. Despite this limitation, Word2vec effectively captures word relationships.

The performance of Word2vec can be evaluated through two main methods:

**1. Intrinsic Evaluation**: Directly tests the quality of word embeddings by assessing how well they capture linguistic properties.

- **Similarity and Relatedness**: Compares model-generated similarity scores between word pairs with human judgments [34].
- **Analogy Tasks**: Tests the model's ability to solve word analogies (e.g., "man" is to "woman" as "king" is to "queen") [34].
- **Clustering**: Groups words based on their vector representations and evaluates the coherence of these clusters.

**2. Extrinsic Evaluation**: Measures the impact of word embeddings on the performance of NLP tasks, determining their practical utility.

- **Text Classification**: Uses word embeddings as features for categorizing texts (e.g., spam detection, sentiment analysis) [16].
- **Named Entity Recognition (NER)**: Identifies entities like person names or organizations in text.

There are two types of Word2vec models i.e., CBOW and SkipGram. The steps below outline the complete process for generating embeddings using both the approaches,

**1.** Tokenization

The first step in the Word2Vec process is tokenizing the sentence into individual words or tokens, allowing the model to process the text at the word level [31].

- **Example Tokenized Words**:
  ["Recent", "studies", "have", "shown", "that", "patients", "whose", "bladder", "cancer", "exhibit", "overexpression", "of", "RB", "protein", "as", "measured", "by", "immunohistochemical", "analysis", "do", "equally", "poorly", "as", "those", "with", "loss", "of", "RB", "function"]

**2. Vocabulary Construction**

Once the text is tokenized, the model constructs a vocabulary by identifying each unique word and assigning it a unique index. This is crucial for building the mapping between words and their respective representations.

- **Vocabulary Indices:**
  - "Recent" = 1
  - "studies" = 2
  - "have" = 3
  - "shown" = 4
  - "that" = 5

- "those" = 23
- "with" = 24
- "loss" = 25
- "function" = 26

**3. One-Hot Encoding**

Each word in the vocabulary is represented as a one-hot encoded vector, where the index of the word is set to 1, and all other indices are set to 0.

- **One-Hot Encoding for "bladder":**
  - Index: 8
  - Vector: [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
- **One-Hot Encoding for "cancer":**
  - Index: 9
  - Vector: [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]

The steps outlined above are common to both Word2Vec models. Below, we detail the specific steps for the CBOW and Skip-gram models separately.

**CBOW:**

It is a popular Neural Network Language Model (NNLM) introduced by Mikolov et al. in 2013 [27]. It is trained to predict a target word based on its surrounding context. The context is defined as a window of words around the target word.

- **Example Window Size of 6 for "bladder":**
  - **Context Window:** ["have", "shown", "that", "patients", "whose", "bladder", "cancer", "exhibit"]

- **One-Hot Encoded Context Window:**
  - "have" = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
  - "shown" = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
  - "that" = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
  - "patients" = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
  - "whose" = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
  - "bladder" = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
  - "cancer" = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
  - "exhibit" = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

- **Matrix Representation:**

```
[ [0, 0, 1, 0, 0, 0, 0, 0, 0, 0], // have
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], // shown
  [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], // that
  [0, 0, 0, 0, 0, 1, 0, 0, 0, 0], // patients
  [0, 0, 0, 0, 0, 0, 1, 0, 0, 0], // whose
  [0, 0, 0, 0, 0, 0, 0, 1, 0, 0], // bladder
  [0, 0, 0, 0, 0, 0, 0, 0, 1, 0], // cancer
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]   // exhibit
]
```

**4. Neural Network Processing**

- **Input Layer:** Receives one-hot encoded vectors representing each word within the context window.
- **Hidden Layer:**
  - Computes weighted sums of the inputs.
  - Applies activation functions to capture non-linear relationships and feature representations.

- **Output Layer:**
    - The model predicts the target word (the center word) using the weighted context and applies an activation function (such as Softmax for classification). The embeddings are learned during this process, as the model is trained to minimize the difference between predicted and actual center words [1][39].

**SkipGram:**

SkipGram is a popular NNLM introduced by Mikolov et al. in 2013 as a counterpart to the CBOW model [33]. Unlike CBOW, which predicts a target word based on its context, SkipGram does the reverse; it uses the target word to predict the surrounding context words. This model is particularly useful for embedding rare words within the text [33].

**Target Word: "Recent" (Index 1)**

**Context Words**: ["studies", "have", "shown", "that", "bladder"]

**Context Word Pairs**:

- **("Recent", "studies")**
    - Target (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
- **("Recent", "have")**
    - Target (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 1, 0, 0, 0, 0, 0, 0]
- **("Recent", "shown")**
    - Target (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 1, 0, 0, 0, 0, 0]
- **("Recent", "that")**
    - Target (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 0, 1, 0, 0, 0, 0]

- **("Recent", "bladder")**
    - Target (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 0, 0, 1, 0, 0, 0]

**Target Word: "studies" (Index 2)**

**Context Words**: ["Recent", "have", "shown", "that", "bladder", "cancer"]

**Context Word Pairs**:

- **("studies", "Recent")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [1, 0, 0, 0, 0, 0, 0, 0, 0]
- **("studies", "have")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 1, 0, 0, 0, 0, 0, 0]
- **("studies", "shown")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 1, 0, 0, 0, 0, 0]
- **("studies", "that")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 0, 1, 0, 0, 0, 0]
- **("studies", "bladder")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 0, 0, 1, 0, 0, 0]
- **("studies", "cancer")**
    - Target (One-Hot Vector): [0, 1, 0, 0, 0, 0, 0, 0, 0]
    - Context (One-Hot Vector): [0, 0, 0, 0, 0, 0, 1, 0, 0] ……..

**Target Word: "aggressive" (Index 9)**

**Context Words**: ["bladder", "cancer", "is"]

**Context Word Pairs**:

- **("aggressive", "bladder")**
  - Target (One-Hot Vector): [0, 0, 0, 0, 0, 0, 0, 0, 1]
  - Context (One-Hot Vector): [0, 0, 0, 0, 0, 1, 0, 0, 0]
- **("aggressive", "cancer")**
  - Target (One-Hot Vector): [0, 0, 0, 0, 0, 0, 0, 0, 1]
  - Context (One-Hot Vector): [0, 0, 0, 0, 0, 0, 1, 0, 0]
- **("aggressive", "is")**
  - Target (One-Hot Vector): [0, 0, 0, 0, 0, 0, 0, 0, 1]
  - Context (One-Hot Vector): [0, 0, 0, 0, 0, 0, 0, 1, 0]

**4.Neural Network Processing**

- **Input Layer**: Receives the one-hot encoded vector of the target word.
- **Hidden Layer**: Converts one-hot encoded vectors into dense representations using the embedding matrix.
- **Output Layer**:
  - The model predicts the context words based on the target word by generating probability distributions for each context word in the vocabulary using a softmax function.
  - The embeddings are learned during this process, where the target word predicts the surrounding context words [39].

**Embeddings Formation**:

- The word embeddings are learned in the hidden layer's weight matrix (embedding matrix).

- After training, the learned embeddings can be extracted from the rows of this matrix, representing each word in a dense, lower-dimensional space.

This vector is multiplied by a weight matrix to produce a dense embedding for the target word. The output layer then uses the SoftMax function to predict the context words by generating a probability distribution over the entire vocabulary for each position within the context window [33].

**Hyperparameters for Word2Vec (SkipGram and CBOW):**

- **Vector Size:** Different dimensions for word vectors were tested, including 250, 300, 500 and 700. The choice of vector size significantly affects the model's ability to capture detailed semantic relationships. Larger sizes offer more expressive power but increase computational demands [1].
- **Window:** Context window sizes of 6, 10, 15 and 20 were evaluated. This parameter determines the number of words surrounding the target word considered during context prediction. A larger window size enhances the model's ability to capture broader context, aiding in understanding long-range dependencies in medical texts [1].
- **Min Count:** This was set to 1, ensuring that all words, even those appearing only once, are included in the training process. This is crucial in biomedicine, where rare terms can have significant meanings [73] [88].
- **SG:** The training algorithm included options for both 0 and 1. Setting sg to 0 employs the CBOW (Continuous Bag of Words) model, which uses the context of a word to predict the word itself by averaging the vectors of the context words. Conversely, setting sg to 1 uses the Skip-gram model, where the model predicts surrounding context words from the target word. This often results in better performance with rare words or phrases [1].

Word2Vec model, through its CBOW and SkipGram architectures, provides a powerful method for capturing the semantic relationships between biomedical terms in our dataset.

Both architectures, CBOW and Skip-gram, offer flexibility in how contextual relationships are modeled, with Skip-gram being especially useful for handling rare terms often encountered in specialized biomedical texts.

## 2.5 Contextualized Embeddings

Word embeddings which are context-dependent are called contextualized word embeddings. This approach offers an understanding of the words meaning based on how it is used or its context. For example: Given the word “bank”, if a contextualized embedding model is used, the model generates two different embedding vectors based on its usage and meaning. If the sentence is related to finance, then the word “bank” would have vectors close to “finance" and "money" [42]. On the other hand, when "bank" is used in the context of a river, its vector would be closer to words like "river" and "water". This adaptability allows for a richer representation of word meanings and relationships.

These word embeddings offer different representations of words each time they appear based on surrounding text, unlike traditional word embeddings like GloVe and Word2vec where there is a single, static word representation irrespective of their usage. This is a much richer and more nuanced approach to forming word embeddings, capturing even minute nuances and context-dependent meanings. For generating our word embeddings, we have utilized BERT, BioBERT, and PubMedBERT.

### 2.5.1 BERT

BERT is a model based on the transformer architecture that captures contextual word representations by considering both the left and right contexts of a word simultaneously [45]. Unlike previous models that often-encoded text in a unidirectional manner, BERT’s bidirectional approach allows it to capture richer semantic representations by pre-training on large, unlabeled text corpora.

The Transformer architecture, introduced by Vaswani et al. in 2017 [32], utilizes a parallelized approach that allows for more efficient training compared to models that use recurrent neural networks (RNNs) or convolutional neural networks (CNNs). This efficiency is achieved by processing data simultaneously across multiple GPUs or TPUs.

The Transformer model is composed of stacked encoders and decoders, each with multiple identical layers. The encoder maps an input sequence to continuous representations using a stack of six identical layers, each containing a multi-head self-attention mechanism and a fully connected feed-forward network, followed by layer normalization. Similarly, the decoder generates an output sequence, adding a third sub-layer to perform multi-head attention over the encoder's output.

Some widely used concepts in BERT are:

- **Self-Attention Mechanism:** This mechanism calculates attention scores for each word in the input sequence relative to other words, allowing the model to weigh the influence of surrounding words when encoding a particular word. The attention score is calculated using query (Q), key (K), and value (V) vectors, derived from input embeddings:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{dk})V$$

- **Multi-Head Attention:** Running several attention mechanisms in parallel allows the model to capture different aspects of the input sequence simultaneously. Each head captures unique features, and their outputs are concatenated and linearly transformed.
- **Feed-Forward Networks:** Each Transformer layer includes a position-wise feed-forward network with ReLU activation, enabling non-linear transformations of the representations.

- **Layer Normalization and Residual Connections:** These techniques are used around each sub-layer to reduce gradient problems and promote smooth gradient flow.

The phases in which BERT operates are as follows:

1. **Pre-training Phase**: BERT is trained on a large corpus using two strategies:
    - **Masked Language Model (MLM):** Randomly masks some input tokens and predicts them based on the context, requiring the model to consider both left and right contexts.
    - **Next Sentence Prediction (NSP):** Trains BERT to predict whether one sentence follows another in a document, helping it capture relationships between sentences.
2. **Fine-tuning Phase**: After pre-training, BERT is fine-tuned on smaller, task-specific datasets. Minimal task-specific layers are added, and the model's weights are slightly adjusted to optimize performance for tasks like question answering or sentiment analysis.

By combining a comprehensive pre-training phase with fine-tuning, BERT effectively captures deep semantic relationships and context-dependent meanings, representing a significant advancement in NLP.

Building on this foundation, we have utilized pretrained models like BioBERT and PubMedBERT in our research, which are specifically designed for the biomedical domain. These models inherit BERT's architecture but are further trained on large biomedical corpora, enabling them to capture the complex and specialized language used in biomedical texts.

This procedure focuses on extracting sentence-level embeddings from BioBERT and PubMedBERT using the mean pooling technique. The method involves averaging the

embeddings across all tokens in the sequence to generate a comprehensive sentence representation. Steps followed for the creation of word embeddings of BERT model are as follows:

**Tokenization:** The first step involves converting raw text into tokens using the WordPiece tokenizer**,** which splits the input text into subword tokens and adds special tokens Classification token[CLS] at the beginning and Separator token [SEP] at the end to mark the sequence boundaries. For example, the sentence "blood cancer treatment" is tokenized into ['blood', 'cancer', 'treat', '##ment'], with [CLS] and [SEP] signaling the start and end of the sequence [42].

**Input Representation:** Once tokenized, input embeddings are created for each token by combining two components:

1. **Token Embeddings**: Vector representations of each token.
2. **Positional Embeddings**: Indicate the position of each token in the sequence to help the model understand word order [10].

**Segment Embeddings**: Distinguish between different segments of text, though they are less relevant for single-sentence tasks. For each token, the token embeddings**,** positional embeddings**,** and segment embeddings are combined. For example, the tokenized sequence ['[CLS]', 'blood', 'cancer', 'treat', '##ment', '[SEP]'] is converted into corresponding input embeddings [10][11].

**Passing Through Transformer Layers:** The input embeddings are then passed through multiple transformer layers in BioBERT or PubMedBERT. These layers apply self-attention mechanisms to help the model learn the relationships between tokens in the sequence. The bidirectional attention allows the model to capture how each token relates to every other token [42].

**Extracting Embeddings from the Output Layer:** After the sequence passes through all the transformer layers, the model produces output embeddings for each token. To generate

sentence-level embeddings, we use the mean pooling technique, which computes the average of all token embeddings across the sequence [11].

In this process, the embeddings from the last four layers of BioBERT are utilized in two distinct approaches:

1. **Summation of the Last Four Layers**:
    - The hidden states from the last four layers (Layer -4 to Layer -1) are summed for each token. This summation combines information from multiple layers, resulting in a single, enriched token embedding that captures deeper context and task-specific insights.
    - Summing the layers leverages the refined features from these final stages, providing a comprehensive token representation.
2. **Individual Extraction from Each Layer**:
    - The embeddings from each of the last four layers are also extracted individually, allowing us to analyze the token representations layer by layer.
    - Extracting individual layers offers a detailed view of how the model's understanding of each token evolves across layers.

By using both summation and individual layer extraction, we capture a rich, context-aware representation while also enabling a more granular analysis of how embeddings are formed at each stage.

**Mean Pooling:** Once the embeddings are obtained, the sum of the token embeddings is averaged across all tokens in the sequence to generate a single sentence-level embedding. This ensures that the final representation takes into account the entire context of the sentence [42].

**Storing the Embeddings:** After extracting the sentence-level embeddings, they are stored for future use. The resulting sentence embeddings are saved to a file, with each embedding linked to the corresponding phrase or sentence from which it was generated.

**BioBERT:**

BioBERT is a specialized version of the BERT model tailored for the biomedical domain. It is pre-trained on an extensive corpus of biomedical literature, including PubMed abstracts and PMC full-text articles, which enhances its ability to understand the complex language and terminology unique to the biomedical field [43].

BioBERT has greatly enhanced biomedical NLP by boosting performance in tasks such as information extraction, retrieval, and the comprehension of biomedical texts [47]. It maintains architecture of BERT and can be fine-tuned for specialized tasks within the biomedical field, including disease name recognition, chemical and gene entity extraction, and relation extraction between biomedical entities.

**Pre-trained Model for BioBERT used to create word embeddings:**

- **BioBERT Model (dmis-lab/biobert-v1.1):** This is a BERT variant specifically fine-tuned on biomedical corpora. The model features a hidden layer size (vector size) of 768, which is standard for the BERT base architecture. It comprises 12 transformer layers, each with 12 attention heads, facilitating deep semantic understanding tailored to biomedical contexts [10].
- **BioBERT Tokenizer:** Using the BertTokenizer from the transformers library, configured with the dmis-lab/biobert-v1.1 model, this tokenizer is adept at processing biomedical text consistent with BioBERT's training. It handles tokenization by segmenting text into tokens that BioBERT has been trained on, including the insertion of special tokens such as [CLS] and [SEP] necessary for BERT's operation [10].

Overall, BioBERT has made substantial contributions to biomedical research, clinical decision-making, and scientific discoveries by providing a deeper understanding of biomedical texts [43].

**PubMedBERT:**

PubMedBERT is a specialized variant of the BERT model, exclusively pre-trained on a vast collection of biomedical literature from the PubMed database [44]. Unlike BioBERT, which is initially pre-trained on general English texts before being fine-tuned on biomedical data, this model is trained solely on biomedical texts from the beginning. This focused training enhances its ability to understand the unique syntax, terminology, and knowledge within the biomedical field. As Gu and colleagues (2020) highlight, this specialized approach enables it to more effectively capture the specific language and concepts of the biomedical domain [44].

**Pre-trained Model for PubMedBERT:**

- **PubMedBERT Model (microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext):** This is a variant of BERT adapted for the biomedical domain, trained extensively on abstracts and full texts from the PubMed database. It is an uncased model, meaning it treats text without distinguishing between uppercase and lowercase letters. The model features a hidden layer size (vector size) of 768, with 12 layers of transformers, each containing 12 attention heads. This configuration allows the model to capture complex, multi-level semantic relationships inherent in biomedical literature [11].
- **PubMedBERT Tokenizer:** Utilizing the AutoTokenizer from the transformers library and configured with the microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext model, the tokenizer is specifically designed to process biomedical text in alignment with the training specifics of PubMedBERT. This includes handling the uncased nature of the text and embedding special tokens such as [CLS] at the beginning and [SEP] at the end of sequences for BERT's processing needs. The tokenizer ensures that the input text is appropriately segmented into tokens that the model has been trained on, preserving the integrity of the input for optimal embedding extraction [11].

The success of PubMedBERT in various biomedical NLP tasks can be attributed to its specialized training, as highlighted by Gu and his team (2020) [14].

**Example of Word Embedding Creation Process**

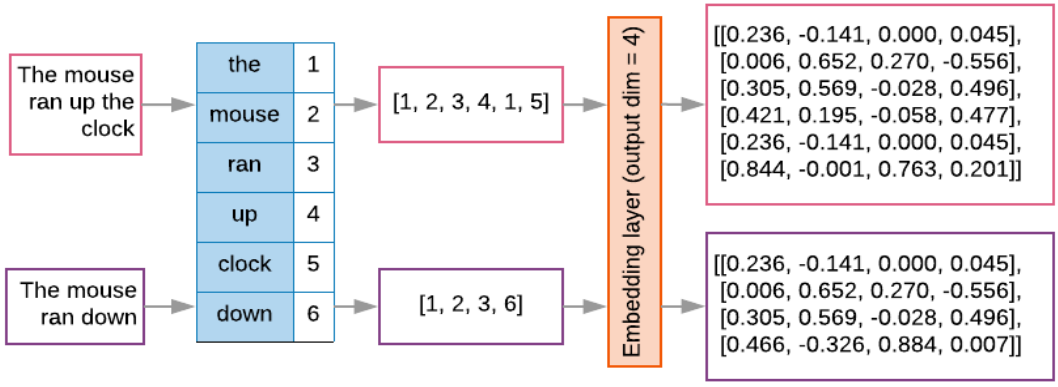


*Figure 2:Process of word embedding creation [70]*

Figure 2 illustrates the process of transforming text into word embeddings through an embedding layer in a neural network. The process begins by tokenizing sentences like "The mouse ran up the clock" and "The mouse ran down," where each word is assigned a unique index based on the vocabulary (e.g., "the" -> 1, "mouse" -> 2, etc.). These tokenized sentences are then represented as sequences of indices [1, 2, 3, 4, 1, 5] for the first sentence and [1, 2, 3, 6] for the second. These sequences are subsequently fed into an embedding layer, which converts each word index into a fixed-dimensional vector, with an output dimension of 4 in this case. Each word index is mapped to a specific vector in the embedding space, capturing the semantic meaning of the words. For example, "the" (index 1) is mapped to [0.236, -0.141, 0.000, 0.045], "mouse" (index 2) to [0.006, 0.652, 0.270, -0.556], and so on. As a result, the token sequences are converted into sequences of embedding vectors, which represent the original sentences in a numerical form suitable for input into machine learning models. This transformation allows the model to process the semantic information contained within the text effectively.

**Empowering Literature-Based Discovery with Advanced Word Embedding Techniques**

Word embeddings are pivotal in LBD, as they are instrumental in identifying previously unrecognized relationships within biomedical texts. These models utilize deep semantic learning from vast natural language corpora, as originally conceptualized by Mikolov et al. (2013), and are adapted in the biomedical domain through specialized versions of BERT tailored for scientific texts [23][33].

**Facilitating Non-Obvious Connections Through Word Embeddings:**

1. **Detecting Hidden Links:** Word embeddings facilitate LBD by clustering related concepts, enhancing the discovery of connections that may span across disparate research articles. For instance, terms like "EGFR" (Epidermal Growth Factor Receptor) and "Lung Cancer" can be closely linked within the embeddings, potentially unveiling novel associations [85].
2. **Bridging Cross-Domain Gaps:** By integrating diverse biomedical data, word embeddings enable researchers to connect chemical data with genetic information, supporting comprehensive cross-domain research [86].
3. **Revealing Deeper Associations:** Beyond recognizing direct synonyms, word embeddings can expose more profound associations between related but not explicitly synonymous terms across various biomedical fields [86].
4. **Understanding Contextual Semantics:** Word embeddings capture the subtle semantic nuances of biomedical terminology, ensuring that terms such as "cancer" are effectively associated with related concepts like "oncology" and "chemotherapy," reflecting their contextual meanings [43].

Through these capabilities, word embeddings significantly enhance the effectiveness of LBD, providing a robust tool for advancing biomedical research and facilitating the discovery of potential therapeutic targets.

## 2.6 Functional Relatedness

Functional relatedness in word embeddings refers to the ability to capture relationships between concepts based on their roles or functions within a particular domain [1]. In the context of biomedical research, functional relatedness describes how entities such as genes, diseases, or chemicals are connected by their biological functions or interactions [2]. For example, the BRCA1 gene is functionally related to breast cancer, as mutations in this gene increase the risk of developing the disease. Similarly, the chemical aspirin is functionally related to inflammation because it inhibits the COX-1 enzyme, reducing pain and inflammation.

Word embeddings represent words as continuous vectors in a high-dimensional space, where words or concepts with similar meanings or roles are positioned closer to each other. Functional relatedness can be measured by examining the similarity between these vectors. For instance, in biomedical word embeddings, if a gene and a disease have a functional relationship (e.g., a gene is linked to causing a disease), their vectors will be closer in the embedding space [42].

- **Semantic Similarity**: This refers to the similarity between terms based on their meaning or usage in a given context. For example, the words "tumor" and "cancer" will appear close in a vector space because of their related meanings [13].
- **Functional Similarity**: This refers to the interaction between different entities within a biological system. For example, functional relatedness can describe how a gene regulates another gene or how a drug targets a disease. Word embeddings are useful for capturing these functional connections across large biomedical datasets [1].

### 2.6.1 Measuring Functional Relatedness in Word Embeddings

Understanding functional relatedness is essential for accurate biomedical research. For example, a gene might be functionally related to a chemical if the chemical interacts with or influences the gene's activity in a biological process. Capturing these relationships in word embeddings can help predict how chemicals might affect certain genes, which is vital for drug development and understanding disease mechanisms. To assess these relationships, several common methods are used to measure functional relatedness:

1. **Cosine Similarity**: Measures the angle between two vectors, showing how closely related two entities are in terms of their interactions [51].
2. **Euclidean Distance (L2 Distance)**: Measures the straight-line distance between two points. It helps determine how far apart entities are in their functional roles, often used in clustering tasks [52].
3. **Manhattan Distance (L1 Distance)**: Adds up the absolute differences in positions to measure distance, useful for understanding how different or similar two entities are in their functions [52].
4. **Pearson Correlation Coefficient**: Measures how changes in one entity relate to changes in another [53].

We chose cosine similarity over other metrics like Euclidean and Manhattan distances because these methods focus more on the relationships and patterns between entities rather than just measuring distance**.** Cosine similarity tells us how similar two entities are in direction, which is more meaningful for understanding how they relate in high-dimensional spaces. While Pearson correlation helps identify linear relationships, making it ideal for detecting how one entity might influence another. On the other hand**,** Euclidean and

Manhattan distance**s** simply measure the distance between points, which doesn't capture the relational aspects we are most interested in.

**Cosine Similarity**

It is a widely used mathematical metric that determines how similar two vectors are, regardless of their size. It is particularly useful for capturing functional relatedness because it focuses on the orientation of vectors in a high-dimensional space rather than their magnitude. Cosine similarity is commonly used in NLP for comparing word embeddings, offering a way to measure the similarity between entities based on how they interact, rather than just how frequently they occur together [51].

The computation of cosine similarity, is done by dividing the dot product of two vectors by the product of their magnitudes. This is also known as normalization. Normalization allows cosine similarity to capture the direction in which entities interact. The value of cosine similarity ranges from [-1 to 1], with specific interpretations for these values:

- **-1**: The vectors are pointing in opposite directions, indicating a negative functional relationship.
- **0**: The vectors are perpendicular, meaning there is no functional relationship between the entities.
- **1**: The vectors point in the same direction, indicating a strong positive functional relationship.

As it emphasizes the direction of interactions rather than their size it is computationally efficient and effective in capturing relationship which makes it a valuable tool in various biomedical and NLP tasks. Additionally, by focusing on direction rather than magnitude, cosine similarity ensures that even when entities interact differently in terms of frequency, their functional connection can still be accurately assessed.

This chapter provided a foundational overview of word embeddings and their application in biomedical research, focusing on how they transform words into vectors to capture

relationships between entities like genes, diseases, and chemicals, which is crucial for tasks like LBD. We explored traditional and contextualized embedding models, such as SkipGram, CBOW, GloVe, BERT, BioBERT, and PubMedBERT, and discussed their roles in understanding functional relatedness. Additionally, we examined the use of cosine similarity for calculating functional relatedness. This background sets the stage for the detailed analysis and experiments in the following chapters.

## 3. Methodology

In this study, we utilize various word embedding techniques to capture the functional relatedness between genes, chemicals, and diseases by analyzing a large corpus of cancer-related medical texts. Our objective is to determine how these word embedding methods can reveal the functional relationships among biomedical concepts within the data, without the need for direct human intervention. To achieve these objectives, our methodology is structured into four distinct phases that collectively address the research tasks of processing the data, creating meaningful word embeddings, visualizing the results, and exploring the functional relationships among biomedical concepts.

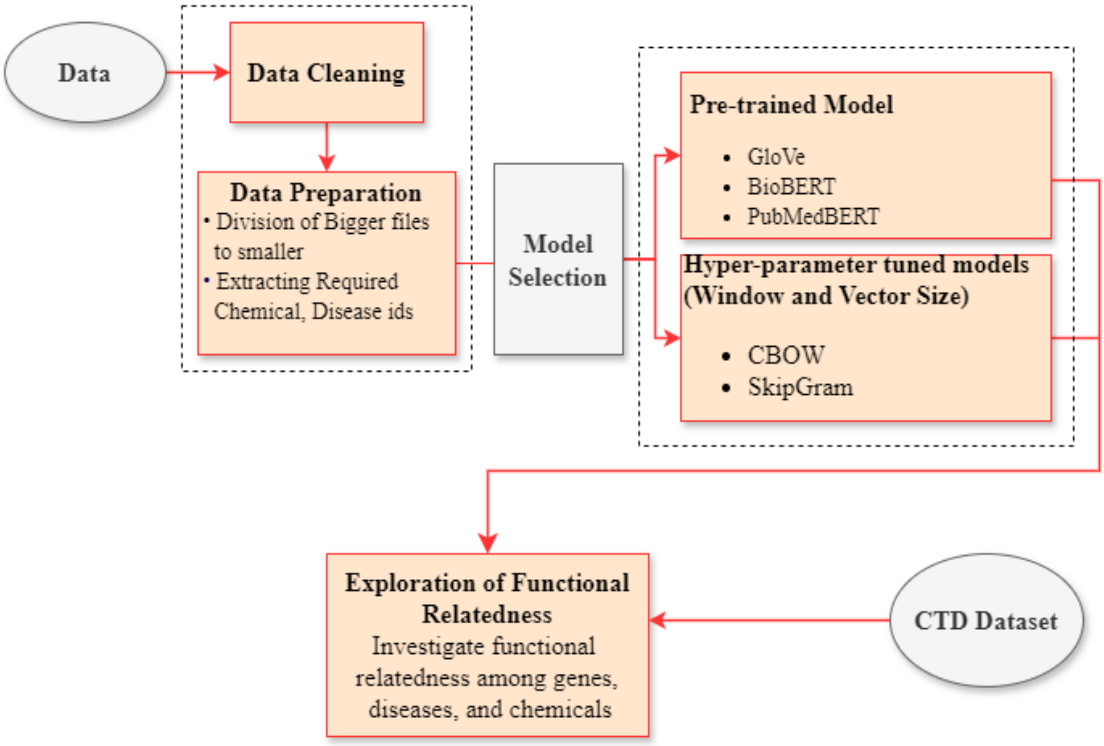


Figure 3 : Overview of research methodology

The workflow of our methodology is outlined in Figure 3 and consists of several key stages designed to systematically explore the functional relatedness of genes, chemicals, and diseases in cancer research. This approach integrates principles from LBD**,** which seeks to

uncover hidden or previously unknown relationships among scientific concepts through the analysis of large-scale literature.

The methodology can be described as follows:

1. **Data Preparation:** The data consists of PubMed abstracts, consisting of both the textual content of the articles and the associated biomedical annotations, providing a detailed foundation for exploring the semantics and relationships of medical terms, particularly those related to cancer [71]. This stage is focused on obtaining and organizing the data. We segment extensive datasets into smaller, more manageable parts and selectively extract relevant chemical, disease, and gene information along with their identifiers to create data dictionaries.

2. **Formation of Word Embeddings:** Following data preparation, the next step involves generating word embeddings using various models like CBOW, SkipGram, GloVe, BioBERT and PubMedBERT.
3. **Exploration of Functional Relatedness:** In the final phase of our study,
    a. We evaluated the models' ability to predict associations between genes, diseases, and chemicals by calculating precision and recall metrics. This was done by comparing the cosine similarity scores against thresholds established for curated CTD pairs and newly identified instance vector pairs of instances from CTD.
    b. We assessed whether the word embeddings generated from PubMed abstracts up to 2022 could capture the functional relatedness of new curated pairs from the CTD August 2024 release. This evaluation involved filtering only the newly curated pairs from CTD 2024 and determining if the word embeddings we created could capture the functional relationships between these new pairs of genes, diseases, and chemicals.

Overall, the methodology incorporates principles of LBD to uncover previously unknown relationships between genes, chemicals, and diseases in cancer research. By processing

PubMed abstracts and applying models like CBOW, SkipGram, GloVe, BioBERT, and PubMedBERT, the workflow enables the identification of functional relationships.

## 3.1 Dataset Overview and Concept Annotation

The dataset consists of 100,000 cancer-related abstracts sourced from PubMed, spanning from January 1976 to December 2022. Abstracts that were not in English, review articles, and those without full text availability were excluded. The biomedical concepts within the text were annotated using the PubTator NLP tool, which identified and tagged specific entities and relationships such as genes, diseases, chemicals, and proteins. This annotation process structured the unstructured text, facilitating more efficient analysis and extraction of meaningful information. [72].

The dataset is formatted in JSON, which includes the PubMed ID (PMID), the article text, and a collection of annotated concepts. Each concept is accompanied by an identifier, a type (e.g., 'Disease', 'Chemical', or 'Gene'), and the term itself for example, the term could be “lung carcinoma” or “BRCA1”. The concept_type field is particularly important for our study, as it allows us to filter and generate embeddings for only diseases, chemicals, and genes.

To illustrate the annotation process, we will now present a sample abstract. This example will demonstrate how biomedical concepts are identified and categorized within the text.

**Sample Abstract:**

"Recent studies have shown that patients whose bladder cancer exhibit overexpression of RB protein as measured by immunohistochemical analysis do equally poorly as those with loss of RB function."

**Annotated Abstract:**


{

"PMID": "10022125",

"ARTICLE": {

"TEXT": "Recent studies have shown that patients whose bladder cancer exhibit overexpression of RB protein as measured by immunohistochemical analysis do equally poorly as those with loss of RB function.",

"CONCEPTS": [

{

"IDENTIFIER": "9606",

"CONCEPT_TYPE": "Species",

"TERM": "patients"

},

{

"IDENTIFIER": "MESH:D001749",

"CONCEPT_TYPE": "Disease",

"TERM": "bladder cancer"

}]}}


In this example, the abstract has been annotated to mark specific biomedical concepts, such as "bladder cancer" (Disease) and "patients" (Species). This JSON format ensures that a medical concept, whether a disease chemical or gene is tagged accurately and associated with its occurrence within the article text [72].

Table 4 provides a detailed breakdown of the key elements within our structured dataset. Below are the key elements from the structured dataset:

*Table 4: Description of various key names of the dataset*

| Key Term | Description |
|---|---|
| **PMID** | PubMed ID, the unique identifier for each article in PubMed |
| **ARTICLE** | Container for the article's content |
| **TEXT** | The body text of the article |
| **CONCEPTS** | An array of annotated biomedical concepts found in the text |
| **IDENTIFIER** | The unique ID for each annotated concept (e.g., MeSH ID) |
| **CONCEPT_TYPE** | The type of concept, such as 'Disease' or 'Chemical' or others |
| **TERM** | The specific term used in the article text for the concept |

Each key is paired with a corresponding value within the JSON structured document. An example is shown in Table 5.

*Table 5: Representation of key-value pairs*

| Key | Value | | |
|---|---|---|---|
| PMID | 1000510 | | |
| ARTICLE.TEXT | In Sarcoma 180 and L1210 ascites tumor models, the initial rate of methotrexate accumulation... | | |
| Concept | IDENTIFIER | CONCEPT_TYPE | TERM |
| | MESH: D012509 | Disease | Sarcoma |

| | | | |
|---|---|---|---|
| | MESH: D007939 | Disease | L1210 ascites tumor |
| | MESH:D008727 | Chemical | methotrexate |
| | CVCL_0382;NCBITaxID:10090 | CellLine | L1210 |

## 3.2 Data Cleaning and Preparation

After obtaining a structured JSON dataset, pre-processing is necessary to enhance its relevance for further analysis, specifically in the context of cancer research. We implemented a filtering strategy to selectively extract and retain only the required information pertaining to diseases and chemicals of interest.

1. **Division of Large JSON Files:** Running PubTator resulted in 10 JSON files. Since these files were still quite large, we divided them into smaller, more manageable segments for easier processing. This segmentation was necessary because processing a single large file with all abstracts turned out to be too resource-intensive. Depending on the original file size, each large file was split into two or three parts or segments. A consistent naming convention was employed to reflect the subdivision, such as 'proper_pubtator_1_1', where the first number indicates the file, and the second denotes the segment.
2. **Creation of Data dictionaries:** A separate data dictionaries were created to focus only on cancer related terms. MeSH IDs and JSON dataset was used to create data dictionaries to extract relevant information related to cancer or neoplasms and exclude unrelated categories [72].

   **Selection Criteria for Diseases and Chemicals:** For diseases, we focused on the "Neoplasms by Site" category within the MeSH hierarchy, which includes cancer-related terms and is represented by MeSH codes starting with C04_588. All

descendants under this category were selected, resulting in 232 unique IDs. For chemicals, all categories were considered, except for "Biomedical and Dental Materials" (D25) and "Pharmaceutical Preparations" (D26), which were excluded due to their lack of direct relevance to cancer research. This approach ensured the dataset focused on chemicals and diseases specifically related to cancer.

Data dictionaries were created for chemicals, diseases, and genes, with no filtering applied to the genes—all gene-related entries were included. A consistent naming convention was used for the subdivision of files, such as 'type_annotations_1_1.csv,' where 'type' represents either gene, disease, or chemical, the first number corresponds to the file, and the second number indicates the segment. Examples of the data dictionaries are provided in Table 6, Table 7, and Table 8, corresponding to chemicals, diseases, and genes, respectively.

*Table 6: Chemical_annotations_1_1.csv*

| Sl. no | annotation_id | annotation_text |
|---|---|---|
| 1 | D008727 | methotrexate |
| 2 | D002955 | leucovorin |
| 3 | D008727 | methotrexate |
| 4 | D008748 | methylcholanthrene |
| 5 | D016604 | aflatoxin B1 |

*Table 7:Disease_annotations_1_1.csv*

| Sl. no | annotation_text | annotation_id |
|---|---|---|
| 1 | Atrial Sarcoma | D012509 |
| 2 | Neck sarcomas | D012509 |
| 3 | Myxoid Sarcoma | D012509 |
| 4 | soft tissue sarcomas | D012509 |
| 5 | Osteo sarcoma | D012509 |

*Table 8: Gene_annotations_1_1.csv*

| Sl no. | annotation_id | annotation_text |
|---|---|---|
| 1 | 367 | androgen receptor |
| 2 | 2908 | glucocorticoid receptor |
| 3 | 11657 | angiotensin II |
| 4 | 5617 | prolactin |

Table 9 illustrates the exact number of concepts at each stage:

*Table 9:Number of concepts at each stage*

| Concept | Original Count | Filtered Count | Notes |
|---|---|---|---|
| Diseases | 31,475,268 | 714,347 | Filtered by "Neoplasms by Site" MeSH |
| Chemicals | 15,113,262 | 6,646,367 | excluding categories such as 'Biomedical and Dental Materials' and 'Pharmaceutical Preparations'. |
| Genes | 17,497,302 | 17,497,302 | No filtering applied |
| Species | 15,321,941 | Not considered | Excluded entirely |
| Cell Lines | 2,539,225 | Not considered | Excluded entirely |

The data filtering process significantly reduced the dataset's size. For instance, disease entries were reduced from 31 million to 714,347, and chemical entries from 15 million to 6.6 million. Gene entries remained unchanged, while species and cell lines were excluded. This targeted filtering made the dataset more manageable and relevant for cancer research.

In our filtering strategy, we treated all terms, including synonyms, as distinct entities. For example, "bladder cancer" and "bladder neoplasms" were treated as different terms, even though they are synonyms. This is a limitation and is mentioned in section 5.2.

## 3.3 Step-by-step approach for Generating Word Embeddings

With the cancer-relevant dataset prepared, the next step involves the process of generating word embeddings. This step-by-step approach ensures the accurate capture and representation of various biomedical entities such as diseases, chemicals, and genes, utilizing their unique identifiers and contextual information from the literature. Figure 4 illustrates the detailed methodology used to systematically generate these embeddings, which forms the core of our analytical framework.

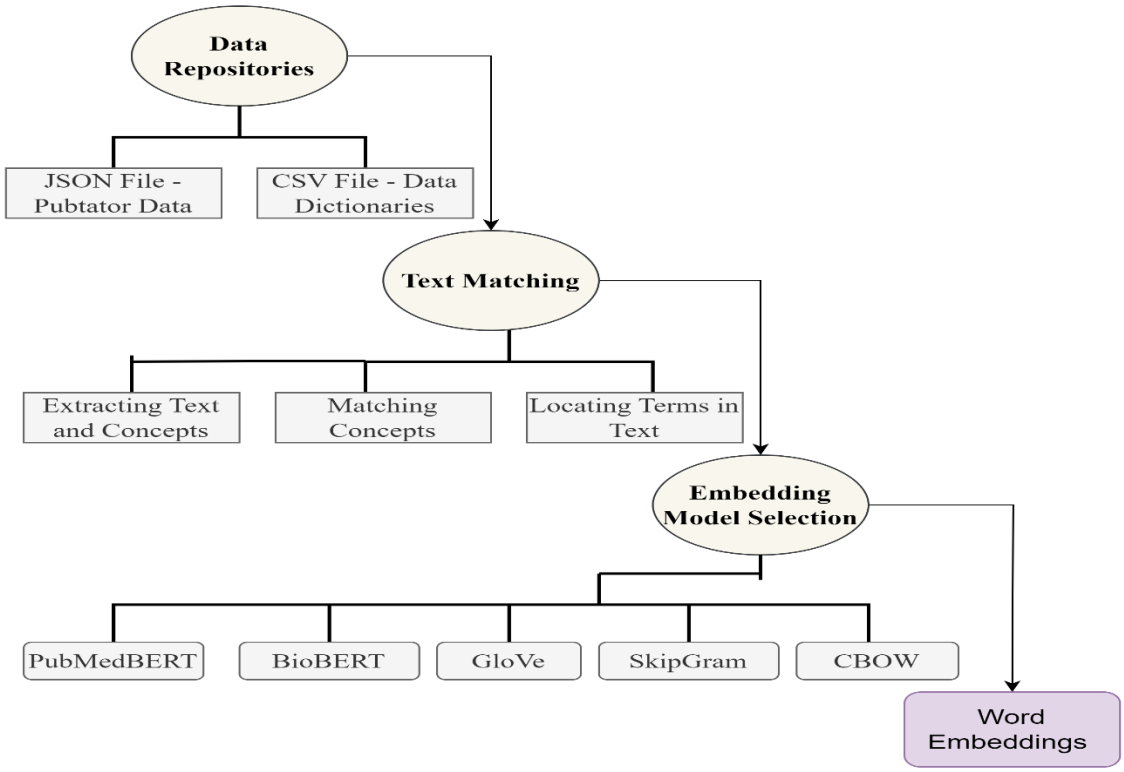


*Figure 4:Flowchart for Generating Word Embeddings from Biomedical Text Data*

**Step 1: Data Repositories**

The first step in generating word embeddings involves identifying and utilizing the key data sources (described in Section 3.2). These include:

- **JSON File:** This file contains biomedical articles with metadata such as PubMed IDs (PMIDs) and corresponding text segments. Each article is annotated with various biomedical concepts, including diseases, species, chemicals, and genes.
- **Data Dictionaries:** These data dictionaries list annotations for various entity types, such as diseases and chemicals, which are matched with MeSH identifiers, and genes, which are associated with gene symbols.

**Step 2: Text Matching**

In this step, the focus is on extracting the text and matching the annotated biomedical concepts to ensure accurate and consistent representation.

- **Extracting Text and Concepts**: Begin by retrieving the ‘ARTICLE’ text and ‘CONCEPTS’ from the JSON file, focusing on entries labeled as 'disease,' 'chemical,' or 'gene' under the 'concept_type' field. This is essential because the JSON data includes various other concept types, such as species and cellLines, which are not relevant for this analysis. By filtering the content based on 'concept_type,' we ensure that only the relevant text related to genes, diseases, and chemicals is extracted.
- **Matching Concepts**:
    - **For Diseases and Chemicals**: In extracted ‘CONCEPTS’ match the ‘IDENTIFIER’ (MeSH ID) from the JSON file with the corresponding entries in the data dictionaries. This ensures that annotations for diseases and chemicals are accurately linked to their respective standardized entries

[72]. This step is necessary because the JSON files contain annotations for a wide range of diseases and chemicals, not just those related to cancer. However, our data dictionaries are filtered to include only cancer-related annotations. To ensure that word embeddings are generated specifically for cancer-related terms, this step is crucial.

- **Locating Terms in Text**: Once the annotations are matched, search within the 'ARTICLE TEXT' for the specific terms from the 'TERM' field of each matched concept. This is crucial as the contextual placement of these terms in the text significantly influences their semantic interpretation. After locating the terms within the text, word embeddings are generated based on the surrounding context to capture their full semantic significance.

**Step 3: Embedding Model Selection**

Word embeddings were created using various models like CBOW, SkipGram, GloVe, BioBERT and PubMedBERT.

**Embedding Generation:**

Once the models are selected, the final step involves generating the word embeddings:

- **Processing the Text**: Tokenize the matched term and context around it using the appropriate tokenizer for each model. Each tokenizer is tailored to the specific model ensuring accurate input preparation.
- **Generating Embeddings**: Feed the tokenized text into each respective model to generate word embeddings. These embeddings capture the relationships and contextual meaning of terms based on the model's unique approach.
- **Maintaining Separate Embedding Datasets**: Store the embeddings generated by each model separately for subsequent analysis.

To handle compound terms, such as "bladder cancer" for diseases, "peroxyl radicals" for chemicals, and "CSF-1 receptor" for genes, we compute embeddings for each individual word within the term. The mean of these individual embeddings is then calculated to produce a unified vector that accurately represents the entire compound term [1][2].

A sample of how the word embeddings appear is shown below. This example illustrates word embeddings for Genes:

Phrase: cln3, Embedding: [0.3112412, -0.0429908148, -0.46422204, -0.31998255, .....]

Phrase: nme1, Embedding: [0.2305971, 0.0265105, -0.422717456, -0.25808057188, .....]

Phrase: cdc42, Embedding: [0.24038206, 0.32040014, -0.49338492, 0.142192199, .....]

Phrase: gas2, Embedding: [-0.05255251, -0.28753986, -0.726874, -0.29208680567, .....]

**Organizational Details of the Word Embeddings**

The generated embedding models include the word embeddings for the entire dataset. The word vectors from these models were stored in files named for the embedding model type, concept type (disease, chemical, gene), and the origin JSON file and segment. For example, filenames like "Glove_disease_1_1," "Glove_chemical_1_1," or "Glove_gene_1_1" are used, where "_1_1" indicates the data segment. This consistent naming convention ensures that each file is easily identifiable and correctly associated with its source data and the embedding method applied.

All sub-files generated during the embedding process were then consolidated into a single file for each category and embedding type. For example, all segments of GloVe disease embeddings were merged into one file named "GloVe_Disease_combined," with a similar approach for chemical and gene embeddings.

## 3.4 Various Libraries Used for Models for Model Implementation

In this study, we utilized various libraries to streamline data processing and generate word embeddings. These libraries played a crucial role in handling large datasets, performing text pre-processing, and creating high-quality word embeddings to capture meaningful relationships between genes, diseases, and chemicals. Below is an overview of the key libraries employed for these tasks.

- **JSON:** Our dataset from PubTator was in JSON format, so we used Python's built-in json library. This library allows us to convert JSON data into Python dictionaries and lists for easier processing
- **pandas (pd):** Employed for data manipulation, pandas was crucial for reading and processing disease annotations stored in CSV files. It provided efficient tools for organizing and analyzing tabular data.
- **Natural Language Toolkit (nltk):** Utilized for text tokenization, nltk's word_tokenize function played a key role in splitting text into individual words. It enabled the conversion of raw text data into tokens, which are essential for subsequent processing steps.
- **os:** This library facilitated interaction with the file system, enabling the management of file paths and directories.

Additionally, several other packages were employed based on the specific word embedding model being implemented, such as Word2Vec, GloVe, and BERT. Word2Vec (SkipGram and CBOW).

We employed specific libraries and tools for generating word embeddings using the Word2Vec model (both SkipGram and Continuous Bag of Words (CBOW)). These included

- **gensim.models.Word2Vec:** We leveraged the Word2Vec implementation from the Gensim library to train the Skip-gram and CBOW models and generate word embeddings.

**GloVe**

The GloVe model implementation primarily relied on the numpy (np) library for numerical operations and array manipulations, which were essential for processing the GloVe embeddings.

**PubMedBERT**

We utilized specific python libraries and AutoModel class for generating word embeddings using the PubMedBERT model. These included

- **Transformers:** This library, developed by Hugging Face, provides pre-trained models and tokenizers for a variety of natural language processing (NLP) tasks. It supports state-of-the-art models like BERT, GPT, and others. In this case, we used it to initialize and apply the PubMedBERT tokenizer and model. The Transformers library simplifies access to these models by offering a unified API for loading pre-trained models, fine-tuning them, and generating embeddings from biomedical text.
- **AutoTokenizer:** Responsible for initializing a tokenizer suitable for the specified pre-trained model, it allowed us to prepare the text data for input to the model. The tokenizer was initialized using the 'microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext' identifier, corresponding to the PubMedBERT model trained by Microsoft.
- **AutoModel:** Used to load the pre-trained model specified by its identifier, the AutoModel class loaded the PubMedBERT model for processing tokenized input and generating embeddings. Once initialized, the model could efficiently generate embeddings for the given biomedical text data, facilitating analysis of disease-related terms extracted from PubMed abstracts.

**BioBERT**

Specific Libraries and BertModel class used to create BioBERT word embeddings are as follows:

- **Transformers:** This library, developed by Hugging Face, provides pre-trained models and tokenizers for a variety of natural language processing (NLP) tasks. It supports state-of-the-art models like BERT, GPT, and others. In this case, we used it to initialize and apply the BioBERT tokenizer and model. The Transformers library simplifies access to these models by offering a unified API for loading pre-trained models, fine-tuning them, and generating embeddings from biomedical text.
- **BertTokenizer**: Responsible for initializing a tokenizer suitable for the specified pre-trained model, it allowed for the preparation of text data for input to the model. The tokenizer was initialized using the 'dmis-lab/biobert-v1.1' identifier, corresponding to the BioBERT model trained by DMIS Lab.
- **BertModel:** Used to load the pre-trained model specified by its identifier, the BertModel class loaded the BioBERT model for processing tokenized input and generating embeddings. Once initialized, the model could efficiently generate embeddings for the given biomedical text data, facilitating the analysis of disease-related terms extracted from PubMed abstracts.

## 3.5 Functional Relatedness for new pairs in CTD 2024

In 2024, the CTD introduced a new dataset. We utilized our word embedding models to examine these new data points for functional relationships. Our goal was to determine if our models, which were trained on abstracts until 2022, could identify the new functional relationships from the 2024 CTD version (i.e., those in CTD 2024 but not in previous versions). The steps followed to calculate the cosine similarity score are outlined below:

- **Loading Files:** Loading both versions of CTD files referred to as old_ctd and new_ctd and ensures both files have the same structure by verifying the column names of each DataFrame.

- **Identifying New Entries**: Finding entries present in the new file but not in the old file. Rows appearing only in the new file were identified as new entries. These new pairs, specifically for gene-chemical, chemical-disease, and disease-gene relationships, were saved in a separate CSV file for further analysis.
- **Loading Embeddings**: Word embeddings for genes, chemicals, and diseases were loaded individually based on the identified gene-chemical, chemical-disease, and disease-gene pairs. The phrase and its corresponding embedding vectors were extracted for further analysis.
- **Cosine Similarity Calculation**: For each corresponding pair, the cosine similarity between their embeddings was calculated. The result, along with the index, corresponding names, and cosine similarity score, was stored in a results list for analysis.

This process was used to identify potential new functional relationships in the 2024 dataset.

### 3.6 Functional Relatedness for curated CTD associations

Having pre-processed the data and applied various word embedding models, including SkipGram, CBOW, GloVe, PubMedBERT and BioBERT, along with their variations, the study proceeds to evaluate how effectively these models capture the functional relatedness between genes, diseases, and chemicals.

This evaluation is conducted by examining high-confidence associations that are based on cosine similarity, using precision and recall as metrics. To that end, our analysis focused on calculating true positives, false negatives and false positives. By balancing precision with recall, the study aimed to optimize the detection of valid biomedical relationships [95].

Figure 5 presents a comprehensive workflow for calculating functional relatedness in gene-chemical-disease associations. Each phase of this process is explained in detail below:

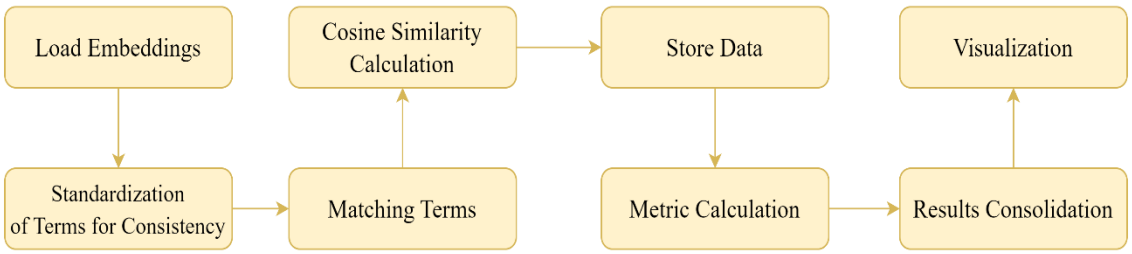


*Figure 5:Flowchart of computing functional relatedness for gene-chemical-disease associations*

- **Load Embeddings**: Word embeddings for genes, chemicals, and diseases are independently loaded for each model. The CTD, containing these pairs, is utilized to align with the embeddings. Each embedding is stored in a structured format, where "Phrase" denotes the specific gene, disease, or chemical term, and "Embedding" represents its corresponding vector, e.g., [vector values].
- **Standardization of Terms for Consistency**: To ensure data consistency we converted all relevant fields such as DiseaseName, ChemicalName, GeneNames in the CTD dataset to lowercase. This step ensures accurate matching of terms between the CTD dataset and the word embedding files.

**Matching Terms:** In this phase, terms from the embedding files are cross-referenced with those in the CTD dataset. Gene, disease, and chemical terms are selected from their respective embedding files according to their pair type disease-gene, gene-chemical, and disease-chemical and are then matched with corresponding fields in the CTD dataset. The matching process is based on an exact comparison of the terms as they appear in both the word embedding files and the CTD dataset. For instance, the term "breast neoplasms" is matched only with "breast neoplasms" and not with "breast cancer," even though both terms convey the same meaning. Similarly, "breast cancer" is matched only with "breast cancer" and not with variations like "breast neoplasms." This is a limitation and is mentioned in section 5.2.

- **Cosine Similarity Calculation**:
    - For each matched pair gene-disease, disease-chemical, and chemical-gene in CTD we calculate the cosine similarity score between the embedding

vectors corresponding to each entity in the pair. This measure quantifies the functional relatedness between the connected terms.

- Additionally, cosine similarity calculations are extended to all unique instances within the three concept types—genes, diseases, and chemicals. Since there is no predefined threshold for cosine similarity, only values above the thresholds of 0.6, 0.7, and 0.8 are considered. Higher cosine similarity scores indicate stronger relationships between the terms, which helps in identifying functional connections in the data more effectively.
    - For each type of concept (e.g., gene), retrieve all of its instances from the CTD pairs (e.g., geneX, geneY, geneZ).
    - For each concept instance, all associated CTD pairs are identified (e.g., geneX-diseaseA, geneX-diseaseB). These represent the correct functional relations.
    - For each word vector representing a concept instance (e.g., geneX), the nearby vectors, referred to as instance vector pairs, with a cosine similarity greater than the set threshold are retrieved. These represent the identified functional relationships.

- **Store Data:** The cosine similarity scores derived from the calculations are stored in two separate CSV files for each word embedding model. One file contains the scores for curated CTD pairs, while the other stores the cosine similarity scores above the set threshold for the instance vector pairs identified in the previous phase.
- **Metric Calculation** For each file, the performance of the model is assessed by calculating the true positives, false negatives, false positives, precision, and recall [95]. The criteria are defined as follows:
    - **True Positives**: Instances where the cosine similarity score exceeds the threshold for pairs that are both in the curated CTD data and identified through instance vector pairs, confirming correct identification.

- **False Negatives**: Instances where pairs are recognized in the curated CTD data but either the cosine similarity score does not exceed the threshold or they are not found in instance vector pairs, indicating missed connections.
- **False Positives**: Instances where the cosine similarity score exceeds the threshold for pairs that are not in the curated CTD data suggesting incorrect identifications.

Based on the values obtained recall and precision are calculated. The mathematical formula to compute precision is as follows:

$$Recall = \frac{(true\ positive)}{(true\ positive + false\ negative)}$$

$$Precision = \frac{(true\ positive)}{(true\ positive + false\ positive)}$$

- **Results Consolidation**: After calculating metrics for all files, the results were merged into a single DataFrame for comparative analysis. This consolidated DataFrame contains precision, recall and other relevant metrics across all models and thresholds, facilitating further analysis and visualization.
- **Visualization**: Heatmaps were generated to visually represent precision and recall values of each model. The model with the highest precision and high recall in the heat map clearly outperformed others in identifying meaningful relationships.

In this chapter, we have outlined a methodology for generating and analyzing word embeddings to explore the functional relatedness between genes, chemicals, and diseases within cancer-related biomedical literature. The process began with data preparation, where we carefully filtered, segmented, and structured the data from PubMed abstracts using NLP tools and MeSH identifiers to focus on cancer-related entities. This provided a solid foundation for generating high-quality word embeddings.

We implemented multiple word embedding models, including SkipGram, CBOW**,** GloVe**,** PubMedBERT**,** and BioBERT**,** each capturing different aspects of semantic relationships. These models were tailored to represent biomedical entities such as diseases, chemicals, and genes with high accuracy.

In the final phase of the study, we assessed the precision and recall of the models for predicting gene-disease, gene-chemical, and disease-chemical associations. This was done by setting thresholds for cosine similarity scores between curated CTD pairs and derived instance vector pairs.

This comprehensive evaluation is crucial as it highlights the models that are most effective in capturing functional relatedness within the biomedical field, thereby guiding future research in identifying and utilizing the most reliable models for detailed biomedical analysis.

# 4. Results

In this results chapter, we evaluate how well word embedding models capture functional relationships between gene-chemical, gene-disease, and disease-chemical pairs by calculating cosine similarity. Different threshold values are applied to calculate key metrics such as true positives, false negatives, false positives, precision, and recall. The performance of these models is visualized using heatmaps to enhance clarity.

We also assess the impact of hyperparameters, including window size and vector size, for CBOW and SkipGram models. Additionally, we analyze embeddings from different layers of PubMedBERT and BioBERT, as well as GloVe variations based on the number of tokens used during pre-training. For PubMedBERT and BioBERT, we calculate the average similarity for chemical, disease, and gene pairs across different layers to identify which layers are best suited for evaluating functional relatedness by comparing their performance.

We also tested whether word embeddings trained on abstracts up to 2022 could capture the functional relatedness of new gene-disease, disease-chemical, and chemical-gene pairs present in the 2024 CTD release.

## 4.1 Average Similarity of PubMedBERT and BioBERT across different layers.

While generating word embeddings using PubMedBERT and BioBERT, we extracted and saved the embeddings from various hidden layers, specifically the last four individual hidden layers and the summation of values from those layers. Here:

- **-1**: Refers to the very last hidden layer.
- **-2**: Refers to the second-to-last hidden layer.
- **-3**: Refers to the third-to-last hidden layer.
- **-4**: Refers to the fourth-to-last hidden layer.
- **Sum**: Refers to the summation of the last four hidden layers' embeddings.

These embeddings were then compared to evaluate the performance of each hidden layer in capturing functional relationships. Each variant of the model was systematically compared by calculating the average cosine similarity between phrases common to the embedding files. Cosine similarity was used as a metric to quantify the alignment of embeddings across different layers. This process was applied to Chemical, Disease, and Gene files. After calculating the average similarities, the results were visualized using bar graphs to facilitate comparison between the different layers.

### 4.1.1 PubMedBERT

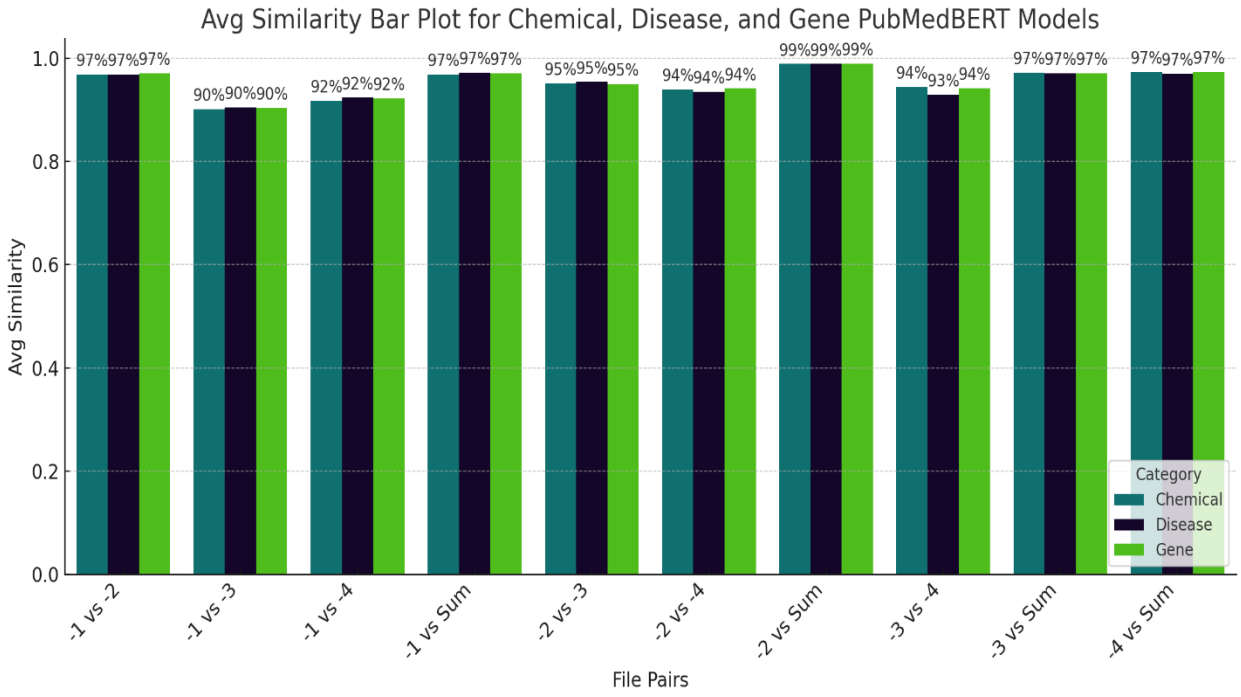


*Figure 6:Average Similarity in Chemical, Disease, and Gene word embeddings for PubMedBERT models*

Figure 6, provides a comprehensive comparison of word embedding values extracted from different layers of the PubMedBERT model. The key observations include:

- **Consistent High Similarity Across Categories**: All categories (Chemical, Disease, and Gene) generally exhibit high average similarity scores, predominantly

above 90%. This reflects a strong consistency in embeddings across different files within each category.

- **Uniformity Across Pair Comparisons**: The pair comparisons (-1 vs -2, -1 vs -3, etc.) show relatively uniform similarity values across all categories for each file pair, suggesting that the embeddings maintain a consistent representation of biomedical concepts across different files.
- **Exceptional Performance with Summed Embeddings**: Particularly high similarity scores are observed for some file pairs, such as "-2 vs Sum", "-3 vs Sum", and "-4 vs Sum", where they approach or reach 99%. This demonstrates that the summation of embeddings from the last four layers captures a comprehensive and consistent representation that aligns well with individual layer embeddings.

### 4.1.2 BioBERT

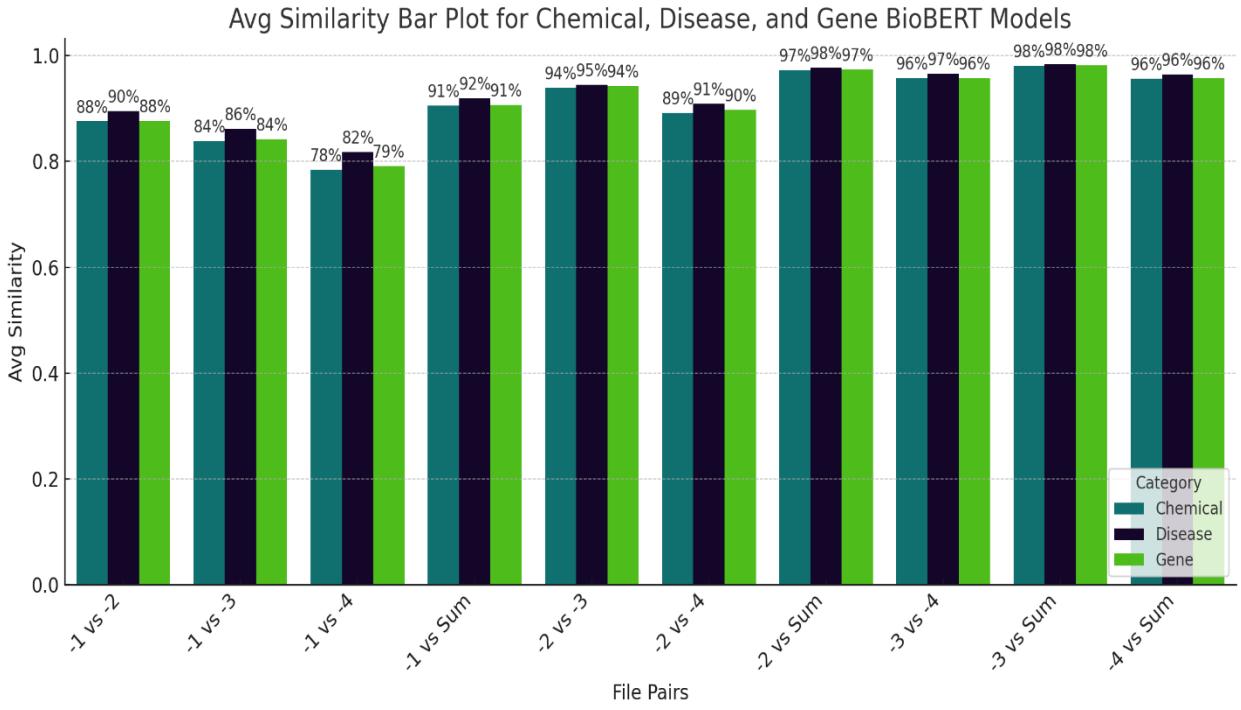


*Figure 7:Average Similarity in Chemical, Disease, and Gene word embeddings for BioBERT models*

Here are some detailed observations from Figure 7, depicting average similarities between different file pairs for Chemical, Disease, and Gene categories in BioBERT models:

- **General High Similarity**: Average similarity scores range from 78% to 98% across all categories, indicating consistent representation of biomedical concepts by the BioBERT model's layers and their summations.
- **Variability in Early Comparisons**: Scores show more variability in early layer comparisons, particularly "-1 vs -3" and "-1 vs -4", dipping to as low as 78%. This suggests different encoding capabilities of specific layers.
- **Enhanced Performance with Summed Embeddings**: Higher similarity scores up to 98% are observed in "-3 vs Sum" and "-4 vs Sum" comparisons. This highlights the effectiveness of combining outputs from the last four layers, which appears to capture comprehensive and representative features of the data.
- **Insight into Layer Functionality**: The variability in similarity among early layers may indicate nuanced differences in how these layers process biomedical entities across different contexts.

The analysis of PubMedBERT and BioBERT models, as depicted in Figures 1 and 2, reveals that both models consistently produce high similarity scores across Chemical, Disease, and Gene categories, highlighting their robustness in embedding biomedical texts. While PubMedBERT displays very high uniformity and effectiveness, especially when combining outputs from the last four layers, BioBERT shows a broader variability in early layer comparisons. Despite these differences, both models excel in synthesized layer performances. This underscores their potential utility in diverse biomedical text analysis applications.

## 4.2 Functional Relatedness

In this study, we evaluated the performance of several word embedding models—CBOW, SkipGram, GloVe, PubMedBERT, and BioBERT—to assess their effectiveness in capturing functional relationships among genes, diseases, and chemicals within cancer-

related data. Cosine similarity scores of instance vector pairs are validated with curated pairs in CTD. Our evaluation centered on precision and recall metrics as calculated using true positives, false positives, and false negatives. Results were visualized using heat maps to illustrate the functional relatedness among gene-disease, disease-chemical, and chemical-gene associations as captured by the various models. Furthermore, we analyzed these models based on the precision and recall values obtained, using the set thresholds. This approach provided a comprehensive understanding of each model's capabilities in handling complex biomedical data relationships.

The naming conventions used in our study were as follows:

- For **BioBERT** and **PubMedBERT**, the notation "BioBERT1/-4" refers to the BioBERT model, with embeddings taken from the fourth-to-last hidden layer. A similar approach was used for PubMedBERT.
- For **GloVe**, we used "42B" and "840B," representing the pre-trained model based on the number of tokens (42 billion and 840 billion tokens, respectively).
- For **CBOW** and **SkipGram**, the notation "SkipGram/W10/300" refers to the SkipGram model with a window size of 10 and a vector size of 300. This pattern was followed for CBOW as well.

### 4.2.1 Disease-Gene Associations

Table 10 presents an example of the PubMedBERT1/-4 model for Disease-Gene pairs, including the Disease Name, Gene Name, and their cosine similarity score.

*Table 10:Cosine Similarity Score for Disease-Gene pairs*

| DiseaseName | DiseaseID | GeneID | GeneName | CosineSimilarity |
|---|---|---|---|---|
| colorectal neoplasms | MESH:D015179 | 201163 | flcn | 0.943988 |
| liver neoplasms | MESH:D008113 | 201163 | flcn | 0.943921 |
| cholangiocarcinoma | MESH:D018281 | 2778 | gnas | 0.943882 |

| | | | | |
|---|---|---|---|---|
| liver neoplasms | MESH:D008113 | 5800 | ptpro | 0.943816 |
| meningioma | MESH:D008579 | 1476 | cstb | 0.943593 |
| breast neoplasms | MESH:D001943 | 6926 | tbx3 | 0.59475 |
| colonic neoplasms | MESH:D003110 | 10765 | kdm5b | 0.593125 |
| glioma | MESH:D005910 | 9636 | isg15 | 0.589403 |
| colonic neoplasms | MESH:D003110 | 3627 | cxcl10 | 0.587069 |

**Cosine Threshold: 0.6**

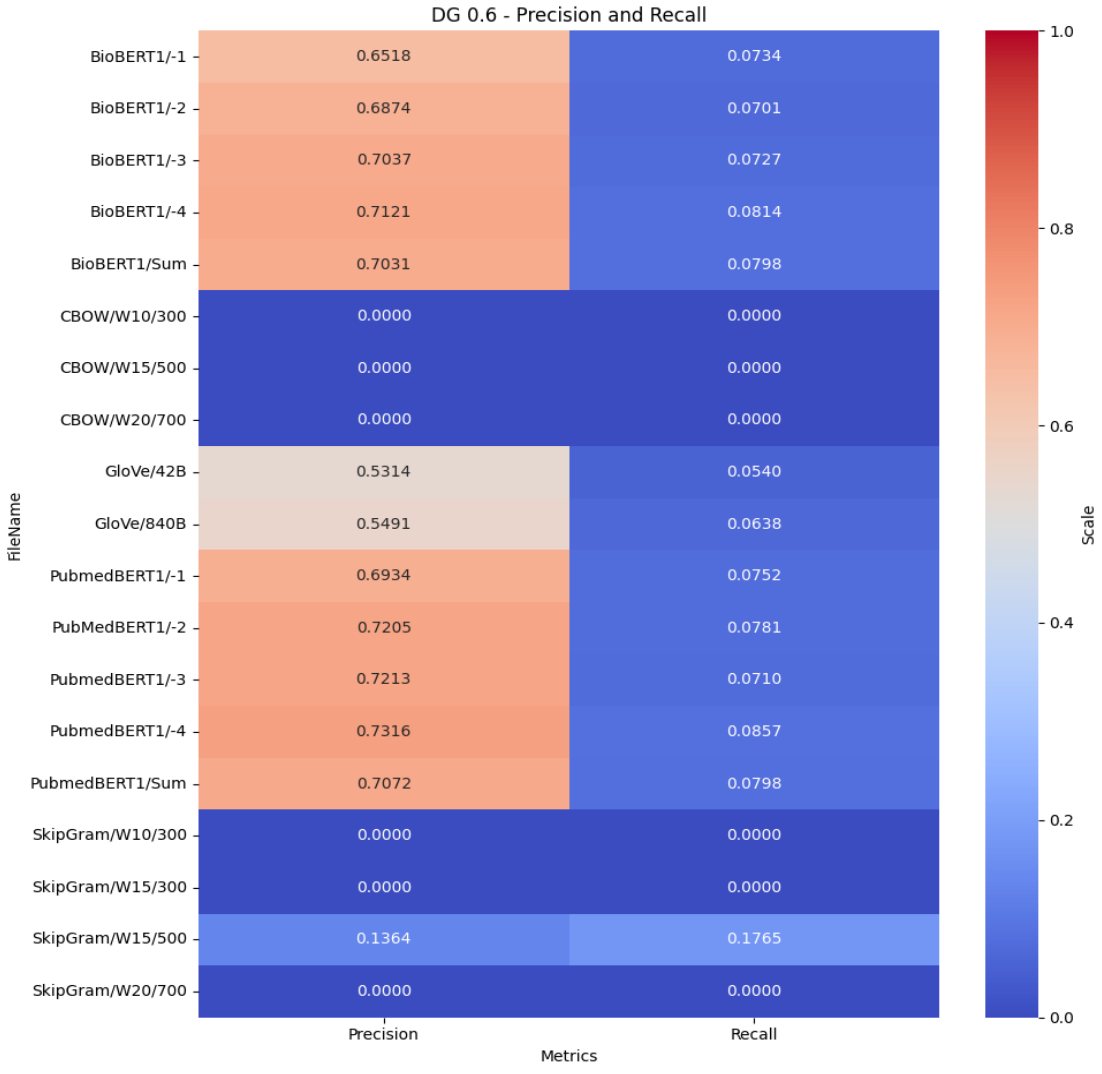


*Figure 8:Precision and Recall values of Disease-Gene pair for cosine threshold 0.6*

Figure 8 provides an analysis of precision and recall values for Disease-Gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.6. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models show relatively strong precision with BioBERT1/-4 leading at 0.7121. The recall rates are low, with BioBERT1/-4 is leading at 0.0814, indicating its effectiveness in capturing relevant cases comprehensively compared to other variants.
- **PubMedBERT Variants:** Similarly, the PubMedBERT models show higher precision, with PubMedBERT1/-4 achieving the highest precision at 0.7316 among all models. It also demonstrates one of the highest recall rates at 0.0857, although recall values remain extremely low across all models. This suggests that PubMedBERT performs better in retrieving comprehensive data compared to other models.
- **GloVe Models:** GloVe variants show lower precision and recall than BERT models, with GloVe/840B performing slightly better than GloVe/42B but still not reaching the effectiveness of the BERT models in biomedical contexts.
- **SkipGram Models:** Most SkipGram configurations show negligible precision and recall. However, SkipGram/W15/500 stands out with modest precision and significantly higher recall at 0.1765, suggesting it might be more capable of retrieving relevant cases broadly but with less accuracy.
- **CBOW Models:** All CBOW variants exhibit zero performance in both precision and recall, confirming their ineffectiveness for detailed biomedical retrieval tasks at this threshold.

**Cosine Threshold 0.7:**

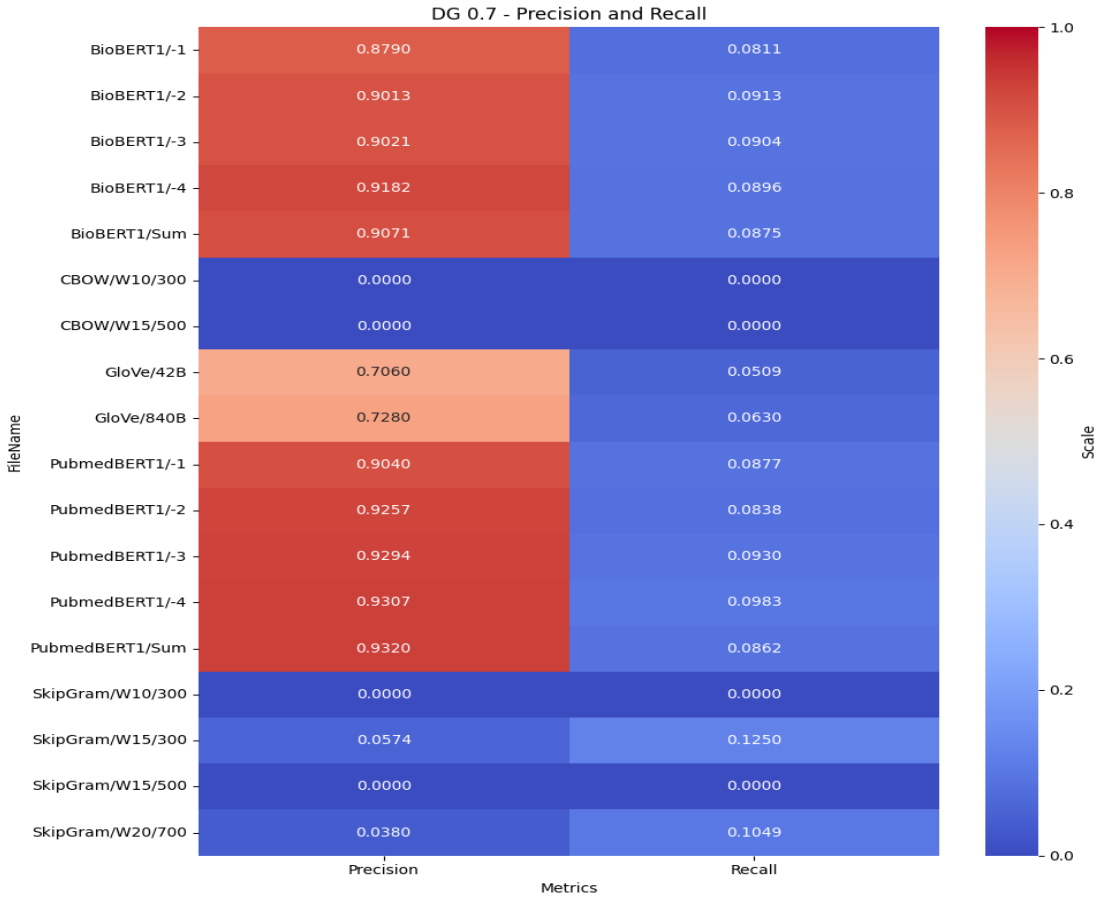


*Figure 9:Precision and Recall values of Disease-Gene pair for cosine threshold 0.7*

Figure 9, provides an analysis of precision and recall values for Disease-Gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.7. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models exhibit exceptionally high precision, with BioBERT1/-4 achieving the highest at 0.9182. The recall figures are low, with BioBERT1/-2 displaying the highest recall, suggesting its enhanced capability to encompass a wider array of relevant cases.

- **PubMedBERT Variants:** PubMedBERT models show excellent precision, with PubMedBERT1/-4 leading at 0.9307 and also achieving the highest recall at 0.0983 among the variants. This performance highlights its capability to accurately identify and comprehensively retrieve relevant biomedical data.
- **GloVe Models:** GloVe models show moderate precision and comparatively lower recall than the BERT variants, with GloVe/840B making slight improvements in both metrics. However, they remain less effective for highly specialized biomedical tasks.
- **CBOW Models:** All CBOW configurations register zero in both precision and recall, reaffirming their unsuitability for intricate biomedical retrieval tasks at this threshold.
- **SkipGram Models:** Most SkipGram configurations exhibit negligible performance. However, SkipGram/W15/300 shows an anomalously high recall, suggesting potential for retrieving a broad array of data but with compromised precision. This model has higher recall because of lower counts of the values.

**Cosine Threshold 0.8:**

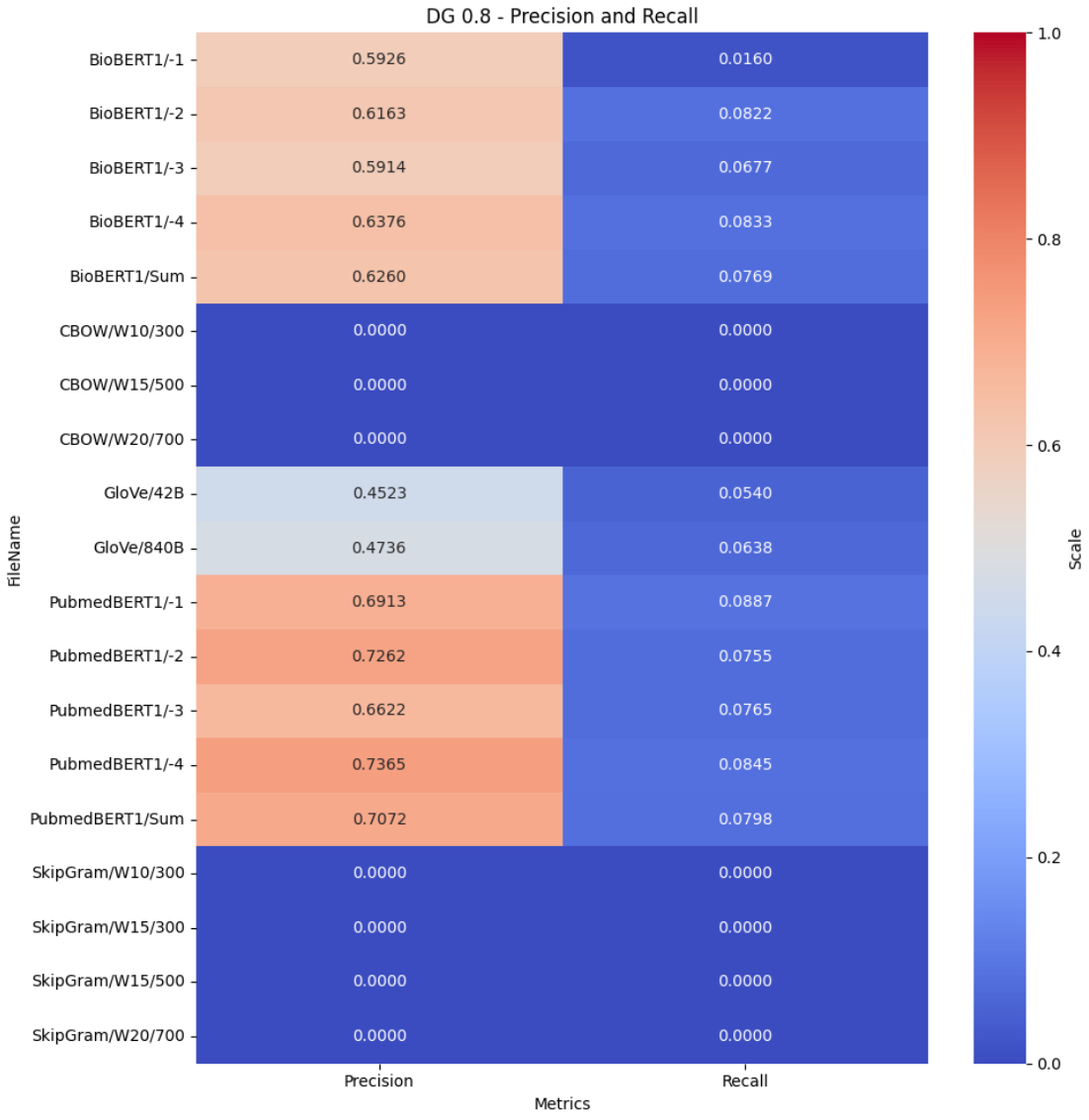


*Figure 10:Precision and Recall values of Disease-Gene pair for cosine threshold 0.8*

Figure 10 provides an analysis of precision and recall values for Disease-Gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.8. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models demonstrate relatively high precision, with BioBERT1/-4 achieving the highest at 0.6376. Recall is generally low, with

BioBERT1/-4 also leading at 0.0833, showing its relative effectiveness in capturing a broader set of relevant cases compared to other variants.

- **PubMedBERT Variants:** PubMedBERT models also exhibit higher precision, particularly PubMedBERT1/-4 at 0.7365, showing it slightly outperforms others in accurately identifying relevant data. The recall is low, with PubMedBERT1/-1 achieving the highest at 0.0887, suggesting effective capabilities in data retrieval.
- **GloVe Models:** GloVe models show lower precision and recall than BERT models, with GloVe/840B slightly outperforming GloVe/42B but still not matching the effectiveness of the BERT variants for specialized tasks.
- **SkipGram Models:** SkipGram/W15/500 registers zero in both precision and recall, underscoring its ineffectiveness for complex biomedical retrieval tasks at this threshold.
- **CBOW Models:** All CBOW configurations show zero performance in both precision and recall, reinforcing their unsuitability for detailed biomedical retrieval tasks.

**Overall Conclusion for Disease-Gene pairs:**

Across the various cosine similarity thresholds (0.6, 0.7, and 0.8), the PubMedBERT and BioBERT models consistently outperform the other embedding models in terms of precision and recall. PubMedBERT1/-4 exhibits the highest performance metrics, showing exceptionally high precision across all thresholds and the highest recall particularly at the 0.7 threshold (0.0983), making it arguably the most robust model for identifying and retrieving relevant biomedical data.

### 4.2.2 Disease-Chemical Associations

Table 11 presents an example of the PubMedBERT1/-4 model for Disease-Chemical pairs, including the Chemical Name, Disease Name, and their cosine similarity score.

*Table 11:Cosine Similarity scores for Disease-Chemical pairs*

| ChemicalName | ChemicalID | DiseaseName | DiseaseID | CosineSimilarity |
|---|---|---|---|---|
| nitrosamines | D009602 | esophageal neoplasms | MESH:D004938 | 0.959488626 |
| isoflavones | D007529 | endometrial neoplasms | MESH:D016889 | 0.959131144 |
| nitrates | D009566 | prostatic neoplasms | MESH:D011471 | 0.958609903 |
| testosterone propionate | D043343 | prostatic neoplasms | MESH:D011471 | 0.958592126 |
| phosphatidylcholines | D010713 | liver neoplasms | MESH:D008113 | 0.957530848 |
| azathioprine | D001379 | endometrial neoplasms | MESH:D016889 | 0.956387177 |
| polyphenols | D059808 | colonic neoplasms | MESH:D003110 | 0.756567442 |
| vincristine | D014750 | prostatic neoplasms | MESH:D011471 | 0.756316716 |
| prednisone | D011241 | liver neoplasms | MESH:D008113 | 0.755612508 |

**Cosine Threshold 0.6:**

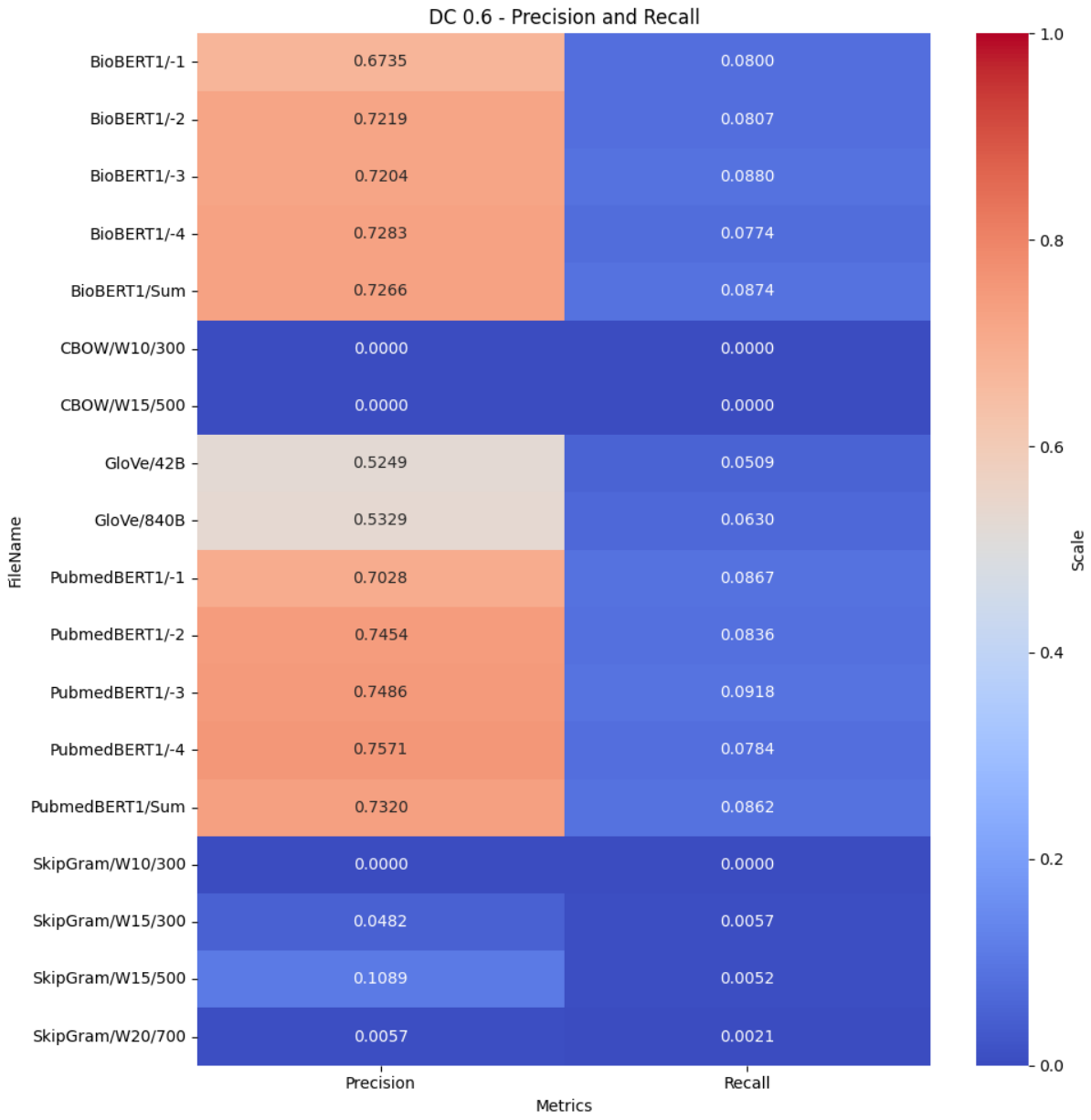


*Figure 11:Precision and Recall values of Disease-Chemical pair for cosine threshold 0.6*

Figure 11 provides an analysis of precision and recall values for Disease-Chemical pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.6. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models exhibit relatively high precision across the board, with BioBERT1/-4 achieving the highest at 0.7283. Recall is low, with BioBERT1/-3 showing the highest recall, indicating its relative effectiveness in capturing a broader set of relevant cases compared to other variants.
- **PubMedBERT Variants:** PubMedBERT models also show relatively high precision, particularly PubMedBERT1/-4 leading at 0.7571, indicating superior accuracy in identifying relevant data. The recall is low, with PubMedBERT1/-3 achieving the highest at 0.0918, suggesting effective capabilities in data retrieval.
- **GloVe Models:** GloVe models demonstrate lower precision and recall than the BERT models, with GloVe/840B performing slightly better in both metrics but still not reaching the effectiveness of the BERT variants for specialized tasks.
- **SkipGram Models:** SkipGram/W15/500 shows very limited performance with low precision and minimal recall, highlighting its limited utility in complex biomedical retrieval tasks at this threshold.
- **CBOW Models:** All CBOW configurations show zero performance in both precision and recall, reinforcing their ineffectiveness for detailed biomedical retrieval tasks.

**Cosine Threshold 0.7:**

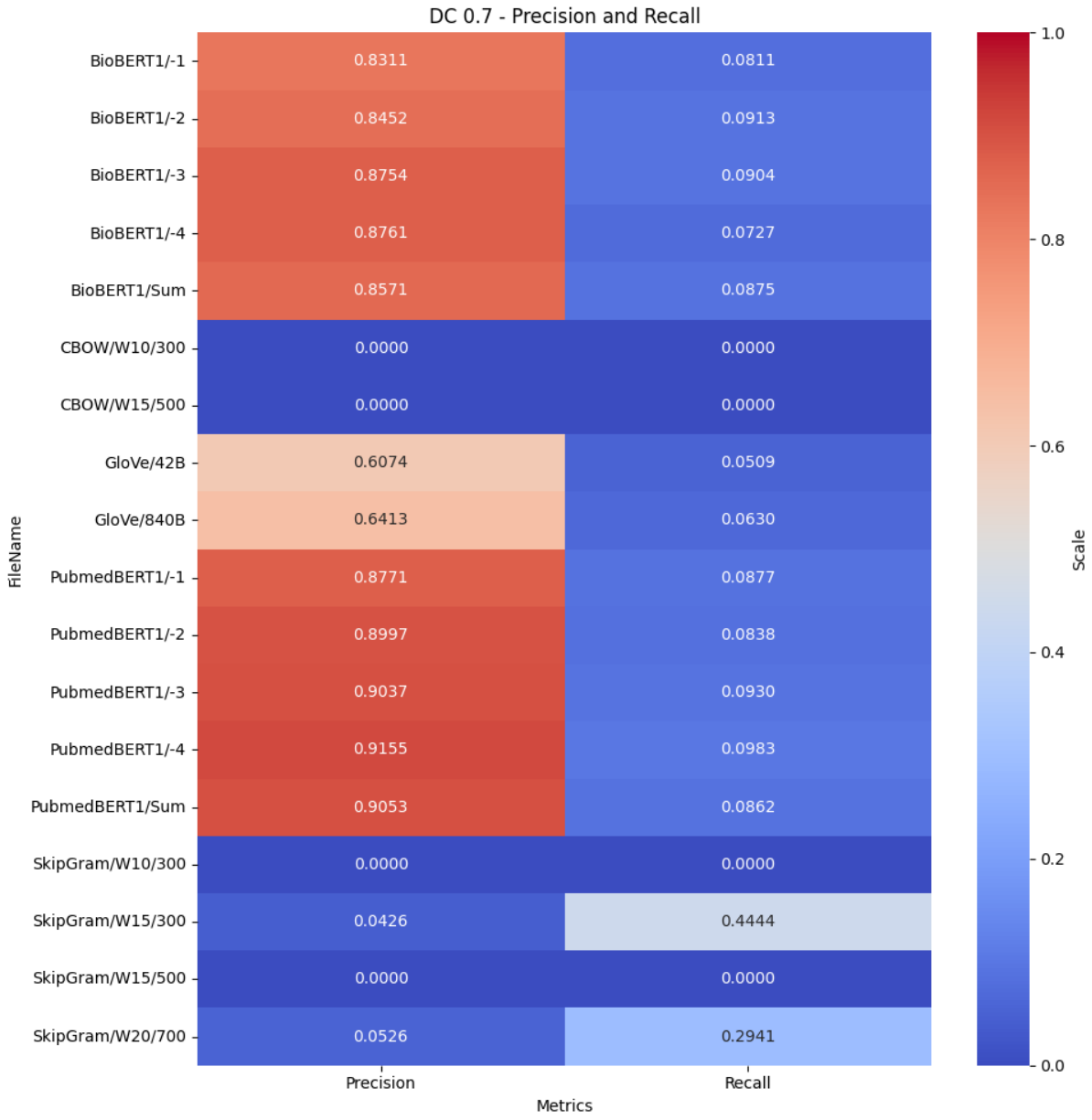


*Figure 12:Precision and Recall values of Disease-Chemical pair for cosine threshold 0.7*

Figure 12 provides an analysis of precision and recall values for disease-chemical pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.7. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models display high precision, with BioBERT1/-4 leading at 0.8761. Recall is low, with BioBERT1/-2 showing the highest recall, suggesting its relative effectiveness in capturing a broader set of relevant cases.
- **PubMedBERT Variants:** PubMedBERT models also exhibit exceptional precision, particularly PubMedBERT1/-4 at 0.9155, which outperforms others in precision and achieves the highest recall at 0.0983 when compared with other models. This underscores its robust capabilities in both accurately identifying and comprehensively retrieving relevant biomedical data.
- **GloVe Models:** GloVe models show lower precision and recall than the BERT models, with GloVe/840B performing slightly better but still lagging behind in effectiveness for specialized biomedical tasks.
- **SkipGram Models:** SkipGram/W15/300 demonstrates very low precision and anomalously high recall, which may need further verification to rule out data anomalies but suggests potential utility in scenarios where broad data capture is more critical than high precision.
- **CBOW Models:** All CBOW configurations show zero performance in both precision and recall, reinforcing their ineffectiveness for detailed biomedical retrieval tasks at this threshold.

**Cosine Threshold 0.8:**

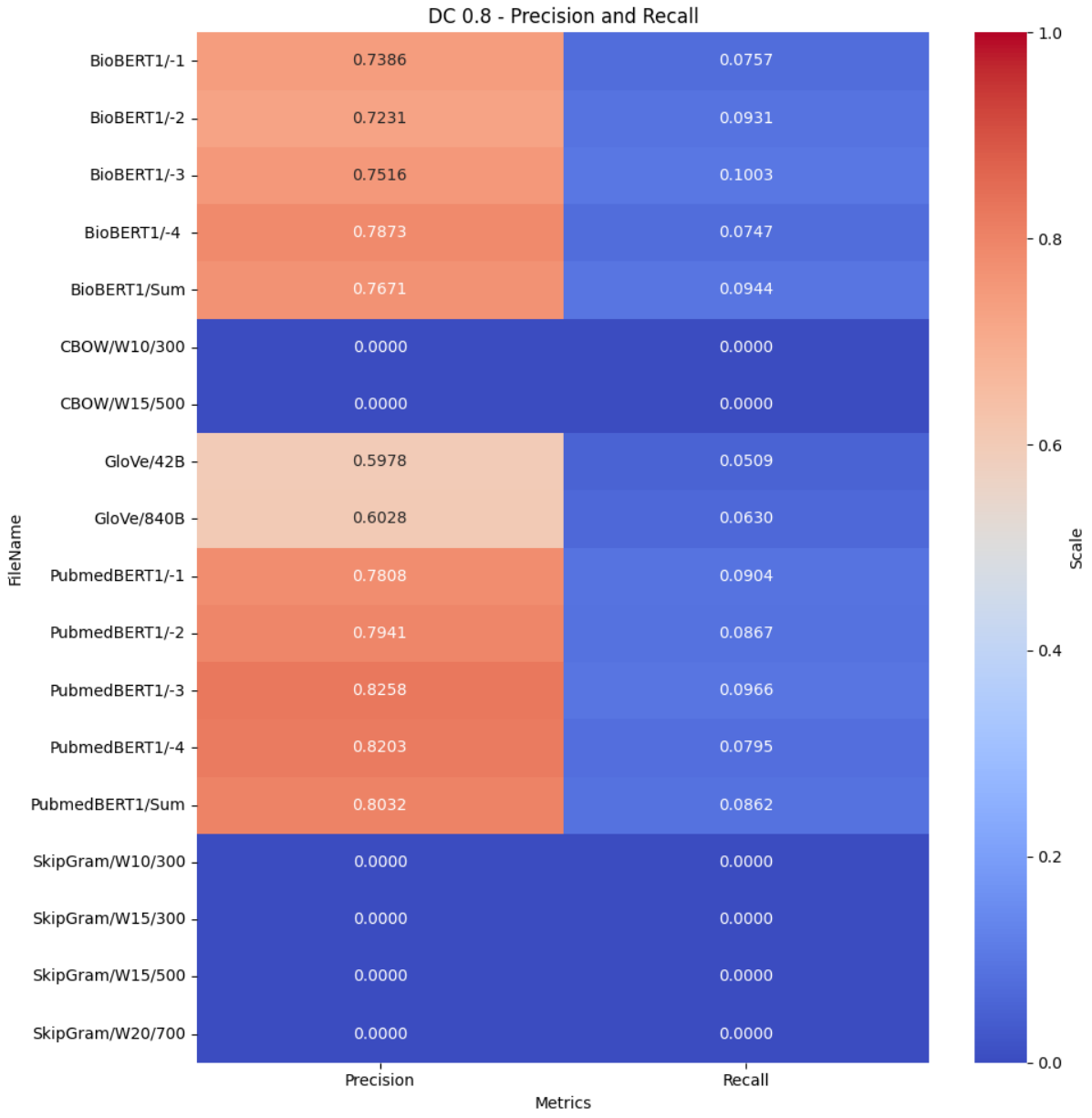


*Figure 13:Precision and Recall values of Disease-Chemical pair for cosine threshold 0.8*

Figure 13 provides an analysis of precision and recall values for disease-chemical pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.7. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models maintain relatively high precision, with BioBERT1/-4 registering the highest at 0.7873. Recall is somewhat varied, with BioBERT1/-3 showing the highest at 0.1003, indicating its effectiveness in capturing a broader range of relevant cases compared to other variants.
- **PubMedBERT Variants:** PubMedBERT models demonstrate high precision, particularly PubMedBERT1/-3 leading at 0.8258, highlighting its strong ability to accurately identify pertinent biomedical data. The recall is low, with PubMedBERT1/-3 again leading at 0.0966, suggesting effective data retrieval capabilities.
- **GloVe Models:** GloVe models present lower precision and recall than the BERT models, with slightly better performance in GloVe/840B. However, these models still do not match the effectiveness of the BERT variants for specialized biomedical tasks.
- **SkipGram Models:** SkipGram/W15/300 shows no performance in either precision or recall, underscoring its limited utility in detailed biomedical retrieval tasks at this higher threshold.
- **CBOW Models:** All CBOW configurations display zero performance in both precision and recall, confirming their unsuitability for intricate biomedical retrieval tasks.

**Overall Conclusion Disease-Chemical pairs:**

The analysis across three cosine similarity thresholds (0.6, 0.7, and 0.8) shows that PubMedBERT and BioBERT models consistently outperform other embedding models in terms of precision and recall, making them the most reliable for biomedical data retrieval tasks. The PubMedBERT1/-4 consistently shows high precision across all thresholds and particularly shines at a threshold of 0.7, where it also achieves the highest recall when compared with other models. This suggests its optimal performance in accurately identifying and comprehensively retrieving relevant biomedical data.

### 4.2.3 Chemical-Gene Associations

Table 12 presents an example of the PubMedBERT1/-4 model for Chemical-Gene pairs, including the Chemical Name, Disease Name, and their cosine similarity score.

*Table 12: Cosine Similarity Score for Chemical-Gene pairs*

| ChemicalName | ChemicalId | GeneId | GeneName | CosineSimilarity |
|---|---|---|---|---|
| perfluorooctanoic acid | C023036 | 55573 | cdv3 | 0.774124406 |
| indomethacin | D007213 | 91584 | plxna4 | 0.774122145 |
| perfluorooctanoic acid | C023036 | 64759 | tns3 | 0.774116386 |
| calcitriol | D002117 | 55840 | eaf2 | 0.774112806 |
| benzo(a)pyrene | D001564 | 2056 | epo | 0.874294305 |
| arsenite | C015001 | 6696 | spp1 | 0.874290557 |
| trichostatin a | C012589 | 5901 | ran | 0.874279046 |
| doxorubicin | D004317 | 6139 | rpl17 | 0.036224524 |
| beta-naphthoflavone | D019324 | 11065 | ube2c | 0.036223694 |
| aflatoxin b1 | D016604 | 340578 | dcaf12l2 | 0.036210815 |

**Cosine Threshold 0.6:**

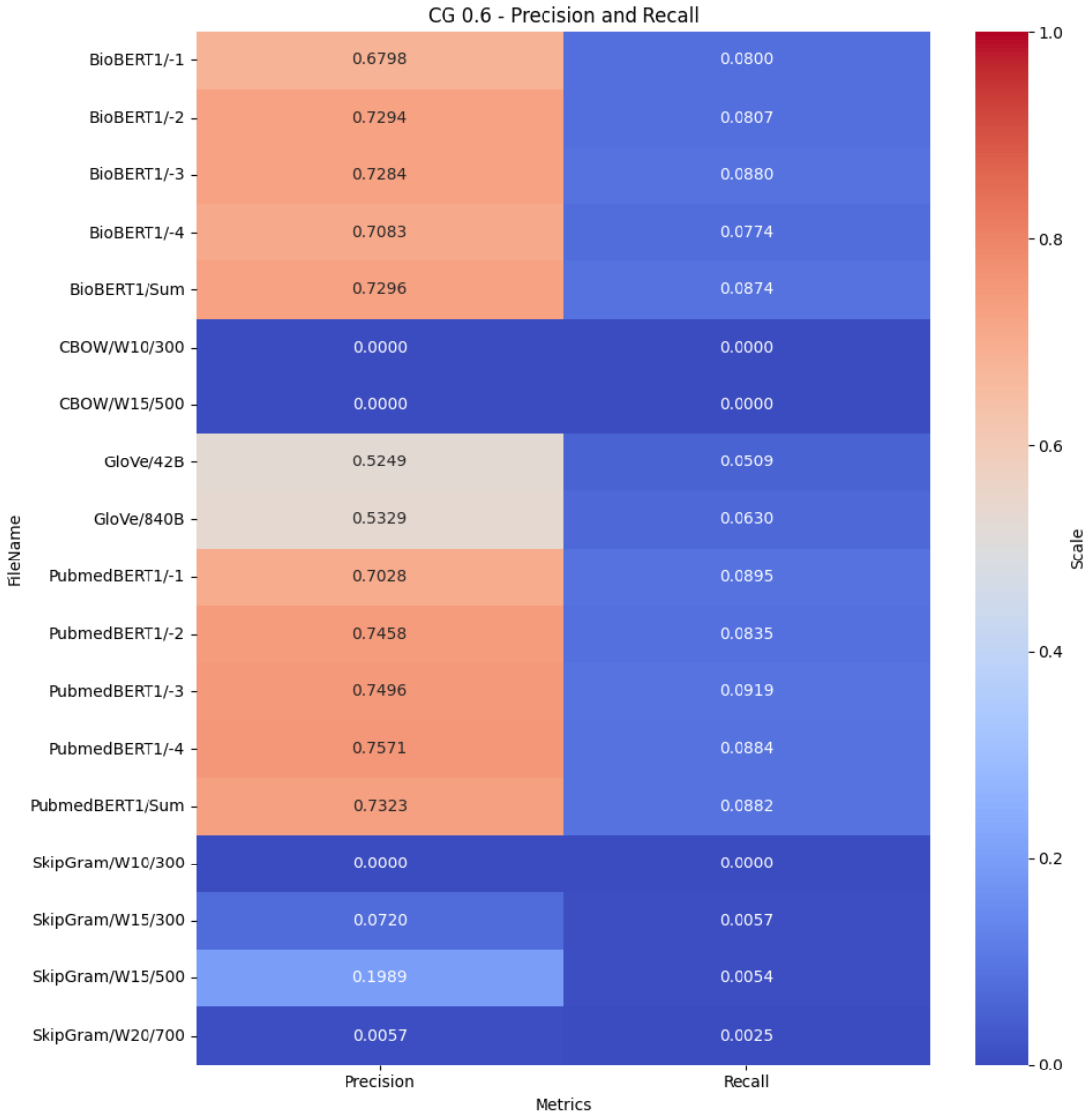


*Figure 14 Precision and Recall values of Gene-Chemical pair for cosine threshold 0.6*

Figure 14 provides an analysis of precision and recall values for chemical-gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.6. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models exhibit relatively good precision across variants, with BioBERT1/-2 and BioBERT1/-3 showing the highest precision. The recall is low, with BioBERT1/-3 leading, indicating its effectiveness in capturing a broader set of relevant cases.
- **PubMedBERT Variants:** The PubMedBERT models exhibit relatively high precision, with PubMedBERT1/-4 achieving the highest precision at 0.7571. Although recall remains low, PubMedBERT1/-3 leads with a recall of 0.0919, indicating effective data retrieval capabilities.
- **GloVe Models:** GloVe models present lower precision and recall than the BERT models. However, GloVe/840B performs slightly better in both metrics but still falls short in effectiveness for complex tasks.
- **SkipGram Models:** SkipGram models generally show minimal performance. Notably, SkipGram/W15/500 has a precision spike at 0.1989 but with very low recall, indicating limited practical utility for detailed retrieval tasks.
- **CBOW Models:** All CBOW configurations display zero performance in both precision and recall, confirming their unsuitability for detailed biomedical retrieval tasks at this threshold.

## Cosine Threshold 0.7

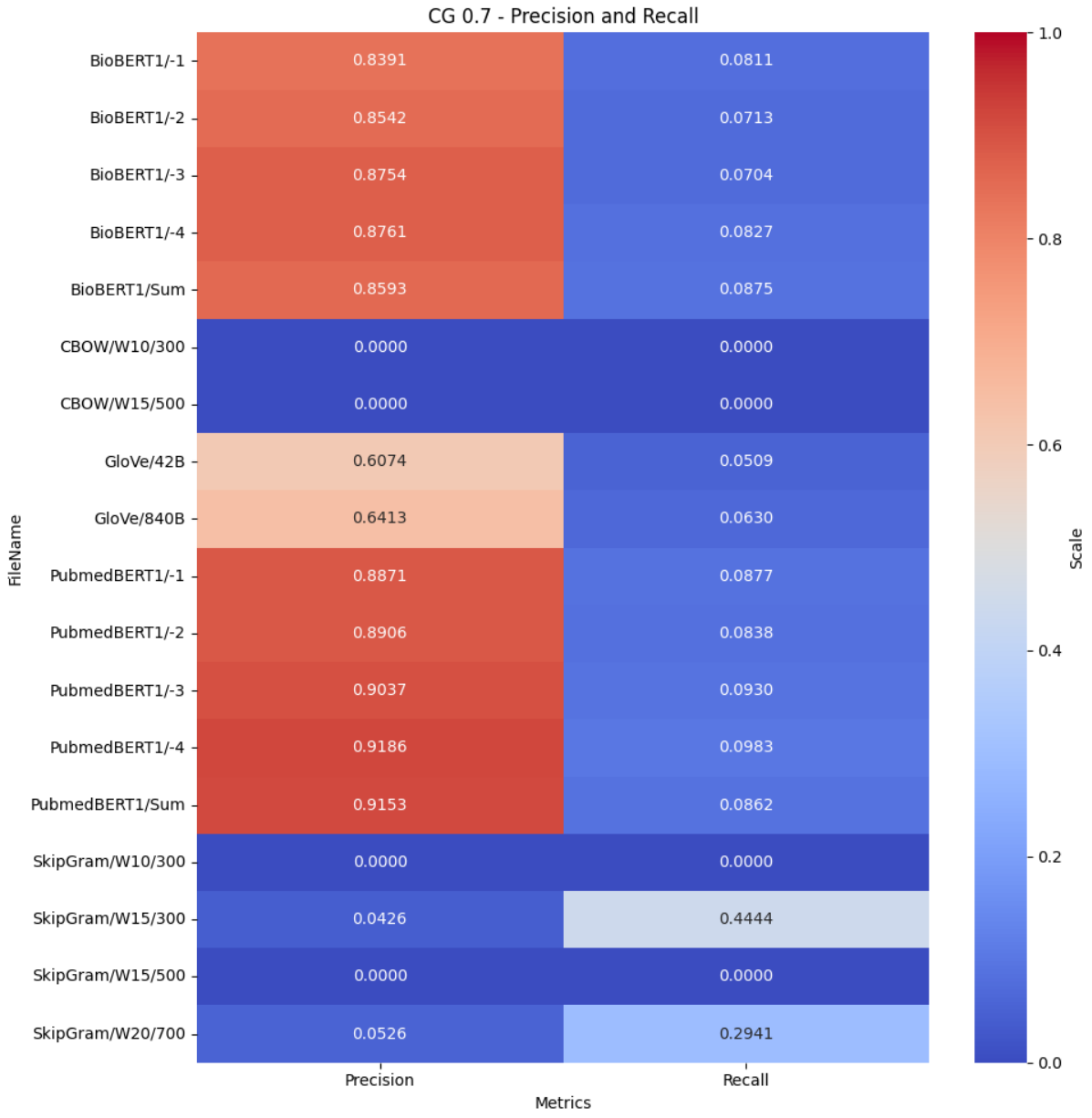


*Figure 15:Precision and Recall values of Gene-Chemical pair for cosine threshold 0.7*

Figure 15 provides an analysis of precision and recall values for chemical-gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.7. The key observations from this analysis are summarized below:

- **BioBERT Variants:** BioBERT models demonstrate high precision across variants, with BioBERT1/-4 showing the highest precision. Recall figures vary, with BioBERT1/Sum showing the highest recall, suggesting it effectively captures a broader set of relevant cases.
- **PubMedBERT Variants:** PubMedBERT models display exceptionally high precision, particularly PubMedBERT1/-4, which shows the highest precision and recall among all models. This indicates its superior capabilities in accurately identifying and comprehensively retrieving relevant biomedical data.
- **GloVe Models:** GloVe models show moderate precision and lower recall than the BERT models, indicating limitations in their effectiveness for specialized biomedical tasks.
- **SkipGram Models:** SkipGram/W15/300 shows a very low precision but anomalously high recall, which might be an outlier or indicate specific conditions under which this model performs uniquely. Further investigation into the conditions causing such a recall spike would be necessary.
- **CBOW Models:** All CBOW configurations show zero performance in both precision and recall, highlighting their ineffectiveness for detailed biomedical retrieval tasks at this threshold.

## Cosine Threshold 0.8

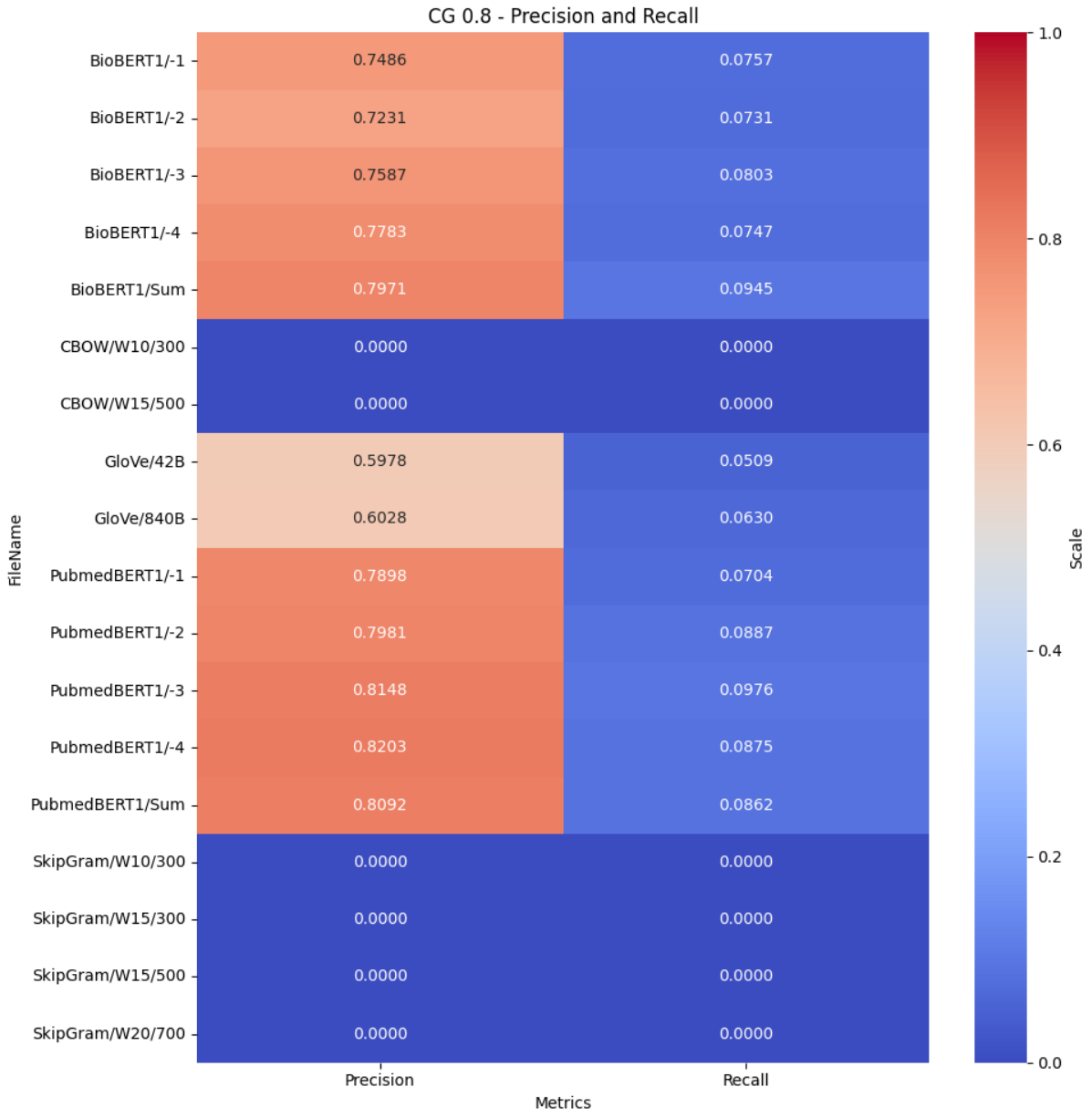


*Figure 16:Precision and Recall values of Gene-Chemical pair for cosine threshold 0.8*

Figure 16 provides an analysis of precision and recall values for chemical-gene pairs across various word embedding models, evaluated at a cosine similarity threshold of 0.8. The key observations from this analysis are summarized below:

- **BioBERT Variants:** The BioBERT models consistently demonstrate high precision across all variants, with BioBERT1/Sum achieving the highest precision. While recall remains low, BioBERT1/Sum still leads, indicating its relative effectiveness in capturing a wider range of relevant cases.
- **PubMedBERT Variants:** PubMedBERT models demonstrate relatively high precision, particularly PubMedBERT1/-4, which shows the highest precision and substantial recall among all models. This indicates its superior capabilities in accurately identifying and comprehensively retrieving relevant biomedical data.
- **GloVe Models:** GloVe models show moderate precision and lower recall than the BERT models, indicating limitations in their effectiveness for specialized biomedical tasks.
- **CBOW Models:** Their inability to register any significant performance highlights the challenges they face in the specialized domain of chemical-gene relationships, particularly at higher thresholds. This trend suggests that CBOW models are not adequately capturing the nuances required for effective data retrieval in this specific context.
- **SkipGram Models:** All SkipGram configurations display zero performance in both precision and recall, highlighting their limitations for detailed biomedical retrieval tasks at this higher threshold.

**Overall Conclusion for Chemical-Gene:**

The examination of various models and thresholds reveals that PubMedBERT and BioBERT variants, when assessed at a cosine similarity threshold of 0.7, provide the most efficient performance for identifying and retrieving chemical-gene relationships. These models demonstrate a robust balance between high precision and relatively high recall. Conversely, CBOW and SkipGram models consistently underperform across all thresholds, suggesting their inadequacy in this specialized domain. This poor performance can likely be attributed to their limited capacity to discern the nuanced semantic connections essential for accurate chemical-gene interaction predictions.

### 4.2.4 Conclusion on Capturing Functional Relatedness Across Disease-Chemical, Disease-Gene, and Chemical-Gene Pairings

There is a clear difference between the precision and recall values obtained. While some models show high precision, the recall values remain consistently low. This may be due to not accounting for synonyms, as previously noted. Additionally, as stated by Yong Hwan Kim (2019), low recall values are common in biomedical data [100].

In our comprehensive analysis across multiple cosine similarity thresholds, PubMedBERT and BioBERT consistently emerged as superior in identifying and retrieving disease-chemical, disease-gene, and chemical-gene relationships, with PubMedBERT slightly outperforming BioBERT. These models showcased robust precision, particularly at a 0.7 threshold where they also exhibited high precision and higher recall, making them optimally balanced for detailed biomedical data retrieval tasks.

Conversely, CBOW and SkipGram models consistently demonstrated inadequate performance across all assessed thresholds. Their failure to achieve significant precision or recall highlights their limitations in handling the complexity and nuance required in the biomedical semantic space. This suggests that these models are not suitable for tasks that require intricate understanding and retrieval of biomedical data, such as in the prediction of gene-disease or chemical-gene associations.

GloVe models exhibited average performance, showing moderate precision and recall that did not match the more specialized BERT variants but outperformed the CBOW and SkipGram models. While GloVe models provide a baseline utility, their lower efficacy in complex biomedical tasks suggests limited applicability in situations where maximum precision and comprehensive data retrieval are essential. This analysis emphasizes the need for employing more advanced models like PubMedBERT and BioBERT for critical biomedical applications.

## 4.3 Exploring New Biomedical Relationships in the 2024 CTD Dataset Using Word Embeddings

In 2024, the CTD introduced a new dataset. We utilized our word embedding models to examine these new data points for functional relationships. Our goal was to determine if our models, which were trained on abstracts until 2022, could identify the new functional relationships from the 2024 CTD version (i.e., those in CTD 2024 but not in CTD 2022). Using cosine similarity, we analyzed the functional relatedness among disease-gene, gene-chemical, and disease-chemical pairings. This method tested the ability of word embeddings to discover, from existing literature. biomedical associations that were not part of CTD at the time; i.e., one could consider these associations as previously unknown.

Below Table 13 provides details on the total number of newly identified relationships across three categories: disease-chemical, disease-gene, and chemical-gene. It also shows how many of these relationships were successfully captured by our word embedding models. Many relationships were not captured, primarily because the word embeddings did not represent one of the terms in the pair. The count was based on pairs having a cosine similarity score, without applying any thresholds, as the objective was to determine whether the word embeddings could capture any level of relatedness between the pairs.

*Table 13: Detection of New Relationships in the 2024 CTD Dataset by Word Embedding Models*

| Pairs | Count of New Relationships | Count of Relationships our word embeddings could capture |
|---|---|---|
| Disease-Chemical | 157 | 42 |
| Disease-Gene | 138 | 58 |
| Chemical-Gene | 191 | 83 |

*Table 14:Cosine Similarity Scores for Newly Identified Disease-Gene Relationships*

| GeneSymbol | DiseaseName | CosineSimilarity |
|---|---|---|
| nfe2l2 | liver neoplasms | 0.7374154700812473 |
| stat3 | lung neoplasms | 0.7290391618260841 |
| igf2bp1 | neuroblastoma | 0.7958682767144976 |
| mycn | neuroblastoma | 0.7638889602410818 |
| alox5 | pancreatic neoplasms | 0.7171280752904491 |

*Table 15:Cosine Similarity Scores for Newly Identified Chemical-Disease Relationships*

| ChemicalName | DiseaseName | CosineSimilarity |
|---|---|---|
| gefitinib | adenocarcinoma of lung | 0.7759782244622887 |
| methionine | adenocarcinoma of lung | 0.759424269961907 |
| urethane | adenocarcinoma of lung | 0.7438209626098845 |
| cordycepin | breast neoplasms | 0.7032463816311467 |
| dibutyl phthalate | breast neoplasms | 0.7508378985609001 |

*Table 16:Cosine Similarity Scores for Newly Identified Chemical-Gene Relationships*

| ChemicalName | GeneSymbol | CosineSimilarity |
|---|---|---|
| arsenic | akt1 | 0.8162029047580391 |
| arsenite | akt1 | 0.7907774311531406 |
| auranofin | akt1 | 0.7715376161277744 |
| baicalein | akt1 | 0.7562700893181521 |
| betanin | akt1 | 0.8023975173012093 |

Tables 14, 15, and 16 present a sample of the newly calculated cosine similarity values for disease-gene, chemical-disease, and chemical-gene relationships, respectively, identified from the 2024 release of the CTD. Each entry includes a chemical, its associated gene or disease, and the cosine similarity score, which measures the functional relatedness as captured by our word embedding models.

The exploration of the 2024 CTD dataset using our word embedding models has provided valuable insights into the dynamic landscape of biomedical relationships. Our models have

effectively identified a significant number of new disease-chemical, disease-gene, and chemical-gene associations, as detailed in Table 13. The capability of these models to discern previously unrecognized biomedical interactions underscores their importance in advancing medical research and understanding complex biological systems. Additionally, the cosine similarity scores, as shown in Table 14,15,16, further validate the models' accuracy in quantifying the strength of these new associations. Overall, the use of word embeddings in this context has proven to be a robust tool for pioneering discoveries in the ever-evolving field of biomedical sciences.

# 5 Conclusion

## 5.1 Summary of Findings

This study has rigorously assessed the application of advanced word embedding models to decipher and quantify complex relationships within the biomedical field, focusing LBD. We utilized top-tier models like PubMedBERT and BioBERT along with traditional models such as GloVe, SkipGram, and CBOW to gauge their effectiveness in recognizing both established and new biomedical relationships documented in the most recent CTD dataset.

Key findings of this thesis include:

- PubMedBERT and BioBERT models have proven to significantly outperform traditional word embedding models by achieving higher precision and recall rates, particularly at a cosine similarity threshold of 0.7. This threshold has been identified as optimal for balancing thorough and accurate biomedical data retrieval.
- Models like CBOW and SkipGram have shown limited effectiveness, struggling with the complexities and depth required for processing biomedical texts.
- GloVe models have demonstrated moderate performance, indicating their suitability for less complex biomedical tasks.

**Contributions of this Work**

The contributions of this thesis are diverse and highlight the significant role of NLP tools in advancing biomedical research:

1. **Advanced Model Evaluation:** This thesis offers an in-depth evaluation of various word embedding models, detailing their strengths and weaknesses in the context of biomedical data analysis. This comprehensive assessment is crucial for guiding

future researchers and practitioners in selecting suitable models for their specific biomedical NLP tasks.

2. **Elucidation of Functional Relatedness:** The application of word embeddings has effectively uncovered intricate functional relationships between genes, diseases, and chemicals. These insights not only bridge existing knowledge gaps but also open avenues for new discoveries, especially in fields like cancer research.
3. **Enhancement of LBD:** This research significantly enriches the field of LBD by showcasing how advanced word embedding tools can sift through extensive biomedical literature to reveal hidden connections. These discoveries have the potential to revolutionize our understanding of disease mechanisms, spur drug discovery, and foster the development of new therapies.
4. **Discovery of New Biomedical Associations:** Utilizing the latest CTD dataset to identify new biomedical relationships underscores the real-world applicability of these models. This capability enhances our understanding of biological functions and interactions, furthering the scope of medical research and diagnostics.

These contributions collectively underscore the transformative potential of advanced word embeddings in biomedical research. They provide a new perspective for analyzing extensive biomedical literature, paving the way for significant scientific advancements and innovations in medical research and practice.

## 5.2 Limitations

Although we achieved significant and promising results, we encountered several limitations during our study.

- **Handling of Synonyms**: The research faces limitations in handling synonyms within biomedical literature, which has likely contributed to the observed low recall values. Synonyms for the same biomedical term can vary widely, impacting the model's ability to consistently recognize and relate terms effectively.

- **Dataset Limitations**: The dataset utilized, consisting of PubMed abstracts, may limit the generalizability of the results to other datasets or contexts.
- **Temporal Scope:** This work identifies previously unknown relations only for the 2024 CTD version. Ideally, analysis should encompass multiple versions of the CTD to validate findings over time.
- **Domain Specificity:** The focus on cancer-related data raises uncertainty about whether the results can be generalized to other domains or health conditions.
- **Assumptions in Word Embeddings:** The assumption that proximity between two concepts in word embeddings indicates a functional relationship is intuitive, but it remains an assumption that requires further validation.

To address these limitations, future work could focus on refining pre-processing techniques to handle synonyms more effectively, as their variation likely contributed to the low recall values. Expanding the dataset beyond PubMed abstracts would improve the generalizability of the findings. Including multiple versions of the CTD in the analysis could help validate the results over time. Additionally, broadening the scope beyond cancer-related data to other health conditions would address concerns about the applicability of the findings. Finally, the assumption that proximity in word embeddings indicates a functional relationship requires further validation. Addressing these issues could improve recall, accuracy, and the broader applicability of word embedding techniques in biomedical research.

### 5.3 Key takeaways and Recommendations:

Based on the results of our comprehensive analysis, several recommendations can be made for future studies focusing on biomedical data retrieval and relationship prediction:

1. **Prioritize PubMedBERT**: PubMedBERT consistently outperformed BioBERT and other models, especially in identifying disease-chemical, disease-gene, and chemical-gene relationships. Future studies should focus on using PubMedBERT,

particularly embeddings from the last two hidden layers, which demonstrated the best performance in capturing functional relationships.

2. **Cosine Similarity Threshold**: A cosine similarity threshold of 0.7 provided the optimal balance between precision and recall, making it suitable for retrieving accurate biomedical relationships. Future work should consider applying this threshold when analyzing biomedical data.
3. **Consider Layer-Specific Embeddings**: When using BERT models like PubMedBERT or BioBERT, researchers should explore embeddings from the last few hidden layers, as these layers tend to capture more meaningful semantic relationships. This approach will likely lead to more accurate predictions in complex biomedical tasks.
4. **Avoid CBOW and SkipGram for Complex Biomedical Tasks**: The poor performance of CBOW and SkipGram models across all thresholds suggests that they are not suitable for handling the intricacies of biomedical data. Future research should avoid these models for tasks involving detailed semantic analysis, such as gene-disease or chemical-gene associations.
5. **Use GloVe as a Baseline**: GloVe models exhibited moderate performance but did not match the specialized BERT-based models. While GloVe can be used as a baseline for simpler tasks, it may not be ideal for high-precision biomedical research where more sophisticated models are necessary.

By focusing on PubMedBERT with a 0.7 cosine similarity threshold and leveraging embeddings from the final two layers, future studies can achieve more reliable results in biomedical data analysis and relationship prediction.

## 5.4 Future Work

The following areas present opportunities for further research and enhancement based on the current study:

- **Enhanced Synonym Handling**: Future methodologies could improve on the handling of synonyms by incorporating more sophisticated natural language processing techniques. Implementing a systematic approach for synonym resolution, such as using standardized taxonomy codes during preprocessing, would allow for more consistent and accurate comparisons of word embeddings with CTD pairs. This could potentially improve recall rates and overall model performance.
- **Expansion of Training Data**: Expanding the dataset to include a broader array of biomedical literature could help in enhancing the models' understanding and representation of complex biomedical relationships. Incorporating more diverse sources and data types, such as clinical trial reports and patient records, might also enhance the models' applicability and accuracy.
- **Model Optimization and Efficiency**: Investigating methods to optimize the training process for BioBERT and PubMedBERT could reduce computational demands and accelerate model training cycles. Techniques such as transfer learning, model pruning, and quantization might enable the models to maintain high performance while being less resource-intensive.
- **Development of Lightweight Models**: Developing more efficient, lightweight models could make advanced NLP tools more accessible to researchers with limited computational resources. This could involve designing custom embedding models tailored for specific biomedical applications that require less computational power.
- **Integration with Other AI Techniques**: Combining word embeddings with other AI and machine learning techniques such as graph neural networks, reinforcement learning, or generative models could uncover deeper insights into biomedical data.

This integrated approach might lead to more nuanced discoveries and applications in drug discovery, disease prediction, and other areas.

These areas of future work not only aim to address the limitations identified in the current study but also seek to expand the scope and impact of using word embeddings in biomedical research.

# Bibliography

[1] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Proceedings of the International Conference on Learning Representations.

[2] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (pp. 1532-1543). doi:10.3115/v1/D14-1162

[3] Islamaj Dogan, R., & Lu, Z. (2012). Analyzing the relationships between biomedical concepts from MEDLINE abstracts. Bioinformatics, 28(13), 206-213. doi:10.1093/bioinformatics/bts276

[4] Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific Data, 6(1), 52. doi:10.1038/s41597-019-0055-0

[5] Jagannatha, A. N., & Yu, H. (2016). Structured prediction models for RNN based sequence labeling in clinical text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 856-865). doi:10.18653/v1/D16-1082

[6] Choi, Y., Chiu, C. Y., & Sontag, D. (2016). Learning low-dimensional representations of medical concepts. In Proceedings of the AMIA Joint Summits on Translational Science (pp. 41-50). PMID: 27570647

[7] Henry, S., McInnes, B. T., & Srinivasan, P. (2017). Literature-based discovery: Models, methods, and trends. Journal of Biomedical Informatics, 74, 20-32. doi:10.1016/j.jbi.2017.08.012

[8] Lever, J., Gakkhar, S., & Gottlieb, E. (2018). A review of literature-based discovery methods. Briefings in Bioinformatics, 19(2), 278-292. doi:10.1093/bib/bbx044

[9] Beam AL, Kohane IS. Big Data and Machine Learning in Health Care. JAMA. 2018 Apr 3;319(13):1317-1318. doi: 10.1001/jama.2017.18391. PMID: 29532063.

[10] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240. doi:10.1093/bioinformatics/btz682

[11] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare, 3(1), 1-23. doi:10.1145/3458754

[12] Zhu, Y., Wang, S., Liu, X., & Zhu, H. (2019). Clinical Natural Language Processing with Word Embeddings: A Deep Learning Approach. Journal of the American Medical Informatics Association, 26(12), 1579-1590.

[13] Wei, Q., Ji, Z., Li, Z., & Duan, H. (2016). Leveraging Word Embeddings to Improve the Semantic Similarity Measure of Medical Terms. Journal of Biomedical Informatics, 61, 197-203.

[14] Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jindi, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. arXiv preprint arXiv:1904.03323

[15] Wang, Q., Li, M., Wang, S., & Ma, Q. (2019). Improved disease prediction from biomedical data using multimodal deep learning model. Scientific Reports, 9(1), 1-12.

[16] Chen, T., Wang, X., Yang, J., Shen, J., & Zhao, H. (2020). A survey on graph neural network models and applications. AI Open, 1, 57-81.

[17] Pan, X., Yan, J., & Zhao, Z. (2016). Exploring biomedical term relationships using large-scale literature data. BMC Bioinformatics, 17(1), 1-10.

[18] Visscher, P. M., et al. (2017). "10 Years of GWAS Discovery: Biology, Function, and Translation." The American Journal of Human Genetics, 101(1), 5-22.

[19] Ekins, S., Mestres, J., & Testa, B. (2007). "In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling." British Journal of Pharmacology, 152(1), 9-20.

[20] Ingelman-Sundberg, M. (2004). "Pharmacogenetics of cytochrome P450 and its applications in drug therapy: the past, present and future." Trends in Pharmacological Sciences, 25(4), 193-200.

[21] MESH Dhammi IK, Kumar S. Medical subject headings (MeSH) terms. Indian J Orthop. 2014 Sep;48(5):443-4. doi: 10.4103/0019-5413.139827. PMID: 25298548; PMCID: PMC4175855.

[22] H. Yang and Hyuck Jai Lee, "Research Trend Visualization by MeSH Terms from PubMed," International Journal of Environmental Research and Public Health, vol. 15, no. 6, pp. 1113–1113, May 2018, doi: https://doi.org/10.3390/ijerph15061113. Available: https://www.mdpi.com/1660-4601/15/6/1113. [Accessed: Mar. 09, 2024]

[23] "The Generative Lexicon," MIT Press, Jan. 12, 2024. https://mitpress.mit.edu/9780262661409/the-generative-lexicon/ (accessed Mar. 09, 2024).

[24] [1] A. Bakarov, "A Survey of Word Embeddings Evaluation Methods," arXiv.org, 2018. Available: https://arxiv.org/abs/1801.09536. [Accessed: Feb. 19, 2024]

[25] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," arXiv.org, 2016. https://arxiv.org/abs/1607.04606 (accessed Mar. 09, 2024).

[26] S. Joshua Johnson, M. Ramakrishna Murty, and I. Navakanth, "A detailed review on word embedding techniques with emphasis on word2vec," ResearchGate, Oct. 03, 2023. Available:https://www.researchgate.net/publication/374418012_A_detailed_review_on_word_embedding_techniques_with_emphasis_on_word2vec. [Accessed: Mar. 09, 2024]

[27] B. Chiu and S. Baker, “Word embeddings for biomedical natural language processing: A survey,” Language and Linguistics Compass, vol. 14, no. 12, Dec. 2020, doi: https://doi.org/10.1111/lnc3.12402.Available:https://compass.onlinelibrary.wiley.com/doi/epdf/10.1111/lnc3.12402. [Accessed: Feb. 19, 2024]

[28] X. Rong, “word2vec Parameter Learning Explained 1 Continuous Bag-of-Word Model 1.1 One-word context,” 2016. Available: https://arxiv.org/pdf/1411.2738.pdf

[29] Giovanni Di Gennaro, A. Buonanno, and Francesco, “Considerations about learning Word2Vec,” The Journal of Supercomputing, vol. 77, no. 11, pp. 12320–12335, Apr. 2021, doi: https://doi.org/10.1007/s11227-021-03743-2. Available: https://link.springer.com/article/10.1007/s11227-021-03743-2. [Accessed: Mar. 09, 2024]

[30] Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani, “Word2Vec Model Analysis for Semantic Similarities in English Words,” Procedia Computer Science, vol. 157, pp. 160–167, Jan. 2019, doi: https://doi.org/10.1016/j.procs.2019.08.153. Available: https://www.sciencedirect.com/science/article/pii/S1877050919310713?ref=pdf_download&fr=RR-2&rr=85d78e9d6eca3c0c. [Accessed: Mar. 01, 2024]]

[31] S. Joshua Johnson, M. Ramakrishna Murty, and I. Navakanth, “A detailed review on word embedding techniques with emphasis on word2vec,” Multimedia Tools and Applications, Oct. 2023, doi: https://doi.org/10.1007/s11042-023-17007-z. Available: https://link.springer.com/article/10.1007/s11042-023-17007-z. [Accessed: Mar. 09, 2024]

[32] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, “Evaluation methods for unsupervised word embeddings,” Association for Computational Linguistics, 2015. Available: https://aclanthology.org/D15-1036.pdf

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” arXiv.org, 2024. https://arxiv.org/abs/1301.3781 (accessed Mar. 09, 2024).

[34] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors,” Association for Computational Linguistics, 2014. Available: https://aclanthology.org/P14-1023.pdf

[35] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, L. E. Barnes, and D. E. Brown, “Text Classification Algorithms: A Survey,” Information, vol. 10, no. 4, pp. 150–150, Apr. 2019, doi: https://doi.org/10.3390/info10040150.

[36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” Jul. 2002. Available: https://aclanthology.org/P02-1040.pdf

[37] G. Marra, A. Zugarini, Stefano Melacci, and M. Maggini, “An Unsupervised Character-Aware Neural Approach to Word and Context Representation Learning,” Lecture Notes in Computer Science, pp. 126–136, Jan. 2018, doi: https://doi.org/10.1007/978-3-030-01424-7_13. Available: https://link.springer.com/chapter/10.1007/978-3-030-01424-7_13. [Accessed: Mar. 09, 2024]

[38] X. Rong, “word2vec Parameter Learning Explained,” arXiv.org, 2014. https://arxiv.org/abs/1411.2738 (accessed Mar. 09, 2024).

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” 2013. Available: https://arxiv.org/pdf/1310.4546.pdf

[40] A. Goldberg, “Alfred Goldberg,” Current Biology, vol. 24, no. 17, pp. R780–R782, Sep. 2014, doi: https://doi.org/10.1016/j.cub.2014.08.014.

[41] Khaled Al-Ansari, “Survey on Word Embedding Techniques in Natural Language Processing,” ResearchGate, Aug. 16, 2020.

https://www.researchgate.net/publication/343686323_Survey_on_Word_Embedding_Techniques_in_Natural_Language_Processing (accessed Mar. 10, 2024).

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv.org, 2018. https://arxiv.org/abs/1810.04805 (accessed Mar. 10, 2024).

[43] J. Lee et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, Sep. 2019, doi: https://doi.org/10.1093/bioinformatics/btz682.

[44] Y. Gu et al., "Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing." Available: https://arxiv.org/pdf/2007.15779.pdf

[45] S. Henry and B. T. McInnes, "Indirect association and ranking hypotheses for literature-based discovery," BMC Bioinformatics, vol. 20, no. 1, Aug. 2019, doi: https://doi.org/10.1186/s12859-019-2989-9. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2989-9. [Accessed: Feb. 19, 2024]

[46] A. Vaswani et al., "Attention Is All You Need," arXiv.org, 2017. https://arxiv.org/abs/1706.03762 (accessed Mar. 10, 2024).

[47] Go Eun Heo, Q. Xie, M. Song, and Jeong Hoon Lee, "Combining entity co-occurrence with specialized word embeddings to measure entity relation in Alzheimer's disease," BMC Medical Informatics and Decision Making, vol. 19, no. S5, Dec. 2019, doi: https://doi.org/10.1186/s12911-019-0934-5.

[48] S. Henry, C. Cuffy, and B. T. McInnes, "Vector representations of multi-word terms for semantic relatedness," Journal of Biomedical Informatics, vol. 77, pp. 111–119, Jan. 2018, doi: https://doi.org/10.1016/j.jbi.2017.12.006. Available:

https://www.sciencedirect.com/science/article/pii/S1532046417302769. [Accessed: Feb. 19, 2024]

[49] J. Camacho-Collados and M. T. Pilehvar, “From Word To Sense Embeddings: A Survey on Vector Representations of Meaning,” Journal of Artificial Intelligence Research, vol. 63, pp. 743–788, Dec. 2018, doi: https://doi.org/10.1613/jair.1.11259.

[50] M. Farouk, “Measuring Sentences Similarity: A Survey,” Indian Journal of Science and Technology, vol. 12, no. 25, pp. 1–11, Jul. 2019, doi: https://doi.org/10.17485/ijst/2019/v12i25/143977.

[51] A. Google, “Modern Information Retrieval: A Brief Overview.” Available: http://singhal.info/ieee2001.pdf

[52]S. Xiang, F. Nie, and C. Zhang, “Learning a Mahalanobis distance metric for data clustering and classification,” Pattern Recognition, vol. 41, no. 12, pp. 3600–3612, Dec. 2008, doi: https://doi.org/10.1016/j.patcog.2008.05.018.

[53] J. L. Rodgers and W. A. Nicewander, “Thirteen Ways to Look at the Correlation Coefficient,” The American Statistician, vol. 42, no. 1, p. 59, Feb. 1988, doi: https://doi.org/10.2307/2685263.

[54] M. Toshevska, F. Stojanovska, and J. Kalajdjieski, “The Ability of Word Embeddings to Capture Word Similarities,” International Journal on Natural Language Computing, vol. 9, no. 3, pp. 25–42, Jun. 2020, doi: https://doi.org/10.5121/ijnlc.2020.9302.

[55] “Introduction to Information Retrieval,” Stanford.edu, 2024. https://nlp.stanford.edu/IR-book/information-retrieval-book.html (accessed Mar. 10, 2024).

[56] A. Bakarov, “A Survey of Word Embeddings Evaluation Methods,” arXiv.org, 2018. https://arxiv.org/abs/1801.09536 (accessed Mar. 10, 2024).

[57] E. Agirre, E. Alfonseca, K. Hall, J. Kravalová, Marius Pasca, and Aitor Soroa, “A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches,” ACL Anthology, pp. 19–27, Jun. 2009, Accessed: Mar. 10, 2024. [Online]. Available: https://aclanthology.org/N09-1003/

[58] M. Ali, T. Zesch, and Mohamed Ben Aouicha, “A survey of semantic relatedness evaluation datasets and procedures,” Artificial Intelligence Review, vol. 53, no. 6, pp. 4407–4448, Dec. 2019, doi: https://doi.org/10.1007/s10462-019-09796-3.

[59] José Camacho-Collados, Mohammad Taher Pilehvar, and R. Navigli, “Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities,” Artificial Intelligence, vol. 240, pp. 36–64, Nov. 2016, doi: https://doi.org/10.1016/j.artint.2016.07.005.

[60] L. Com and G. Hinton, “Visualizing Data using t-SNE Laurens van der Maaten,” Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008, Available: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

[61] J. R. Hobbs, “Information extraction from biomedical text,” Journal of Biomedical Informatics, vol. 35, no. 4, pp. 260–264, Aug. 2002, doi: https://doi.org/10.1016/s1532-0464(03)00015-7.

[62] H. Xue, J. Li, H. Xie, and Y. Wang, “Review of Drug Repositioning Approaches and Resources,” International Journal of Biological Sciences, vol. 14, no. 10, pp. 1232–1244, Jan. 2018, doi: https://doi.org/10.7150/ijbs.24612.

[63] Kwang Il Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and Albert László Barabási, “The human disease network,” Proceedings of the National Academy of Sciences of the United States of America, vol. 104, no. 21, pp. 8685–8690, May 2007, doi: https://doi.org/10.1073/pnas.0701361104.

[64] Y. Goldberg and O. Levy, “word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method,” arXiv.org, 2014. https://arxiv.org/abs/1402.3722 (accessed Mar. 10, 2024).

[65] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd ed.). Prentice Hall.

[66] Everitt, B. S., & Skrondal, A. (2010). The Cambridge Dictionary of Statistics (4th ed.). Cambridge University Press.

[67] Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics, 3(1), 1-27.

[68] Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

[69] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning (ICML), 2020.

[70] “Step 3: Prepare Your Data,” Google for Developers, 2024. Available: https://developers.google.com/machine-learning/guides/text-classification/step-3. [Accessed: Jun. 20, 2024]

[71] “Wei Y, et al. (2019) | SGD,” Yeastgenome.org, 2019. https://www.yeastgenome.org/reference/S000246988 (accessed Jun. 20, 2024).

[72] Lu Z. PubMed and beyond: a survey of web tools for searching biomedical literature. Database (Oxford). 2011 Jan 18;2011. doi: 10.1093/database/baq036. PMID: 21245076; PMCID: PMC3025693.

[73] Lipscomb CE. Medical Subject Headings (MeSH). Bull Med Libr Assoc. 2000 Jul;88(3):265-6. PMID: 10928714; PMCID: PMC35238.

[74] Garmash, E., & Monz, C. (2016). Ensemble learning for multi-source neural machine translation. arXiv preprint arXiv:1606.05138.

[75] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.

[76] Henrickson, L., Hindle, A., & Stroulia, E. (2020). Analyzing the Evolution of Technical Debt Using Topic Modeling. IEEE Transactions on Software Engineering, 46(11), 1214-1232.

[77]S, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[78] Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., King, B. L., McMorran, R., Wiegers, J., & Mattingly, C. J. (2021). The Comparative Toxicogenomics Database: update 2021. Nucleic Acids Research, 49(D1), D1138-D1146.

[79] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579-2605, 2008.

[80] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Communications in Statistics, vol. 3, no. 1, pp. 1-27, 1974.

[81] W. M. S. S. T. L. He, "Learning word vectors for sentiment analysis," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014, pp. 1555-1565.

[82] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models,” arXiv.org, 2020. https://arxiv.org/abs/2010.15036 (accessed Jul. 30, 2024).

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

[84] Smalheiser, N. R., & Swanson, D. R. (1998). Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. *Computer Methods and Programs in Biomedicine, 57*(3), 149-153.

[85] Swanson, D. R. (1987). Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science, 38*(4), 228-233.

[86] Zhou, X., & He, Y. (2007). Literature-based discovery of new candidates for drug repurposing. *Briefings in Bioinformatics, 8*(1), 7-20.

[87] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).

[88] Chiu, B., Crichton, G., Korhonen, A., & Pyysalo, S. (2016). How to Train Good Word Embeddings for Biomedical NLP. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*.

[89] "Disease Genes Help | CTD," *Ctdbase.org*, 2024. https://ctdbase.org/help/diseaseGeneDetailHelp.jsp (accessed Sep. 10, 2024).

[90] Davis AP, Wiegers TC, King BL, Wiegers J, Grondin CJ, Sciaky D, Johnson RJ, Mattingly CJ. Generating Gene Ontology-Disease Inferences to Explore Mechanisms of Human Disease at the Comparative Toxicogenomics Database. PLoS One. 2016 May 12;11(5):e0155530. doi: 10.1371/journal.pone.0155530. PMID: 27171405; PMCID: PMC4865041.

[91] Davis AP, Wiegers TC, Roberts PM, King BL, Lay JM, Lennon-Hopkins K, Sciaky D, Johnson R, Keating H, Greene N, Hernandez R, McConnell KJ, Enayetallah AE, Mattingly CJ. A CTD-Pfizer collaboration: manual curation of 88,000 scientific articles

text mined for drug-disease and drug-phenotype interactions. Database (Oxford). 2013 Nov 28;2013:bat080. doi: 10.1093/database/bat080. PMID: 24288140; PMCID: PMC3842776.

[92] King, Benjamin L., et al. "Ranking Transitive Chemical-Disease Inferences Using Local Network Topology in the Comparative Toxicogenomics Database." *PLoS ONE*, vol. 7, no. 11, 7 Nov. 2012, p. e46524, https://doi.org/10.1371/journal.pone.0046524. Accessed 24 Dec. 2021.

[93] G. B. Taksler, N. L. Keating, and M. B. Rothberg, "Implications of false-positive results for future cancer screenings," *Cancer*, vol. 124, no. 11, pp. 2390–2398, Apr. 2018, doi: https://doi.org/10.1002/cncr.31271.

[94] T. F. Monaghan *et al.*, "Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value," *Medicina*, vol. 57, no. 5, p. 503, May 2021, doi: https://doi.org/10.3390/medicina57050503.

[95] A. Ramponi, Stefano Giampiccolo, D. Tomasoni, Corrado Priami, and R. Lombardo, "High-Precision Biomedical Relation Extraction for Reducing Human Curation Efforts in Industrial Applications," *IEEE Access*, vol. 8, pp. 150999–151011, Jan. 2020, doi: https://doi.org/10.1109/access.2020.3014862.

[96] Weeber, Marc & Kors, Jan & Mons, Barend. (2005). Online tools to support literature-based discovery in the life sciences. Briefings in bioinformatics. 6. 277-86. 10.1093/bib/6.3.277.

[97] Pratt W, Yetisgen-Yildiz M. A study of biomedical concept identification: MetaMap vs. people. AMIA Annu Symp Proc. 2003;2003:529-33. PMID: 14728229; PMCID: PMC1479976.

[98] Cohen, T., Schvaneveldt, R. W., & Widdows, D. (2012). "Reflective Random Indexing and indirect inference: A scalable method for discovery of implicit connections." *Journal of Biomedical Informatics*, 45(1), 1–14. DOI: 10.1016/j.jbi.2011.08.005

[99] Hulsen T, Jamuar SS, Moody AR, Karnes JH, Varga O, Hedensted S, Spreafico R, Hafler DA, McKinney EF. From Big Data to Precision Medicine. Front Med (Lausanne). 2019 Mar 1;6:34. doi: 10.3389/fmed.2019.00034. PMID: 30881956; PMCID: PMC6405506.

[100] Yong Hwan Kim and M. Song, "A context-based ABC model for literature-based discovery," *ResearchGate*, Apr. 24, 2019. https://www.researchgate.net/publication/332649115_A_context-based_ABC_model_for_literature-based_discovery (accessed Sep. 11, 2024).