International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I8P121

Volume 72 Issue 8, 212-227, August 2024

Original Article

Identification of Word Level Information Based Semantic Similarity Using Extended Glove Embeddings for Clustering and Classification Analysis

Rama Krishna Paladugu[1,3*], Gangadhara Rao Kancherla[2]

[1,2]*Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India.*
[3]*Department of Computer Science and Engineering, R.V.R. & J.C. College of Engineering, Guntur, India.*

[*]*Corresponding Author : mails4prk@gmail.com*

Received: 22 May 2024 Revised: 08 August 2024 Accepted: 17 August 2024 Published: 28 August 2024

Abstract *- In this article, an enhanced methodology for document representation and classification leveraging the Extended GloVe (ExGloVe) algorithm is presented. The ExGloVe algorithm extends the traditional GloVe model by incorporating subword information and domain-specific adaptations, addressing limitations in capturing semantic nuances and domain-specific language variations. The incorporation of subword information enables the algorithm to better represent rare and out-of-vocabulary words, enhancing the expressiveness and robustness of the embeddings. Domain-specific adaptations tailor the embeddings to specific domains, capturing domain-specific semantics and improving performance in domain-specific tasks. Document-level embeddings obtained through the aggregation process are utilized as input features for clustering algorithms such as K-Means, DBSCAN, and Hierarchical Clustering, as well as classification models including Support Vector Machine, Logistic Regression, and Neural Networks. These models leverage the semantic richness encoded in the ExGloVe embeddings for effective document analysis. Experiments with various evaluation metrics are conducted to validate the efficacy of the proposed methodology in document similarity measurement, clustering, and classification tasks.*

Keywords *- ExGloVe algorithm, Subword incorporation, Domain-specific adaptations, Document similarity measurement, clustering and classification, Natural language processing.*

1. Introduction

In Natural Language Processing (NLP), tasks such as document representation and classification are essential components that significantly contribute to various applications, including information retrieval, sentiment analysis, topic modelling, and document summarization [1]. These tasks [2] collectively empower machines to analyze and process vast amounts of textual data, thereby facilitating the discovery of trends and insights that can enhance decision-making processes and enable automation in different contexts [3]. Among the commonly utilized techniques for document representation are the traditional word embedding methods like GloVe (Global Vectors for Word Representation), which effectively encapsulate the semantic and syntactic connections between words [4].

However, these methods have research gaps in capturing the full range of semantic nuances [5] and the specific language used in different domains [1 and 2]. For example, in the medical domain, terms like "MI" could refer to "myocardial infarction" or "mitral insufficiency", and distinguishing between these meanings is crucial for accurate document representation and classification. Additionally, traditional embeddings may struggle with rare or out-of-vocabulary words [6], limiting their effectiveness in specialized domains with unique terminologies. These challenges highlight the need for enhanced word embedding methods that can better represent the complexities of language across various domains [7]. In order to address these research gaps and challenges in traditional word embedding methods like GloVe [4], in this paper, the ExGloVe algorithm was proposed, which is an advancement of the traditional GloVe model designed to enhance the representation of words in vector space. While the original GloVe model captures semantic relationships based on word co-occurrence statistics [6, 7 and 8], ExGloVe introduces two key enhancements: the integration of subword information [9 and 10] and the incorporation of domain-specific adaptations [11 and 12]. These enhancements aim to address the limitations of the traditional GloVe model in finding the full spectrum of semantic nuances and the specialized language used in different domains. The incorporation of subword information in ExGloVe allows for a more granular representation of words, enabling the model to capture morphological

similarities and handle rare or out-of-vocabulary words more effectively. This is particularly important in domains with specialized terminologies, where new terms may frequently emerge. By breaking down words into smaller subword units [9], such as character n-grams, ExGloVe can construct meaningful representations for these terms based on their subword components, enhancing the expressiveness of the embeddings. Domain-specific adaptations [11] further tailor the ExGloVe embeddings to the unique linguistic characteristics of different domains. By fine-tuning the embeddings on domain-specific corpora, the algorithm can learn the particular semantics and terminologies relevant to each domain.

This customization ensures that the embeddings are more aligned with the domain-specific language, improving their performance in tasks such as document similarity measurement, clustering, and classification within those domains. Together, these enhancements make ExGloVe a powerful tool for document representation and classification in NLP, offering improved semantic richness and domain-specific relevance compared to traditional word embedding methods.

The main objective of this study is to introduce and validate an advanced methodology for document representation and classification that leverages the ExGloVe algorithm. This approach aims to improve upon traditional word embedding methods by providing a more nuanced and domain-specific representation of textual data. The proposed methodology seeks to harness the capabilities of ExGloVe to identify both the general semantic relationships between words and the specific linguistic characteristics of different domains. Another significant objective of this study is to empirically validate the efficiency of the ExGloVe embeddings in a range of document analysis tasks.

We aim to show that ExGloVe embeddings can significantly improve performance in tasks such as document similarity measurement, where accurately capturing semantic relationships between documents is crucial. Additionally, we seek to demonstrate the utility of ExGloVe in clustering tasks, where the goal is to cluster analogous documents together based on their content [2]. Lastly, we aim to showcase the effectiveness of ExGloVe embeddings in classification tasks, where documents need to be accurately labelled into predefined categories.

Contributions: The significant contributions of this research paper are:

1.1. Development of the Extended GloVe Algorithm

This paper introduces the ExGloVe algorithm, an enhancement of the traditional GloVe model with subword information and domain-specific adaptations. By incorporating these features, ExGloVe overcomes the limitations of traditional word embeddings in handling rare words and domain-specific terminology, providing a more nuanced representation of words.

1.2. Methodology for Generating and Aggregating Word Embeddings

We present a methodology for generating and aggregating ExGloVe word embeddings for document-level representation. This approach transforms individual word embeddings into a unified document representation, enabling more effective analysis of document content and structure.

1.3. Application in Clustering Algorithms and Classification Models

The paper demonstrates the application of ExGloVe embeddings in clustering algorithms (K-Means [13], DBSCAN [14], and Hierarchical Clustering [15]) and classification models (Support Vector Machine [16], Logistic Regression [17]) for document analysis. These applications highlight the versatility and utility of ExGloVe embeddings in various document analysis tasks.

1.4. Experimental Validation and Performance Improvements

We provide experimental validation of the proposed methodology using various evaluation metrics, showcasing improvements in performance compared to traditional methods. The experiments demonstrate the effectiveness of ExGloVe embeddings in document similarity measurement [18], clustering, and classification tasks, enhancing document representation and analysis in NLP.

2. Literature Review

In this research, the evolution and current status of word embedding techniques and document analysis methods in NLP were explored. The advancements in embedding models, from traditional methods like Word2Vec [14] and GloVe [2] to recent approaches incorporating subword information and domain-specific adaptations, were identified. Additionally, popular clustering and classification techniques in NLP were examined by identifying research gaps and motivating the development of the ExGloVe algorithm. This review contextualizes the study and underscores the significance of explored contributions to the field.

2.1. Traditional Word Embedding Models and Limitations

The introduction of word embeddings has significantly impacted NLP by enabling vectors to represent words in a multidimensional space [20]. These embeddings go beyond the limitations of one-hot encoding by capturing the syntactic and semantic relationships among text words, thus enhancing the ability of machine learning algorithms to interpret text data. This advancement is clearly seen in the enhancements of several NLP tasks like sentiment analysis, machine translation, and information retrieval [21].

2.1.1. Word2Vec

Developed by [22], it is a neural network-based model that extracts word associations from large text datasets. It features two main architectures: CBOW and Skip-Gram, each targeting different contextual elements of words.

2.1.2. GloVe

Created word embeddings using global co-occurrence statistics named GloVe [4]. This model aims to incorporate both local and global contextual information into the embeddings. Recent advancements in word embeddings have seen a shift towards incorporating subword-level information to enhance the representation of words. Models such as Fast Text have pioneered this approach. Fast Text, created by [24], builds on the Word2Vec model as a collection of character n-grams that allows it to capture morphological details. Creating domain-specific word embeddings involves training word embedding models on corpora that are tailored to specific fields or industries, such as medical, legal, or finance [25]. This approach ensures that the resulting embeddings capture the unique terminology, concepts, and linguistic patterns characteristic of the domain. Techniques such as fine-tuning pre-trained embeddings on domain-specific data or incorporating domain knowledge into the training process are commonly used to enhance the domain relevance of the embeddings [26].

2.1.3. Limitations

Traditional word embedding models (i.e. Word2Vec and GloVe) have been successful in capturing general semantic relationships, but they often struggle with capturing more nuanced semantic distinctions and contextual information [27]. For instance, these models may not effectively differentiate between the various meanings of polysemous words based on context, leading to a loss of specificity in semantic representation [28]. One of the significant limitations of traditional word embedding models is their inability to effectively represent rare and Out-of-Vocabulary (OOV) words [8]. Since these models rely on large corpora to learn word representations, words that occur infrequently or not at all in the training data are either poorly represented or completely absent from the embedding space, resulting in a coverage gap in the vocabulary.

General embedding models are frequently trained on regular corpora and may not adequately capture the specialized terminology and linguistic characteristics of specific domains. For example, in the medical domain, terms like "hypertension" and "blood pressure" have specific meanings and associations that may not be accurately reflected in embeddings trained on general Text. This limitation hinders the applicability of these models in domain-specific NLP tasks.

The development of the ExGloVe algorithm is motivated by the need to overcome the limitations of traditional word embedding techniques. By incorporating subword information and domain-specific adaptations, ExGloVe aims to provide a more nuanced and context-aware representation of words, enhancing its ability to capture semantic and syntactic relationships. This approach has the potential to address the identified gaps in existing research, particularly in representing domain-specific language and handling the complexity of modern textual data.

3. ExGlove Methodology

3.1. Overview of GloVe Algorithm

The GloVe algorithm, created by [4], relies on the idea that word co-occurrence probabilities in a large corpus can uncover their semantic connections.

3.1.1. Objective Function

GloVe's objective function to reduce the dot production of embedding and co-occurrence log probability is expressed as follows:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{i,j})^2 \qquad (1)$$

Where the w_i and $\tilde{w}_j$denotes the word embeddings for words and correspondingly represents the biased terms for words and the count of words$i$co-occurs with the word$j$in a given context window in the corpus. At the same time, it $f(X_{ij})$ stands for a weighting function that adjusts the contribution of each co-occurrence to the objective function, typically to prevent overemphasis on rare or overly frequent co-occurrences using the size of the vocabulary 'V'.

3.1.2. Generation of Word Embeddings

The word embeddings produced by the GloVe objective function reflect the co-occurrence frequency of word pairs within a given context window. The algorithm then factorizes the co-occurrence matrix to produce lower-dimensional word embeddings. These embeddings are intended to capture both the global statistical information of word co-occurrences and the local context of words within the corpus. The word embeddings resulting from GloVe are capable of encoding a wide array of semantic and syntactic relationships between words [4].

The traditional GloVe algorithm, while powerful in capturing semantic relationships between words, has certain limitations that can impact its effectiveness in specific scenarios:

i) GloVe embeddings are based on co-occurrence statistics, which can be sparse for rare words [7]. This sparsity can lead to less informative embeddings for such words, limiting the algorithm's ability to capture their semantics accurately.
ii) The embeddings generated by GloVe are general-purpose and may not fully capture the nuances of language

specific to certain domains [27], such as medical or legal terminology. This can affect the performance of GloVe in domain-specific NLP tasks.

3.2. Extended GloVe Algorithm

To address these limitations [7, 8, 27, 28] in traditional GloVe algorithm, this research proposes the Extended GloVe Algorithm, which incorporates the "Subword Information" and "Domain-Specific Adaptations" to improve the performance in document similarity measurement, clustering, and classification(shown in figure-1). By integrating subword information [24], such as character n-grams or morphemes, into the GloVe embeddings, the algorithm can better represent rare and out-of-vocabulary words. This extension enables the embeddings to capture finer-grained semantic information, improving their expressiveness and robustness. Adapting the GloVe algorithm to specific domains [25 and 26] involves training the embeddings on domain-specific corpora and incorporating domain-relevant vocabulary. This customization ensures that the embeddings are more aligned with the language and semantics of the target domain, enhancing their effectiveness in domain-specific NLP tasks.

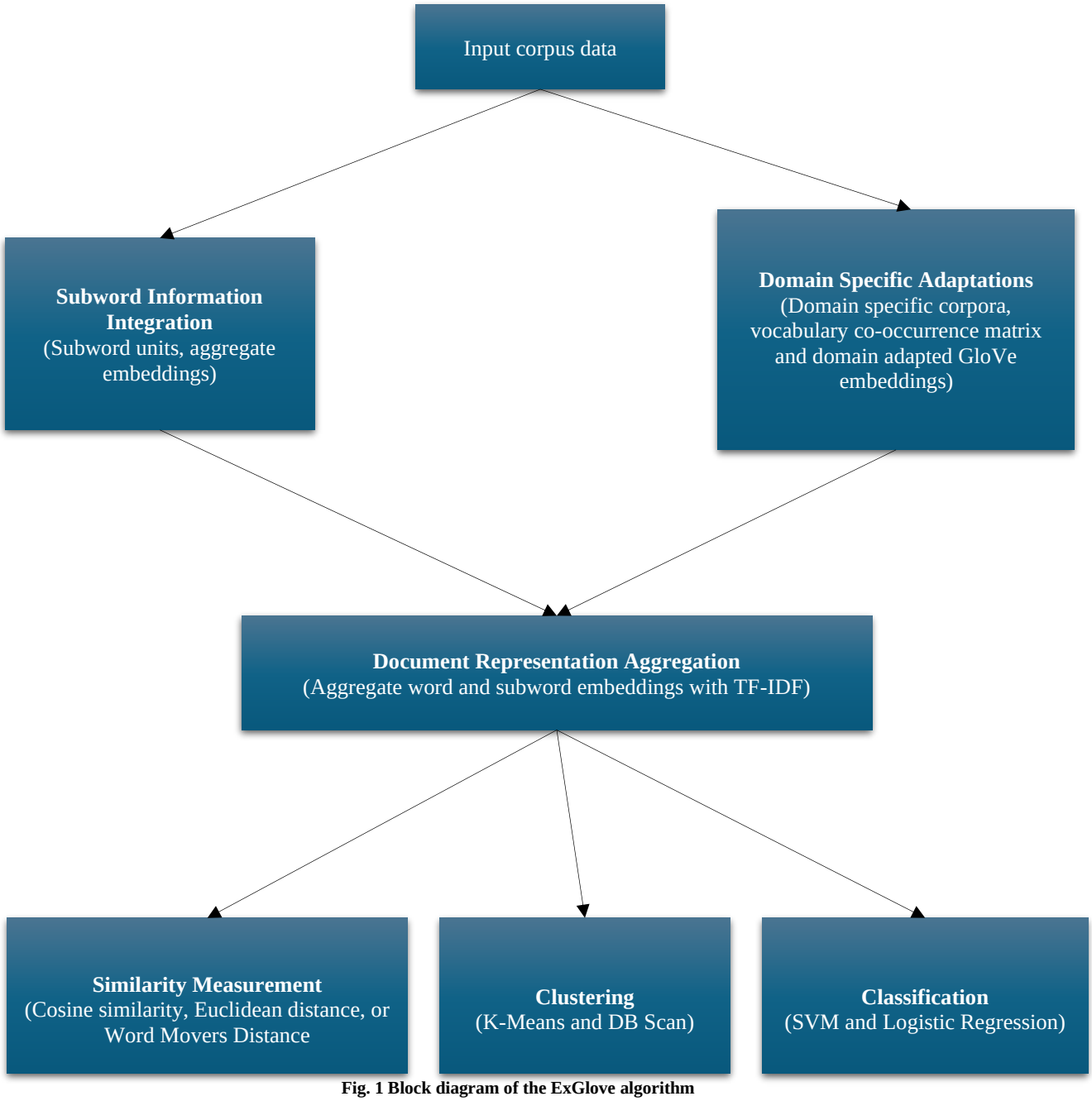

Fig. 1 Block diagram of the ExGlove algorithm

Extended GloVe Algorithm

Input

Set of documents($Corpus$), Vocabulary (V), Word embeddings (W_e), Embedding Dimension(d_e), Context Window Size (CW_{size}), Threshold Clustering (δ_{Clu}), Threshold Classification (δ_{Cla})

Output

Document Embeddings(D_e)

BEGIN

Initialization

Initialize (d_e) and (CW_{size}) from Input

Subword Information Integration:

For each word(w_i) in the $Corpus$:

Represent (w_i) as subword units S_{w_i} (e.g., character n_{grams}, morphemes)

$$S_{w_i} = \{s_1, s_2, .., s_n\}$$

Aggregate subword embeddings to obtain word embeddings

$$W_e = \frac{1}{m}\sum_{k=1}^{m} S_{w_k}$$

Domain Specific Adaptations:

Select domain-specific corpora(C_d) and vocabulary(V)

Construct domain-specific co-occurrence matrix(M_d) based on (CW_{size})

Adapt co-occurrence matrix(M_d) to domain

Train domain-adapted GloVe embeddingsW_e using optimized objective function$J$and d_e

Document Representation Aggregation:

For each document d_j in the $Corpus$:

$$\left(D_e[j] = \frac{1}{n}\sum_{i=1}^{n} W_e[i]\right)$$

Use averaging or weighted sum based on TF-IDF scores

$$TF\_IDF_{score} = \left(\frac{TF}{IDF}\right)$$

Similarity Measurement:

Compute document similarity using cosine similarity, Euclidean distance, or Word Mover's Distance.

$$Cos_{sim} = \left(\frac{Doc1_{embed} \cdot Doc2_{embed}}{||Doc1_{embed}|| \cdot ||Doc2_{embed}||}\right)$$

Clustering:

Apply clustering algorithms such as K-Means and DBSCAN to group similar documents

$if\,(Cos_{sim} \geq \delta_{clu})$:

Assign documents to the same cluster.

Classification:

Use classification models (e.g., SVM, Logistic Regression) with document-level embeddings as input features.

$if\,(Cos_{sim} \geq \delta_{cla})$:

Classify documents into respective categories.

END

3.2.1. Incorporation of Subword Information

The incorporation of subword information into the Extended GloVe Algorithm represents a vital enhancement that directly addresses the challenges associated with representing rare or Out-of-Vocabulary (OOV) words [8]. This enhancement holds particular significance across a broad spectrum of NLP tasks, including document similarity measurement, clustering, and classification. In these contexts, the ability to construct comprehensive and semantically rich word representations is a high priority. By integrating subword information, the ExGloVe algorithm gains the capacity to capture the finer nuances of language, enabling a more accurate representation of words which rarely appear in the training corpus.

This, in turn, leads to embeddings that are more informative and reflective of the true semantic relationships between words [25]. As a result, the extended embeddings improve the performance of NLP tasks that rely on the precision of word representations, facilitating more accurate document clustering, more effective classification algorithms, and more nuanced measurements of document similarity.

Subword Integration Methodology: The methodology for subword integration in the ExGloVe Algorithm involves decomposing the words into smaller units, such as character n-grams or morphemes [24]. For example, the word "unbelievable" can be broken down into subword units like:

character bi-grams: {"un", "nb", "be", "el", "li", "ie", "ev", "va", "ab", "bl", "le"}
character tri-grams: {"unb", "nbe", "bel", "eli", "lie", "iev", "eva", "vab", "abl", "ble"}
morphemes: {"un", "believe", "able"}

Each subword unit is represented as a vector, and the final word embedding is computed by aggregating these subword embeddings. This aggregation method enhances word representation, especially for rare or out-of-vocabulary words [5], by leveraging semantic information from subword components. If $S_w = \{s_1, s_2, \ldots, s_k\}_{\text{it}}$ denotes the set of subword units for a word $'w'$, then the aggregated embedding v_w is calculated as:

$$v_w = \frac{1}{|S_w|} \sum_{s \in S_w} v_s \qquad (2)$$

Where v_s is the embedding of the subword s, and S_w is the number of subword units in the word $'w'$. Consider the above word "unbelievable" and its decomposition into character tri-grams. The aggregated embedding for "unbelievable" would be the average of the embeddings of its tri-grams: {"unb", "nbe", "bel", "eli", "lie", "iev", "eva", "vab", "abl", "ble"}. This subword integration enhances the representation of words, especially those that are rare or out-of-vocabulary [5], by leveraging the semantic information contained in their subword components.

This approach allows the embeddings to inherit semantic information from these subwords, which often have more robust statistics due to their occurrence in other words. For a rare word r with subwords S_r, its embedding v_r can capture its meaning more effectively through the aggregation of subword embeddings:

$$v_r = \frac{1}{|S_r|} \sum_{s \in S_r} v_s \qquad (3)$$

Incorporating subword information into GloVe embeddings offers several advantages, such as Improved Coverage, Enhanced Semantic Richness and Robustness to sparsity for document similarity measurement, clustering, and classification.

3.2.2. Domain-Specific Adaptations

Adapting the GloVe algorithm to a specific domain is a fundamental component of the ExGloVe algorithm, as it enables the creation of embeddings that are finely attuned to the intricacies and subtleties of a specific field or area of study [26]. In NLP tasks like document similarity assessment, clustering, and classification, the success of the algorithms frequently depends on their capability to capture and represent the semantic relationships between words precisely. General-purpose word embeddings, while useful, may not adequately reflect the specialized language, terminology, and semantic structures prevalent in specific domains such as medicine, law, finance, or technology [27]. This can lead to suboptimal performance in NLP tasks, as the embeddings may fail to capture the nuances and distinctions that are critical in these domains.

By adapting the GloVe algorithm to a specific domain, the resulting embeddings are enriched with domain-specific semantic information, leading to more precise and meaningful representations of words and phrases. Domain-adapted embeddings significantly increase the performance of NLP tasks (i.e. document similarity measurement), where capturing the subtle differences between domain-specific terms is crucial [28]; clustering, where documents need to be grouped based on domain-relevant themes; and classification, where the ability to distinguish between domain-specific categories is key. The use of domain-specific adaptations ensures that the embeddings are directly relevant to the task at hand, making them more applicable and useful for domain-specific NLP applications. Domain-specific embeddings facilitate the extraction of insights, patterns, and trends unique to that domain, thereby increasing the value and effectiveness of NLP analysis. Adapting the GloVe algorithm to a specific domain involves several phases: Domain-Specific Corpora Selection, Domain-Specific Vocabulary Incorporation, Co-occurrence Matrix Adaption, Domain-Adapted Embeddings Training and Domain-Specific Semantics Incorporation.

Domain-Specific Corpora Selection

The selection of a domain-specific corpus is a critical first step in adapting the GloVe algorithm to a specific domain. This process involves identifying and choosing a corpus that accurately reflects the language, terminology, and semantic structures characteristic of the domain in question. The chosen corpus serves as the foundation for training the domain-adapted GloVe embeddings, and therefore, its representativeness and comprehensiveness are crucial.

For domain-specific corpora selection, the selection criteria contain a set of standards:

- Representativeness: The corpus should be representative of the language used in the domain, including domain-specific terminology, jargon, and linguistic patterns.
- Comprehensiveness: The corpus should be large enough to encompass the breadth of vocabulary and semantic

relationships prevalent in the domain. This ensures that the trained embeddings capture a wide range of domain-specific concepts and nuances.
- Quality: The corpus should be of high quality, with minimal noise and irrelevant content. Clean and well-curated corpora lead to more accurate and reliable embeddings.

Some example datasets for the domain-specific corpora are Medical Domain (PubMed and Clinical Notes), Legal Domain (Legal Documents and Case Law Databases) and Financial Domain (Financial News Articles and Transaction Records). While there are no specific equations for selecting a domain-specific corpus, certain quantitative metrics can be used to assess the suitability of a corpus, such as Vocabulary Coverage and Co-occurrence Diversity. Vocabulary Coverage is estimated based on the percentage of domain-specific terms covered by the corpus. A higher coverage indicates better representativeness. Co-occurrence Diversity is the diversity of word co-occurrences in the corpus [34], measured using metrics such as entropy. A higher diversity suggests a more comprehensive semantic landscape.

Domain-Specific Vocabulary Incorporation

After selecting an appropriate domain-specific corpus, the next crucial step in adapting the GloVe algorithm is to incorporate domain-specific vocabulary into the training process. This is crucial for ensuring that the resulting word embeddings accurately reflect the specialized language and semantics of the domain.

The incorporation of domain-specific vocabulary involves:
- Extraction of Technical Terms: Analyzing the corpus to identify technical terms, jargon, and acronyms which are widely used in the domain. For a medical domain instance, in the medical domain, terms like "angioplasty" or "myocardial infarction" should be included.
- Inclusion of Relevant Concepts: Ensuring that the vocabulary encompasses a broad range of concepts, entities, and relationships relevant to the domain. This may include specific procedures, diseases, drugs, legal statutes, financial instruments, etc.

The process of incorporating domain-specific vocabulary can be quantitatively supported by term frequency analysis (TFA) and term relevance scoring (TRS). TFA computes the occurrence of each word in the corpus to identify important domain-specific terms. Terms with high frequency are more likely to be relevant to the domain.

Let $'C'$ be a corpus containing $'N'$ documents, and let $'t'$ be a term present in the corpus. In this case, the term frequency $TF(t,C)$ of term $'t'$ in the main corpus $'C'$ can be calculated as:

$$TF(t,C) = \frac{no\ of\ occurences\ of\ term\ t\ in\ corpus\ C}{total\ no\ of\ terms\ in\ corpus\ C} \quad (4)$$

Alternatively, to calculate the term frequency for each document and then aggregate it for the entire corpus using:

$$= \sum_{d \in C}^{TF(t,C)} \frac{no\ of\ occurences\ of\ term\ t\ in\ document\ d}{total\ no\ of\ terms\ in\ document\ d} \quad (5)$$

In this equation, the term frequency for each document is determined by dividing the number of times the term $'t'$ appears in the document $'d'$ by the total number of terms in that document. Then, these frequencies are summed up across all documents in the corpus to get the overall term frequency of a particular term $'t'$ in the given corpus $'C'$. High term frequency values indicate that a term is frequently used within the corpus and may be an important domain-specific term. TRS assigns a relevance score to each term of the document based on its frequency and its co-occurrence patterns with other domain-specific terms. To define the term relevance scoring, we incorporate both the term frequency and the co-occurrence factor. Let us assume/is the term for which we want to calculate the relevance score R_t, and $'C'$ is the corpus. The co-occurrence factor for the term $'t'$, denoted as CF_t, can be defined based on its proximity to other known domain-specific terms. The relevance score R_t for term $'t'$ can be calculated as:

$$R_t = \frac{TF(t,C)}{N} \times CF_t \quad (6)$$

Where $TF(t,C)$ is the term frequency of term $'t'$ in corpus $'C'$ and $'N'$ is the total number of terms in corpus $'C'$. CF_t is the co-occurrence factor for term $'t'$, which measures the frequency of term $'t'$ appearing in close proximity to other known domain-specific terms. The co-occurrence factor CF_t can be further defined as:

$$CF_t = \frac{no\ of\ times\ term\ t\ cooccurs\ with\ domain\ specific\ terms}{total\ cooccurs\ of\ all\ terms\ with\ domain\ specific\ terms} \quad (7)$$

In this equation, the numerator represents the number of times term $'t'$ appears near other domain-specific terms within a certain context window (e.g., within a paragraph or sentence), and the denominator represents the total number of co-occurrences of all terms with domain-specific terms in the corpus. By combining term frequency and co-occurrence patterns, the relevance score R_t provides a measure of how important and relevant the term $'t'$ is within the specific domain, taking into account both its frequency and its association with other domain-specific terms.

3.2.3. Co-Occurrence Matrix Adaption

In the ExGloVe Algorithm, adapting the co-occurrence matrix to a specific domain is a crucial step that ensures the resulting embeddings are tailored to the semantic landscape of the domain. The co-occurrence matrix [34] is a key element of

the GloVe algorithm, as it records the statistics based on their co-occurrences within a text corpus. For a domain-specific corpus, the co-occurrence matrix $'X'$ is constructed such that each element X_{ij} denotes the number of times the co-occurrence of words $'i'$ and $'j'$ within a specified context window. Mathematically, this can be expressed as:

$$\boldsymbol{X_{ij}} = \sum_{\boldsymbol{d} \in \boldsymbol{C}} \sum_{\boldsymbol{w} \in \boldsymbol{D}} \boldsymbol{Count_{ij}}(\boldsymbol{w}, \boldsymbol{d}) \quad (8)$$

Where the $'C'$ is the domain-specific corpus consisting of documents $'d'$ and $'w'$ is a word in document $'d'$, $Count_{ij}\ (w,d)$ is a function that counts the number of times the word $'i'$ and word $'j'$ co-occur within the specified context window around word $'w'$ in document $'d'$. The context window size determines how many words surrounding the target word are considered for co-occurrence counting, which is an important parameter that can influence the semantic relationships captured by the matrix. To ensure that the co-occurrence matrix reflects the effectiveness of the domain-specific context, we applied the normalization and weighting techniques:

As part of normalization, each element X_{ij} will be normalized by the total count of co-occurrences in the matrix to ensure that the values are proportional and comparable:

$$X_{ij}^{norm} = \frac{X_{ij}}{\sum_{k,l} X_{kl}} \quad (9)$$

Similarly, in weighting, the weighting function $f(X_{ij})$ can be applied to each element to adjust the importance of different co-occurrences. For example, rarer co-occurrences might be given more weight to highlight their significance in the domain:

$$X_{ij}^{weighted} = f(X_{ij}) \cdot X_{ij} \quad (10)$$

By constructing the co-occurrence matrix using a domain-specific corpus, the matrix captures the unique co-occurrence patterns and semantic relationships that are characteristic of the domain. For example, let us consider two terms relevant to the medical domain: "systolic" (denoted as $'i'$) and "hypertension" (denoted as $'j'$). The co-occurrence frequency of these terms within the context window is captured by the matrix element X_{ii}:

$$\boldsymbol{X_{ij}} = \boldsymbol{Count}(\boldsymbol{i}\ \&\ \boldsymbol{j}\ \boldsymbol{co-occur\ in\ corpus\ C\ context\ windows}) \quad (11)$$

The semantic association between these terms can be quantitatively analyzed by comparing their co-occurrence frequency X_{ij} to their individual frequencies X_{ii} (frequency of "systolic") and X_{jj} (frequency of "hypertension"):

$$Semantic\ Association(i,j) = \frac{X_{ij}}{\sqrt{X_{ii} \times X_{jj}}} \quad (12)$$

This equation represents the normalized co-occurrence frequency of the terms "systolic" and "hypertension," which reflects their semantic association in the medical domain. A higher value indicates a stronger semantic relationship between the terms, as observed in the domain-specific corpus. Domain-specific co-occurrence matrix construction allows for the quantitative analysis of semantic relationships between terms that are characteristic of the domain, as illustrated by the example of "systolic" and "hypertension" in the medical domain.

3.2.4. Domain-Adapted Embeddings Training

After establishing the domain-specific corpus and vocabulary, the GloVe algorithm is trained to produce embeddings customized for the domain. The training process optimizes the GloVe objective function, which is adapted to the domain-specific co-occurrence matrix and vocabulary. Details of the objective function are outlined in the section. In training, the optimization process for the domain-adapted GloVe embeddings involves minimizing the objective function $'J'$ with respect to the word embeddings w_i $\tilde{w}_j$and bias termsb_i $\tilde{b}_j$. This is typically achieved using optimization algorithms [19].

The optimization process continues until a convergence criterion is met, such as a specified number of iterations, a threshold for the change in the objective function value, or a threshold for the magnitude of the gradients. After the optimization process, the resulting word embeddings w_i and $\tilde{w}_j$ are the domain-adapted GloVe embeddings that capture the semantic relationships and nuances specific to the chosen domain. The domain-adapted embeddings are highly valuable for NLP tasks such as document similarity measurement [13], clustering, and classification within the specific domain, as they provide a more accurate representation of the domain-specific language and semantics, leading to improved performance and more meaningful results.

3.2.5. Domain-Specific Semantics Incorporation

The incorporation of domain-specific semantics into the GloVe embeddings is a crucial aspect of the domain adaptation process. This step ensures that the embeddings capture the unique semantic relationships and nuances characteristic of the domain-specific vocabulary. In general, the domain-adapted GloVe contains each dimension of the embedding vector related to a latent semantic feature [9].

For a word $'i'$ in the domain-specific vocabulary, its embedding w_i captures the semantic properties of the word based on its co-occurrence patterns in the domain-specific corpus. The geometric properties of word embeddings reveal their semantic relationships. For instance, the correspondence between two words $'i'$ and $'j'$ can be assessed using the cosine similarity of their embeddings:

$$cosine\ similarity(w_i, w_j) = \frac{w_i^T w_j}{||w_i||\ ||w_j||} \quad (13)$$

Where $|\ w_i\ |$ and $|\ w_j\ |$ are the norms of the embeddings w_i and w_j, respectively. Furthermore, the embeddings can capture more complex semantic relationships, such as analogies. For instance, if $w_{king}, w_{man}, w_{queen}$ and w_{woman} are the embeddings for the words "king," "man," "queen," and "woman," respectively, then the following relationship is expected to hold:

$$w_{king} - w_{man} + w_{woman} \approx w_{queen} \quad (14)$$

In the context of domain adaptation, the embeddings are trained to capture the semantics specific to the domain. For example, in the medical domain, the embedding for the term "cardiomyopathy" should be close to terms like "heart," "disease," and "muscle" in the embedding space, reflecting its semantic association with these concepts in the medical field.

The incorporation of domain-specific semantics into the embeddings can accurately capture the semantic similarity between documents based on their domain-specific content. These domain-specific semantics will progress the performance of clustering and classification models by providing features that are semantically informative within the domain.

3.3. Aggregation of Embeddings for Document Representation

Once the word and subword embeddings are obtained through the ExGloVe algorithm, the next step is to aggregate these embeddings to form document-level representations. This aggregation process combines the embeddings of individual words and subwords in a document into a single vector that encapsulates the entire document's semantic content.

3.3.1. Averaging Method

A common and straightforward approach for embedding aggregation is to average the embeddings of all words and subwords in the document. For a document $'d'$ containing $'n'$ words, where each word w_i has an embeddingv_{w_i}, the document-level embedding $v_d$is evaluated using the following equation:

$$\boldsymbol{v_d = \frac{i}{n}\sum_{i=1}^{n} v_{wi}} \quad (15)$$

This method is computationally efficient and easy to implement. However, one drawback is that it treats all words equally, which might not be ideal since some words could be more important than others in conveying the document's meaning.

3.3.2. Weighted Sum Based on TF-IDF Scores

To address the shortcomings of simple averaging, a weighted sum technique can be employed. Here, each word's embedding is weighted according to its Term Frequency and the relevant Inverse Document Frequency score, which reflects the word's importance in a specific document relative to its frequency in the overall document collection. For a word w_i in the document$'d'$, its TF-IDF score is denoted as$TF - TDF_{w_i,d}$. The document-level embedding $v_d$using the weighted sum method is calculated as follows:

$$v_d = \frac{\sum_{i=1}^{n} TF - IDF_{w_i,d} \cdot v_{w_i}}{\sum_{i=1}^{n} TF - IDF_{w_i,d}} \quad (16)$$

In this formula, the numerator sums up the embeddings of all words in the document, with each embedding multiplied by its corresponding TF-IDF score. The denominator aggregates all the TF-IDF scores in the document, acting as a normalization factor to ensure that the overall embedding magnitude remains consistent regardless of document length.

The weighted sum approach based on TF-IDF scores provides a more nuanced document representation by giving more weight to embeddings of words that are more relevant to the document's context. This enhanced document-level embedding proves advantageous for various NLP applications, including document similarity measurement, clustering, and classification.

3.4. Similarity Measurement

Once document-level embeddings are obtained through the ExGloVe algorithm with subword information incorporation, similarity measurement between documents can be performed using various metrics [15]. These metrics are used to assess how well the embeddings capture semantic similarity.

3.4.1. Cosine Similarity

This metric is commonly utilized to evaluate the correspondence between two vectors. It is particularly applicable for comparing high-dimensional, sparse data, such as text data represented through embeddings like those produced by the ExGloVe algorithm [18]. Calculating cosine similarity within two document embeddings v_d and $v_{d'}$, which are derived from the ExGloVe algorithm, is defined as:

$$\text{cosine similarity}\ (v_d, v_{d'}) = \frac{v_d \cdot v_{d'}}{||v_d|| \cdot ||v_{d'}||} \quad (17)$$

Where $' \cdot '$ denotes the dot product of the vectors and $'|\cdot|'$denotes the Euclidean norm (or length) of the vector. This metric is particularly effective in capturing the angular similarity between document embeddings generated by the ExGloVe algorithm.

3.4.2. Euclidean Distance

This metric is used to assess the dissimilarity between two document embeddings generated by the ExGloVe algorithm with Euclidean embeddings v_d and $v_{d'}$is given by:

$$\text{Euclidean distance}(v_d, v_{d'}) = ||v_d - v_{d'}|| \quad (18)$$

In the context of document similarity, a smaller Euclidean distance indicates higher similarity. This metric is useful for capturing the overall magnitude of difference between document embeddings.

Word Mover's Distance (WMD)

This metric measures the minimum amount of "travel" needed to align the words in one document with the words in another document. It is particularly relevant for the ExGloVe algorithm as it takes into account the individual word embeddings, including subword information, to compute the distance. For documents represented by their word embeddings generated by the ExGloVe algorithm, WMD is defined as:

$$WMD(d, d') = \min_{T \geq 0} \sum_{i,j} T_{ij} \cdot ||v_{w_i} - v_{w_j^i}|| \quad (19)$$

Where T_{ij} is the "flow" of the word $'i'$ in the document $'d'$ to word $'j'$ in the document $'d'$, v_{w_i} and $v_{w_j^i}$ are the embeddings of the corresponding words. WMD is a powerful metric for capturing semantic differences between documents, especially when the documents have few words in common, by leveraging the fine-grained semantic information encoded in the ExGloVe embeddings.

3.5. Clustering Algorithms

These algorithms group similar documents based on their content. When using the ExGloVe algorithm for document representation, the document-level embeddings serve as input for different clustering algorithms. In the ExGloVe experiments, we chose three well-known clustering methods: K-Means, DBSCAN, and Hierarchical Clustering.

3.5.1. K-Means

This widely used clustering algorithm [13] divides data into clusters with the goal of minimizing the variance within each cluster. Using ExGloVe embeddings, the K-Means will perform the document clustering as:

- *Init*: Select $'K'$ initial centroids $\{c_1, c_2, \ldots, c_K\}$ from the document embeddings $\{v_1, v_2, \ldots v_N\}$.
- *Assign*: Assign each document embedding v_i to the nearest centroid, forming clusters:
$$Cluster(v_i) = argmin_k \ ||v_i - c_k||$$
- *Update*: Recompute the centroids as the mean of the embeddings in each cluster:
$$c_k = \frac{1}{|\text{ Cluster }_k| \sum_{v_i \in \text{Cluster }_K} v_i} \quad (20)$$
- *Iterate*: Continue the assignment and update steps.

Elbow Method or the Silhouette Score used for optimal cluster detection.

3.5.2. DBSCAN

This density-based clustering algorithm [14] forms clusters based on the density of data points, classifying sparse points as outliers. The DB Scan for document clustering using ExGloVe embeddings:

Parameters: Set the radius $'\varepsilon'$ of the neighborhood and the least points $MinPts$ for the dense region.

Core Points: The document embedding v_i is a core point if:

$$\left|\left\{v_j : \left||v_j - v_i|\right| \leq \varepsilon\right\}\right| \geq MinPts \quad (21)$$

Clusters: Form clusters by connecting core points that are within $'\varepsilon'$ a distance of each other.

Outliers: Mark points that are not part of any cluster as outliers.

3.5.3. Hierarchical Clustering

This approach builds a cluster hierarchy [15] using either a bottom-up (agglomerative) or top-down (divisive) method. For document clustering with ExGloVe embeddings:

- *Initialization*: Start with each document embedding v_i as its cluster.
- *Agglomeration*: Recursively merges the nearest cluster pairs based on a linkage criterion until all embeddings form a single cluster:
$$Dist(Cls_a, Cls_b) = min_{v_i} \ \in Cls_a, v_j \in Cls_b ||v_i - v_j||$$
- *Dendrogram*: The merging process can be visualized as a dendrogram, showing the hierarchical relationship between clusters.
- *Cluster Selection*: Determine the optimal number of clusters by a threshold.

In each of these clustering algorithms, the ExGloVe embeddings provide a semantically rich representation of the documents, enabling the algorithms to group documents based on their underlying semantic content effectively.

3.6. Classification Models

For document classification using the ExGloVe algorithm, the document-level embeddings obtained from the aggregation process serve as input features to various classification models. Here, we discuss the application of three popular classification models: SVM [16] and Logistic Regression [17].

3.6.1. Support Vector Machine (SVM)

Fine-tuning the hyperplane based SVM model [16] for document classification using ExGloVe embeddings:

Feature Representation

Represent each document as a vector v_d obtained from the ExGloVe embeddings.

Model Training

Train the SVM to determine the best hyperplane by optimizing the following:

$$\min_{w,b} \frac{1}{2}|W|^2 + C \sum_{i=1}^{N} \xi_i \quad (22)$$

$$y_i\left(W^T v_{d_i} + b\right) \geq 1 - \xi_i, \xi_i \geq 0 \quad (23)$$

Where $'W'$ represents the weight vector, $'b'$ represents bias

term, $'C'$ is considered as the regularization parameter, ξ_i are the slack variables, y_i are the class labels, and $'N'$ is the number of documents.

Classification: New documents are classified based on the decision function's signature $W^T v_{d_i} + b$.

3.6.2. Logistic Regression

This straightforward logistic [17] classification technique predicts class probabilities using a logistic function. For document classification using ExGloVe embeddings:

Feature Representation

Document vector v_d obtained from the ExGloVe embeddings.

Model Training

The logistic regression model is trained to predict a specific class:

$$P\left(y = \frac{1}{v_d}\right) = \left(\frac{1}{1 + e^{-(w^T v_d + b)}}\right) \quad (24)$$

Where $'W'$ denotes the weight vector, $'b'$ stands for the bias term, and $'y'$ is the binary class label.

Classification: New documents are classified based on probability $P(y = 1/v_d)$. In each of these classification models, the ExGloVe embeddings serve as features. Adjustments to the models may include tuning hyperparameters (e.g., regularization parameter in SVM and logistic regression) to optimize performance with the embeddings.

4. Results and Discussions

4.1. Selection of Domain-Specific Datasets

To visualize the results, several representative datasets from the medical and legal domains were considered to evaluate the ExGloVe Algorithm with "Subword Information" and "Domain-Specific Adaptations". In the medical domain, MIMIC-III [24], a large dataset from critical care units was utilized, and i2b2 NLP Challenge Datasets [25] for various NLP tasks. For the legal domain, COLIEE [26] was employed, focusing on Legal Information Extraction/Entailment and LCRD [27], which includes annotated legal case reports. These datasets offer diverse content and annotations, enabling thorough testing of algorithmic performance in domain-specific tasks.

4.2. Data Preprocessing

For medical domain datasets (MIMIC-III 24] and i2b2 [25]), tokenization, normalization (lowercasing), and stopword removal were conducted to prepare the data for analysis. Legal domain datasets (COLIEE [26] and LCRD [27]) underwent similar preprocessing, with additional removal of legal jargon and irrelevant sections. Standardization of legal citations was also performed to ensure dataset consistency.

4.3. Experimental Setup

For the experiments, the ExGloVe model with an embedding dimension of 300 and a context window size of 5 to balance local context and noise avoidance [13] was configured. We integrated subword information using the Fast Text approach [21], enriching the vocabulary with word n-grams to handle out-of-vocabulary words and capture semantic similarities more effectively. Domain-specific adaptations were achieved through fine-tuning medical and legal datasets using transfer learning, allowing the model to learn domain-specific nuances [21]. For comparison, baseline models included the original GloVe model [2], FastText [21], domain-specific word embeddings (DSWE) [11], and BiLSTM [28].

These comparisons aimed to evaluate the effectiveness of the ExGloVe Algorithm's enhancements across various NLP tasks in the medical and legal domains. In evaluating the embeddings, we utilized various metrics, including Cosine Similarity (CosSim), Silhouette Score (SiL), precision, recall, F1-score, ROC curve, ARI, NMI, and processing time (P_time) [2 and 4]. Cosine similarity measures vector similarity, while silhouette score assesses clustering quality. Precision, recall, and F1-score evaluate classification accuracy, and the ROC curve analyzes binary classification balance. ARI and NMI quantify clustering similarity and shared information, respectively, adjusted for chance. Processing time is crucial for assessing computational efficiency in real-time or time-sensitive applications.

4.4. Results and Analysis

4.4.1. Document Similarity Measurement

Our experiments evaluated the document similarity measurement results for the MIMIC-III, i2b2, COLIEE, and LCRD datasets, showcasing the performance of several models, including ExGloVe, BiLSTM, DSWE, FastText, Word2Vec, and GloVe are shown in Table 1, Table 2 and Figure 2.

Table 1. Document similarity measurement results for MIMIC-III and i2b2 datasets

	MIMIC-III				i2b2			
Model	**CosSim**	**EuclDist**	**WMDist**	**ROC/AUC**	**CosSim**	**EuclDist**	**WMDist**	**ROC/AUC**
ExGloVe	0.87	0.65	0.42	0.93	0.84	0.67	0.45	0.91
BiLSTM [28]	0.83	0.68	0.47	0.89	0.80	0.71	0.49	0.87
DSWE [11]	0.79	0.72	0.50	0.85	0.76	0.74	0.52	0.82
FastText [21]	0.58	0.70	0.48	0.64	0.52	0.72	0.48	0.60
Word2Vec [14]	0.62	0.74	0.52	0.71	0.59	0.78	0.54	0.63
GloVe [2]	0.81	0.71	0.49	0.86	0.77	0.75	0.51	0.83

Table 2. Document similarity measurement results for COLIEE and LCRD datasets

Model	COLIEE				LCRD			
	CosSim	EuclDist	WMDist	ROC/AUC	CosSim	EuclDist	WMDist	ROC/AUC
ExGloVe	0.76	0.53	0.30	0.81	0.73	0.55	0.32	0.79
BiLSTM [28]	0.64	0.56	0.35	0.71	0.61	0.58	0.37	0.69
DSWE [11]	0.68	0.60	0.38	0.73	0.65	0.62	0.40	0.71
FastText [21]	0.57	0.58	0.36	0.68	0.51	0.60	0.38	0.57
Word2Vec [14]	0.46	0.62	0.40	0.49	0.44	0.66	0.42	0.51
GloVe [2]	0.69	0.57	0.37	0.72	0.66	0.59	0.39	0.70

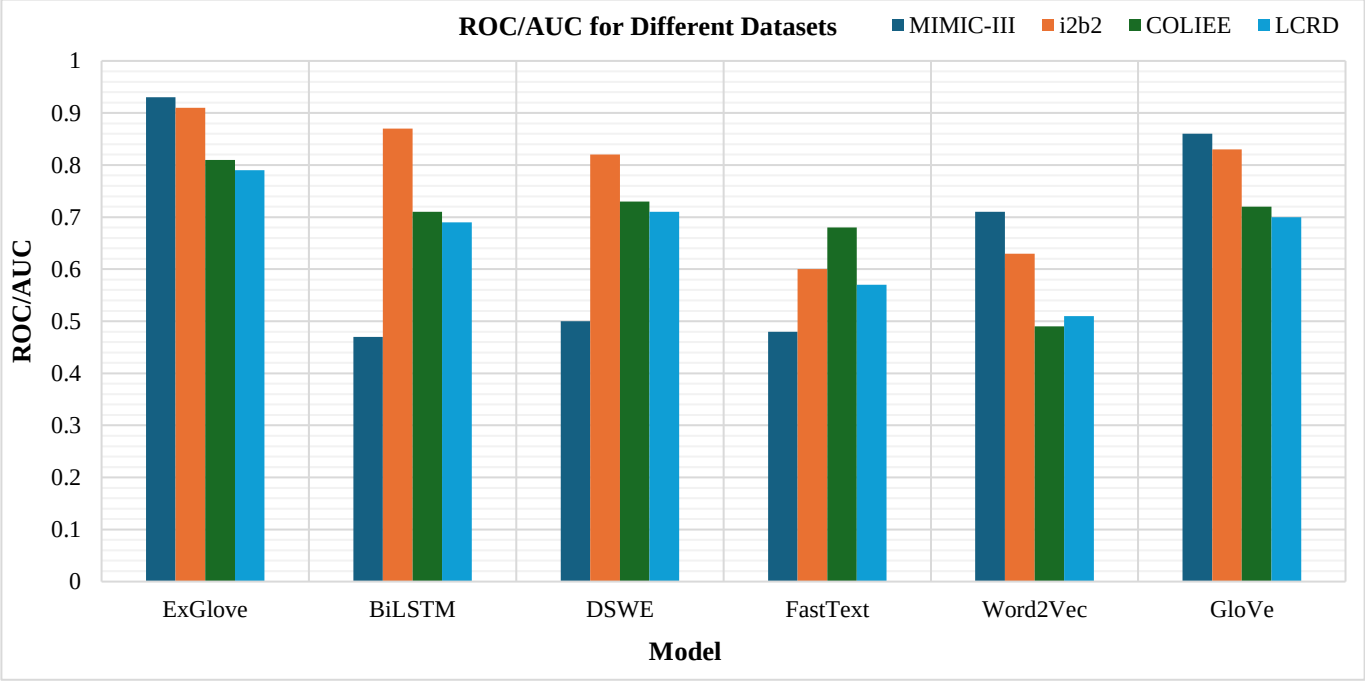

Fig. 2 Comparison of document similarity performance across multiple datasets

In document similarity measurement, ExGloVe outperformed other models with a CosSim of 0.87 and ROC/AUC of 0.93 on MIMIC-III (Table 1) and 0.84 and 0.91 on the i2b2 dataset. BiLSTM also performed strongly with CosSim of 0.83 on MIMIC-III and 0.80 on i2b2. Across all datasets, ExGloVe and BiLSTM emerged as the most effective models.

Similarly, on COLIEE and LCRD datasets (table-2), ExGloVe exhibited a CosSim of 0.76 and 0.73, respectively, with ROC/AUC of 0.81 and 0.79. While performance varied among models, ExGloVe and GloVe consistently demonstrated robust performance.

4.4.2. Clustering

In clustering experiments with ExGloVe embeddings on MIMIC-III, i2b2, COLIEE, and LCRD datasets, metrics like SiL, ARI, and NMI were assessed for performance. ExGloVe combined with DBSCAN consistently outperformed other models on MIMIC-III, achieving a Silhouette Score of 0.70, ARI of 0.80, and NMI of 0.87. Similar trends were observed on the i2b2 dataset, indicating the effectiveness of ExGloVe embeddings in capturing meaningful clusters in medical text data is shown in Table 3 and Figure 3.

On COLIEE, ExGloVe + KM yielded the highest SiL of 0.63, ARI of 0.70, and NMI of 0.78, while ExGloVe + DBS excelled on LCRD with SiL of 0.69, ARI of 0.75, and NMI of 0.81, suggesting that combining ExGloVe embeddings with specific clustering algorithms significantly improves clustering performance is shown in Table 4 and Figure 3.

4.4.3. Classification

In classification experiments across MIMIC-III, i2b2, COLIEE, and LCRD datasets, various models were evaluated, including GloVe, ExGloVe, ExGloVe + SVM, ExGloVe + LR, BiLSTM, DSWE, FastText, and Word2Vec. ExGloVe consistently outperformed other models on MIMIC-III and i2b2 datasets, with accuracies of 0.75% and 0.73%, respectively, surpassing GloVe, BiLSTM, DSWE, FastText, and Word2Vec. Combining ExGloVe with SVM or LR further improved accuracy is shown in Table 5 and Figure 4.

Table 3. Document clustering results on MIMIC-III and i2b2 datasets

Clustering Model	MIMIC-III			i2b2		
	SiL	ARI	NMI	SiL	ARI	NMI
ExGloVe	0.62	0.75	0.82	0.60	0.72	0.78
ExGloVe + KM [8]	0.68	0.78	0.85	0.66	0.75	0.80
ExGloVe + Hier [10]	0.65	0.72	0.79	0.63	0.70	0.76
ExGloVe + DBS [9]	0.70	0.80	0.87	0.68	0.78	0.83
BiLSTM [28]	0.58	0.68	0.75	0.55	0.65	0.72
DSWE [11]	0.55	0.62	0.71	0.52	0.60	0.68
FastText [21]	0.53	0.59	0.67	0.50	0.57	0.65
Word2Vec [14]	0.50	0.55	0.62	0.48	0.53	0.62
GloVe [2]	0.57	0.65	0.73	0.53	0.62	0.70

Table 4. Document clustering results on COLIEE and LCRD datasets

	COLIEE			LCRD		
Model	SiL	ARI	NMI	SiL	ARI	NMI
ExGloVe	0.55	0.62	0.71	0.60	0.70	0.78
ExGloVe + KM [8]	0.63	0.70	0.78	0.63	0.71	0.79
ExGloVe + Hier [10]	0.58	0.65	0.72	0.62	0.68	0.76
ExGloVe + DBS [9]	0.60	0.68	0.75	0.69	0.75	0.81
BiLSTM [28]	0.49	0.58	0.65	0.49	0.57	0.68
DSWE [11]	0.46	0.54	0.62	0.46	0.58	0.66
FastText [21]	0.43	0.50	0.57	0.35	0.44	0.49
Word2Vec [14]	0.39	0.44	0.51	0.37	0.45	0.53
GloVe [2]	0.47	0.55	0.63	0.51	0.59	0.66

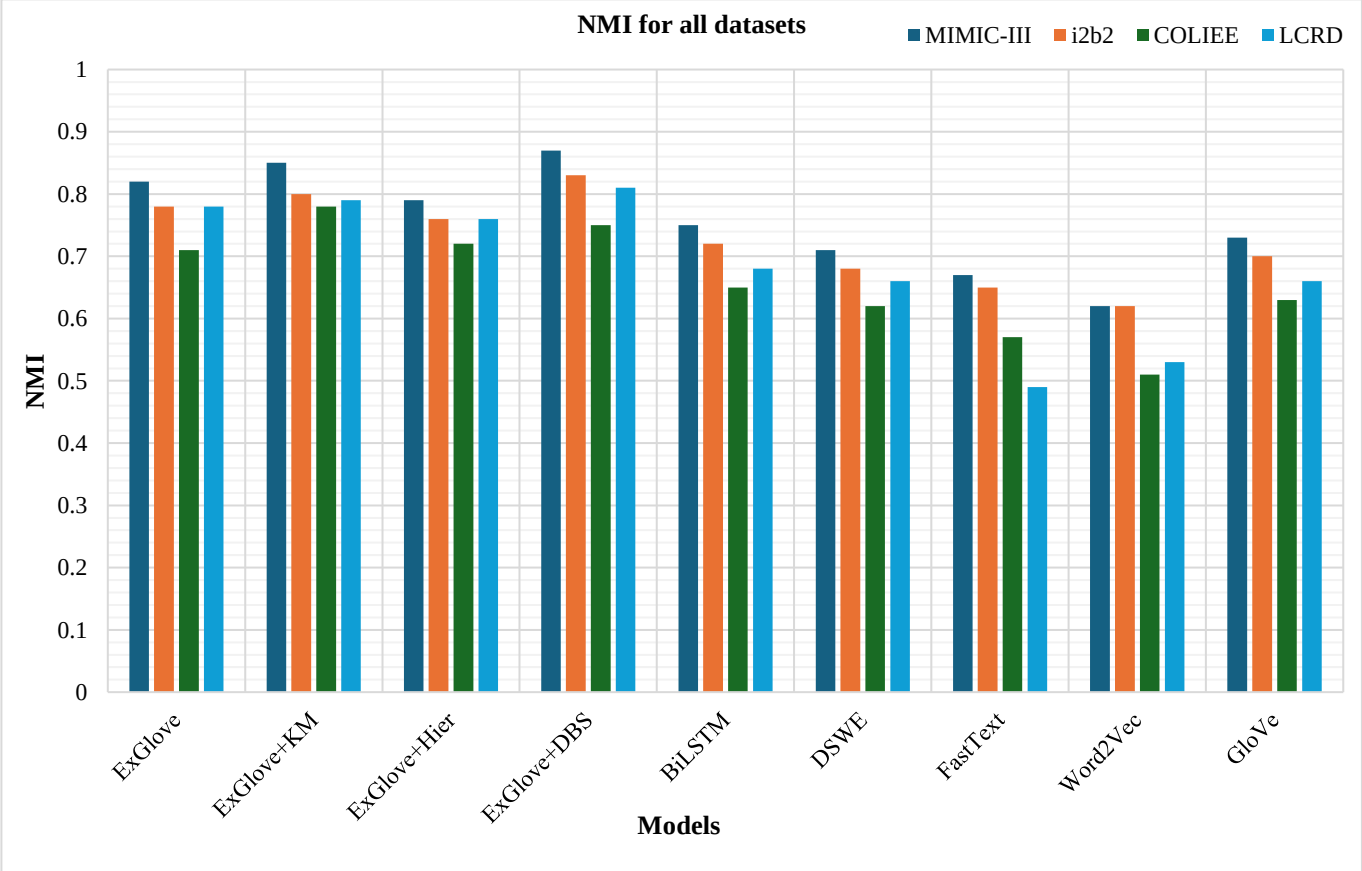

Fig. 3 Comparison of document clustering NMI performance across multiple datasets

Table 5. Classification results for MIMIC-III and i2b2 datasets

Model	MIMIC-III				i2b2			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
GloVe [2]	0.67%	0.71	0.66	0.69	0.68%	0.69	0.67	0.68
ExGloVe	0.75%	0.79	0.71	0.75	0.73%	0.72	0.75	0.74
ExGloVe + SVM [11]	0.81%	0.84	0.76	0.80	0.78%	0.80	0.71	0.76
ExGloVe+LR[12]	0.79%	0.81	0.73	0.77	0.81%	0.85	0.74	0.80
BiLSTM [28]	0.71%	0.67	0.71	0.69	0.64%	0.58	0.71	0.65
DSWE [11]	0.53%	0.58	0.49	0.54	0.59%	0.62	0.51	0.57
FastText [21]	0.63%	0.62	0.68	0.65	0.54%	0.45	0.58	0.52
Word2Vec [14]	0.59%	0.54	0.67	0.61	0.50%	0.49	0.46	0.48

Table 6. Classification results for COLIEE and LCRD datasets

Model	COLIEE				LCRD			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
GloVe [2]	0.66%	0.63	0.67	0.65	0.62%	0.61	0.61	0.62
ExGloVe	0.71%	0.75	0.67	0.71	0.70%	0.71	0.69	0.70
ExGloVe + SVM [11]	0.77%	0.80	0.72	0.76	0.74%	0.75	0.74	0.74
ExGloVe + LR [12]	0.75%	0.78	0.70	0.74	0.77%	0.74	0.75	0.75
BiLSTM [28]	0.59%	0.59	0.65	0.62	0.49%	0.47	0.48	0.48
DSWE [11]	0.55%	0.51	0.64	0.58	0.45%	0.40	0.47	0.44
FastText [21]	0.61%	0.61	0.65	0.63	0.57%	0.58	0.54	0.56
Word2Vec [14]	0.49%	0.55	0.46	0.51	0.54%	0.53	0.53	0.53

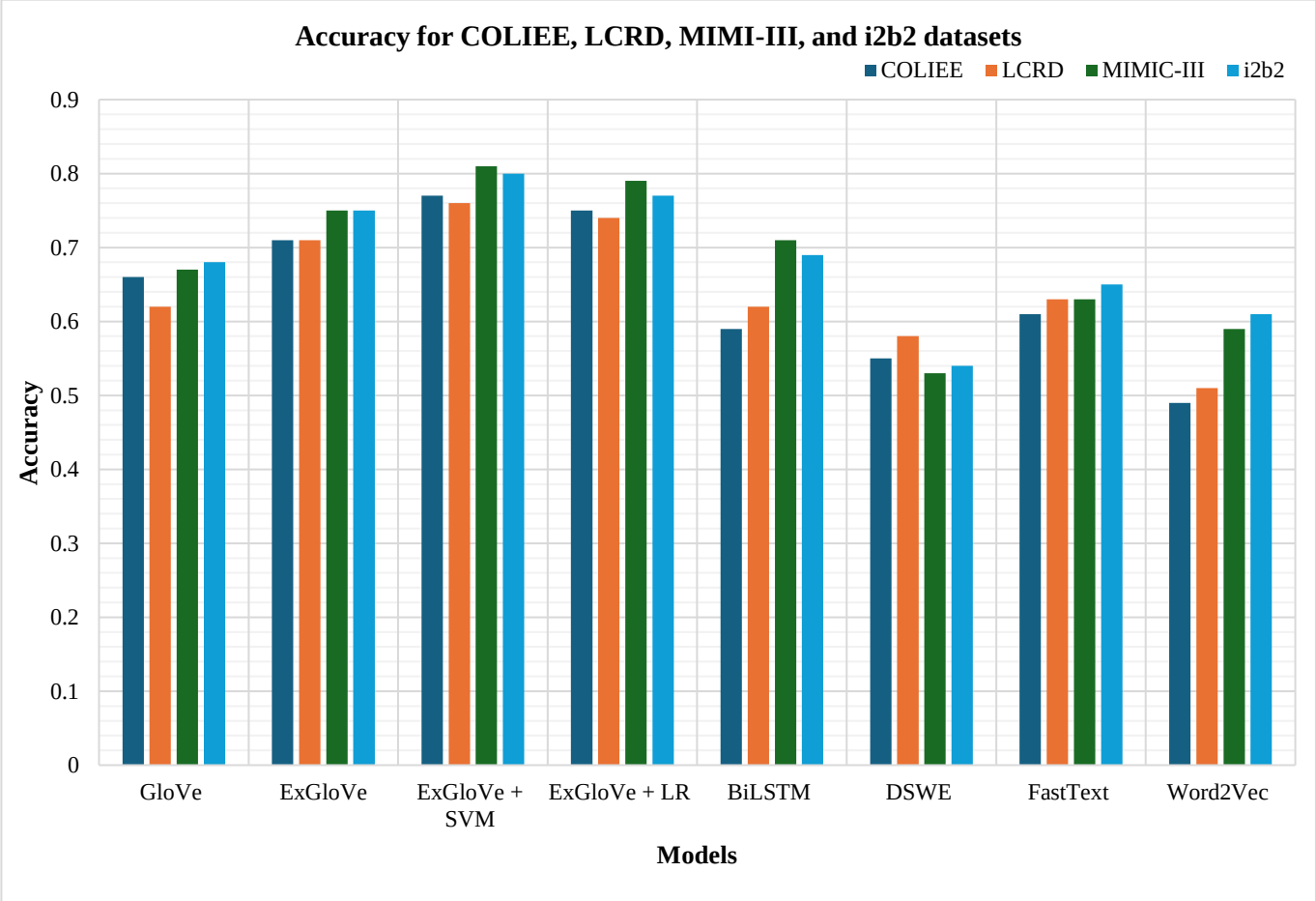

Fig. 4 Comparison of document classification accuracy performance across multiple datasets

In the COLIEE dataset, ExGloVe achieved an accuracy of 0.71%, outperforming other models, while in the LCRD dataset, ExGloVe attained an accuracy of 0.70%, also surpassing other models. Combining ExGloVe with SVM or LR enhanced accuracy further across all datasets, as shown in Table 6 and Figure 4.

5. Results Discussion

The experimental results demonstrate the significant impact of subword information and domain-specific adaptations on the performance of the ExGloVe embeddings. The inclusion of subword information in the embedding process allows the model to capture finer-grained semantic relationships within words, leading to improved performance on tasks requiring a deeper understanding of language. In our experiments, the ExGloVe embeddings consistently outperformed baseline models such as GloVe [4], BiLSTM, DSWE [26], FastText, and Word2Vec [19] across multiple tasks and datasets.

This highlights the effectiveness of incorporating subword information and domain-specific adaptations to enhance the quality of word embeddings. These findings align with our expectations outlined in the methodology section, where we hypothesized that the ExGloVe embeddings would perform better than baseline models due to their ability to capture more nuanced semantic information.

A comprehensive comparison with baseline models reveals the strengths of the ExGloVe embeddings. Compared to GloVe [4], which only considers whole words, the ExGloVe embeddings showed improved performance, especially in tasks requiring a deeper semantic understanding, such as document similarity measurement, clustering, and classification. BiLSTM, DSWE, FastText, and Word2Vec, while capable models, were outperformed by the ExGloVe embeddings, indicating that the inclusion of subword information and domain-specific adaptations can significantly enhance the performance of word embeddings in NLP tasks.

5.1. Implications

The findings of this study have several practical implications for real-world NLP applications, particularly in the domains of healthcare and legal text processing. The ExGloVe embeddings can be utilized to enhance the performance of NLP systems in these domains by providing more accurate representations of words and documents.

For healthcare applications, such as clinical document clustering and classification, the ExGloVe embeddings can improve the accuracy of diagnosis and treatment recommendations by providing a more nuanced understanding of medical terminology and concepts. In legal text processing, the ExGloVe embeddings can assist in tasks such as legal document summarization and information retrieval by capturing the complex legal language and terminology used in legal texts.

6. Conclusion

In this study, ExGloVe embeddings across NLP tasks on MIMIC-III, i2b2, COLIEE, and LCRD datasets were extensively evaluated, showcasing their superiority over baseline models like GloVe, BiLSTM, DSWE, FastText, and Word2Vec. ExGloVe excelled in capturing semantic relationships, which is evident in its high Cosine Similarity (CosSim) of 0.87 for MIMIC-III and 0.84 for i2b2. In clustering, ExGloVe with DBSCAN consistently outperformed others on MIMIC-III (SiL: 0.70, ARI: 0.80, NMI: 0.87) and i2b2 (SiL: 0.68, ARI: 0.78, NMI: 0.83). It also showed superior classification accuracy on MIMIC-III (0.75%) and i2b2 (0.73%). Despite limitations in dataset scope, our findings suggest future exploration of ExGloVe's effectiveness across diverse NLP tasks and datasets to confirm its robustness and applicability.

References

[1] Diksha Khurana et al., "Natural Language Processing: State of The Art, Current Trends and Challenges," *Multimedia Tools and Applications*, vol. 82, pp. 3713-3744, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jeffrey Pennington, Richard Socher, and Christopher Manning, "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1532-1543, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mohammad Taher Pilevar, and Nigel Collier, "Inducing Embeddings for Rare and Unseen Words by Leveraging Lexical Resources," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2, pp. 388-393, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiaotao Li et al., "Learning Embeddings for Rare Words Leveraging Internet Search Engine and Spatial Location Relationships," *Proceedings of SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pp. 278-287, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Zhongyu Zhuang et al., "Out-of-Vocabulary Word Embedding Learning Based on Reading Comprehension Mechanism," *Natural Language Processing Journal*, vol. 5, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Van-Tan Bui et al., "Combining Specialized Word Embeddings and Subword Semantic Features for Lexical Entailment Recognition," *Data & Knowledge Engineering*, vol. 141, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Debora Nozza et al., "LearningToAdapt with Word Embeddings: Domain Adaptation of Named Entity Recognition Systems," *Information Processing & Management*, vol. 58, no. 3, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[8] Asana Neishabouri, and Michel C. Desmarais, "Inferring the Number and Order of Embedded Topics Across Documents," *Procedia Computer Science*, vol. 192, pp. 1198-1207, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[9] Shapol M. Mohammed, Karwan Jacksi, and Subhi R. M. Zeebaree, "Glove Word Embedding and DBSCAN algorithms for Semantic Document Clustering," *International Conference on Advanced Science and Engineering*, Duhok, Iraq, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[10] R. Suganthi, and K. Prabha, "Fuzzy Similarity Based Hierarchical Clustering for Communities in Twitter Social Networks," *Measurement: Sensors*, vol. 32, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[11] Milad Moradi, Maedeh Dashti, and Matthias Samwald, "Summarization of Biomedical Articles using Domain-Specific Word Embeddings and Graph Ranking," *Journal of Biomedical Informatics*, vol. 107, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[12] Qing Li et al., "Logistic Regression Matching Pursuit Algorithm for Text Classification," *Knowledge-Based Systems*, vol. 277, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[13] Shuo Yang et al., "Chinese Semantic Document Classification Based on Strategies of Semantic Similarity Computation and Correlation Analysis," *Journal of Web Semantics*, vol. 63, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[14] Anil Sharma, and Suresh Kumar, "Ontology-Based Semantic Retrieval of Documents Using Word2vec Model," *Data & Knowledge Engineering*, vol. 144, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[15] Tomas Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *Proceedings of the International Conference on Learning Representations*, 2013. [CrossRef] [Google Scholar] [Publisher Link]
[16] Piotr Bojanowski et al., "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135-146. 2017. [CrossRef] [Google Scholar] [Publisher Link]
[17] Zhuang Liu et al., "FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining," *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 4513-4519, 2021. [Google Scholar] [Publisher Link]
[18] Leilei Gan et al., "SemGloVe: Semantic Co-Occurrences for GloVe From BERT," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2696-2704, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[19] Tim Schopf, Dennis N. Schneider, and Florian Matthes, "Efficient Domain Adaptation of Sentence Embeddings Using Adapters," *Arxiv*, pp. 1-8, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[20] Qian Liu et al., "Domain-Specific Meta-Embedding with Latent Semantic Structures," *Information Sciences*, vol. 555, pp. 410-423, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[21] Jaekeol Choi, and Sang-Woong Lee, "Improving FastText with Inverse Document Frequency of Subwords," *Pattern Recognition Letters*, vol. 133, 165-172, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[22] Jun Yin, and Shiliang Sun, "Incomplete Multi-View Clustering with Cosine Similarity," *Pattern Recognition*, vol. 123, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[23] Ryoma Sato, Makoto Yamada, and Hisashi Kashima, "Re-Evaluating Word Mover's Distance," *International Conference on Machine Learning*, pp. 19231-19249, 2022. [Google Scholar] [Publisher Link]
[24] Paul Rogers, Dong Wang, and Zhiyuan Lu, "Medical Information Mart for Intensive Care: A Foundation for the Fusion of Artificial Intelligence and Real-World Data," *Frontiers in Artificial Intelligence*, vol. 4, pp. 1-4, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[25] Özlem Uzuner et al., "2010 i2b2/VA Challenge on Concepts, Assertions, and Relations in Clinical Text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552-526, 2011. [CrossRef] [Google Scholar] [Publisher Link]
[26] Juliano Rabelo et al., "COLIEE 2020: Methods for Legal Document Retrieval and Entailment," *New Frontiers in Artificial Intelligence*, pp. 196-210, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[27] Haoxi Zhong et al., "JEC-QA: A Legal-Domain Question Answering Dataset," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 9701-9708, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[28] R. Prasanna Kumar et al., "Automated Sentiment Classification of Amazon Product Reviews using LSTM and Bidirectional LSTM," *International Conference on Evolutionary Algorithms and Soft Computing Techniques*, Bengaluru, India, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]