Advancements in Word Sense Disambiguation: A Poly-Encoder Bert Model Perspective

Linhan Xia *, Jiaxin Cai , Enpei Huang , Junbang Liu

Posted Date: 6 March 2024

doi: 10.20944/preprints202403.0316.v1

Keywords: NLP; Bert model; Semcor dataset; Transformer; Word semantic Disambiguation

Article

Advancements in Word Sense Disambiguation: A Poly-Encoder Bert Model Perspective

Linhan Xia [1,*,†,‡], Jiaxin Cai [1,†,‡], Enpei Huang [1,†], Junbang Liu [1,†]

Beijing Normal University-Hong Kong Baptist University United International College (UIC) 2000 Jintong Road, Tangjiawan Zhuhai, Guangdong, 519087 China

* Correspondence: temp.xialinhan@gmail.com.com; Tel.: (+86-186-8228-2213)

† Current address: Beijing Normal University-Hong Kong Baptist University United International College

‡ These authors contributed equally to this work.

Abstract: In the application domain, accurate word sense identification is crucial for improving the performance of machine translation, information retrieval and end-to-end communication tasks. However, word polysemy is a major obstacle to accurate semantic identification. Therefore, word semantic disambiguation has always been one of the key challenges in natural language processing and has attracted the attention of a large number of researchers. This research proposes an innovative disambiguation algorithm based on the large-scale Bert model and the Polly encoder framework, and introduces WordNet as a benchmark for word semantic. By exploiting the ability of the pre-trained model to extract and learn semantic information, and using a specially designed forward propagation algorithm and loss function to fine-tune the large-scale Bert model, the model has high Accuracy and robustness. In this research, several experiments were conducted on the Semcor 3.0 semantic dataset. The experimental results show that the model proposed in this research shows excellent performance on the Semcor test set, with an Accuracy of 86.1% and an F1 score of 0.847, which is a significant improvement over the traditional model.

Keywords: NLP; Bert model; Semcor dataset; transformer; word semantic disambiguation

1. Introduction

How to enable computers to accurately understand the specific semantics of a given word in natural language [1,2], so that they can perform better in machine translation [3] and end-to-end communication tasks [4], has long been a key challenge in the field of natural language processing. In recent years, the deepening of deep learning technology [5,6] in the field of natural language processing has provided researchers with new perspectives to solve this problem, but there are still significant challenges in dealing with the complexity and diversity of the words to be processed.

This research proposes an innovative word-level disambiguation [7,8] algorithm that combines a Bert-based pre-trained model [9,10] with WordNet [11–13]. While using the Poly-Encoder architecture [14]. The Bert model is widely recognised for its excellent performance in a wide range of NLP tasks. Meanwhile, WordNet as a rich semantic knowledge base provides a wide range of lexical semantic information [12] which is crucial for the disambiguation work in this research [15].

1.1. Motivation and Significance

In recent years, representative natural language processing techniques [16,17], such as ChatGPT [18], have developed rapidly due to their demonstrated superior ability to process massive amounts of data and learn complex semantic relationships. Advances in such models have led to significant achievements [19] in sentence comprehension and generation. However, the cost of technological advances such as multi modal modelling is significant. First, these advanced models require a large amount of computational resources [20], which not only imposes a high hardware overhead, but also leads to significant energy consumption [21], which is a serious environmental challenge. Second, to maintain their advanced performance, these models need to be continuously trained on huge datasets, further increasing energy consumption and carbon emissions [21].

 doi:10.20944/preprints202403.0316.v1

Research into more efficient and environmentally friendly approaches to semantic comprehension is particularly important due to the challenges faced by existing large-scale language models. In this context, it is particularly important to explore innovative approaches for semantic comprehension. The semantic annotation [22] approach, which achieves a deeper comprehension of sentences by assigning semantic tags to each word based on well-designed algorithms and external semantic repositories, provides a breakthrough solution. This approach mitigates environmental impact by reducing reliance on computing resources and effectively reducing energy consumption.

1.2. Research Objectives

The task of word disambiguation is to determine the correct meaning of polysemous words in the target text(shown in Figure 1)) [7], such as "book", "bank", etc., through a set of specially designed algorithms. The main goal of this research is to develop a semantic disambiguation algorithm based on Poly Encoder [14] architecture and Large Scale Bert model, and to use WordNet as a semantic benchmark to achieve word semantic disambiguation, which brings a new perspective to the semantic disambiguation task.

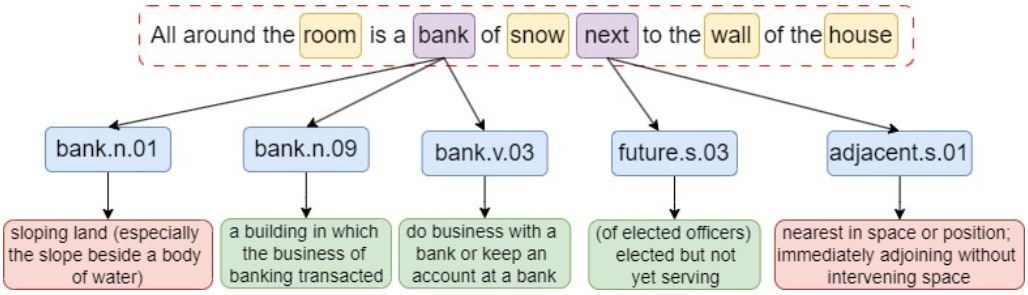

Figure 1. Example of Word semantic Disambiguation.

2. Related Work

2.1. LESK Algorithm

Michael Lesk [23] proposed the Lesk algorithm for word semantic disambiguation in 1986, which laid the foundation for this task as the earliest proposed algorithm for word semantic disambiguation. Lesk assumes that the optimal semantics of a word can determine the target semantics of the target word in context according to the cross-correlation between the text context and the target word. Lesk first proposed the idea of disambiguation based on the semantically annotated corpus represented by WordNet, i.e., by querying all possible semantics of the target word in context, and then evaluating the degree of agreement between each potential meaning defined by the corpus and the context of the target word, and selecting the semantic with the highest degree of agreement as the disambiguation result of the word.

Although Lesk's algorithm lays the foundation for later disambiguation algorithms, it still has some limitations. These include a high dependence on the lexical quality and completeness of the corpus, and poor performance when dealing with highly ambiguous contextual semantics. These shortcomings also point in the direction of optimisation for subsequent researchers researching the problem of word disambiguation.

2.2. Disambiguation base on Word2Vec

Orkphol et al. [24] proposed a approach that integrates Word2Vec and WordNet technologies for semantic selection. This approach will use the weighting mechanism of inverse document frequency [25] and the weighted moving average approach to synthesize the vectors of all words in the sentence to obtain the semantic vector of the target sentence. The equation is as shown in the Equation (1), and then calculate the target Cosine correlation value between the semantic vector of the vocabulary set and the whole sentence vector.

 doi:10.20944/preprints202403.0316.v1

$$S = \sum_{i=0}^{|W|} v(W[i]) \tag{1}$$

In their approach, Orkphol and colleagues employ structures unique to WordNet, like superlatives, hyponyms, antonyms, and semantic relations, to pinpoint the most fitting semantics through cosine correlation. This is achieved by averaging the vectors of all relevant words to represent the target semantics' mean vector.

Nonetheless, in scenarios where the contextual clues are sparse, they turn to the distribution probabilities of word semanticss to aid in accurately identifying the appropriate word semantic, as outlined in their Equation (2). This approachological pivot ensures robust word semantic disambiguation even under challenging conditions with limited context.

$$P(S_{ij}|W_i) = \frac{C(W_i, S_{ij}) + 1}{C(W_i) + C(S_i)} \tag{2}$$

The core idea of this approach is to combine the semantic representation capability of the Word2Vec word embedding model with the special structure of WordNet, and to compute the cosine value between vectors as the semantic correlation between natural languages. Unlike traditional algorithms that rely on word semantic matching, it uses vectors to mine subtle semantic differences between word semanticss, which greatly improves the Accuracy of the Lesk algorithm. However, it does not perform well when faced with semantics with complex structures.

2.3. Disambiguation base on Dependence Adaptability

Lu and Huang [26] developed a strategy for word semantic disambiguation called dependency-based adaptation for WSD. This approach automates the acquisition of necessary knowledge through deep utilization of dependency syntactic analysis, overcoming knowledge acquisition challenges encountered in the advancement of WSD technology. This research process begins by conducting a syntactic analysis of a large corpus to create a Dependency Knowledge Base (DKB). Then, sentences containing words with multiple meanings are analyzed to identify their Dependency Constraint Sets (DCS), and the keywords for each semantic of each word are identified using WordNet. To accurately identify the correct semantic, the dependency daptability of each keyword in the DCS must be computed based on the DKB. The formula for calculating the dependency daptability is detailed below:

Assuming that exists a dependency tuple $r(w_1, w_2)$to calculate the dependency Adaptability of w_1w_2 under the dependency relation r the following parameters need to be defined :

1. $a = freq^r(w_1, w_2)$:the total number of dependency tuples with a specific dependency relation r, where the dominant word is w_1 and the dependent word is w_2.
2. $b = freq^r(w_1, *)$:the total number of dependent tuples with dependency r and dominator w_1.
3. $c = freq^r(*, w_2)$:total number of dependent tuples with dependency r and dominator w_2.N^r: total number of dependency tuples with dependency r.
4. N^r: total number of dependency tuples with dependency r.

This approach employs the Point-wise Mutual Information (PMI) statistical model to assess the compatibility of lexical representatives or meanings for ambiguous words within the dependency constraint tuples. The formula for PMI is shown in Equation (3).

$$PMI^T(w_1, w_2) = \log \frac{N^r \times freq^r(w_1, w_2)}{freq^r(w_1, *) \times freq^r(*, w_2)} = \log \frac{N^r \times a}{b \times c} \tag{3}$$

This approach utilizes the pointwise interoperable information to gauge the compatibility of w_1 and w_2 within the dependency relation r. Consequently, the formula for determining the dependency daptability is outlined in Equation (4).

 doi:10.20944/preprints202403.0316.v1

$$daptability(w_1, w_2, r) = PMI^r(w_1, w_2) = \log \frac{N^r \times a}{b \times c} \tag{4}$$

However, in order to determine the dependency Adaptability of a specific lexical semantic of an ambiguous word within the constraint set, it is necessary to assess the dependency Adaptability of each representative word of every lexical semantic across all tuples of dependency constraints. This evaluation is carried out using the equation provided in Equation (5).

$$daptability(s_i, R) = \sum_{C_k \in C_{s_i}} a_k \times MAX_{w_t \in C_k, r_j \in R} daptability(w_{kt}, r_j) \tag{5}$$

This research assessedd the effectiveness of the introduced methodology using the SemEval 2007 dataset for coarse-grained English whole-word tasks. The methodology achieved an F1 score of 74.53%, outperforming unsupervised and knowledge-based approaches that do not leverage an annotated corpus.

Lu's algorithm automates knowledge gathering, reducing reliance on manually curated databases by developing a knowledge base through dependency syntax analysis. It is well-suited for analyzing extensive corpora. However, its reliance on syntactic dependency analysis can lead to varied disambiguation outcomes for different words, requiring the use of sophisticated algorithms for syntactic analysis.

2.4. Disambiguation Algorithm Based on Bi-LSTM Neural Network Model

Kågebäck and his team [27] proposed in 2016 a approach for semantic disambiguation (WSD) based on bidirectional long short-term memory networks (BiLSTM), which takes advantage of the BiLSTM model, which recognizes long term dependencies and combines them with contextual cues to accurately determine word sentiments.In addition, BiLSTM excels at utilizing word sequences for WSD because it transmits state information that is ordered and does not need to be processed through a nonlinear function, therefore is able to maintain gradient integrity more efficiently.

The structure of this model is organized around a softmax layer, a hidden layer, and the BiLSTM itself. In order to ensure the uniformity of the disambiguation approach, both the hidden layer and BiLSTM are given the ability to share parameters between different words and their meanings. At the same time, the softmax layer uses the type of word being processed to select the correct weight matrix and bias vector.

For any given word at position n, the predicted word distribution $y(n)$ is derived using the equation outlined in Equation (6).

$$y(n) = softmax(W_{wa} y_n a + b_{ay} w_n) \tag{6}$$

Formula in hidden layer is shown in Equation (7)

$$a = W_{ha}[h_{Ln-1}; h_{Rn+1}] + b_{ha} \tag{7}$$

For the input x_n at position n within the text D undergoing disambiguation, this model processes it through the Bi-LSTM as Equation (8):

$$x_n = Wxv(w_n), n \in [1, \ldots, |D|] \tag{8}$$

The model utilizes statistical knowledge and log-linear approaches to capture complex linguistic properties, thereby improving semantic Accuracy. It uses Glove embedding vocabulary to help the model train from raw text to vocabulary classification, thereby avoiding the need for external human adjustments. Such changes can improve the generalization ability of the model and be applied to languages with fewer semantic library resources. Although this model has its unique advantages, it

 doi:10.20944/preprints202403.0316.v1

requires higher computational cost compared to traditional RNNs. In addition, in order to maximize model performance, it has higher requirements on training data.

2.5. WSD Algorithm based on Gloss-Bert model

Huang and his team [28] proposed Gloss Bert, a Bert-based algorithm for word semantic disambiguation. Gloss Bert combines the reasoning ability of the Bert model with the semantic benchmark of WordNet to disambiguate the meaning of target words through a specific tuning strategy. This simplifies the complex task of semantic disambiguation to a binary classification task. The word semantic disambiguation model is fine-tuned using the cross-entropy loss function as the output layer loss function. To label the correct semantic sequence, the Gloss of all semantic of the target word is queried and combined with the specification of Bert input data in the context of the target word through data preprocessing (as shown in the following Table 1). The label for the correct semantic sequence is marked as 1, while the label for the erroneous semantic sequence is marked as 0.

Table 1. Data preprocessing approach proposed by Huang [28].

Context	Label
1.[CLS] Your research ... [SEP] systematic investigatio... [SEP]	1
2.[CLS] Your research ... [SEP] a search for knowledge [SEP]	0
3.[CLS] Your research ... [SEP] inquire into [SEP]	0
4.[CLS] Your research ... [SEP] attempt to find out in a ... [SEP]	0

The Gloss Bert model demonstrates strong generalization ability on the Semeval dataset, achieving an F1-Score of 0.874. This is due to its simplification of the word semantic disambiguation task to a binary classification task and the Bert model's capacity to reason about semantic information. The binary classification task simplifies the reasoning process and enhances the model's performance, while the Bert model provides rich semantic information. The proposed approach surpasses traditional algorithms and eliminates the necessity for a term-specific classifier.

Huang's algorithm introduces the Bert model into the WSD task for the first time, marking significant progress in word disambiguation. The introduction of the Bert model provides crucial inspiration for the development of this research.

2.6. WordNet Knowledge Graph Word Semantic Disambiguation Algorithm

Clarifying ambiguity by analyzing word positions in a large structured semantic network [29] is at the heart of the disambiguation approach to the WordNet knowledge graph. The technique exploits the dense semantic connections of WordNet to provide a strong semantic foundation for determining word meanings.

The algorithm has three major steps. First, it identifies the full range of semantics that a target word may contain based on the WordNet framework. Then, it assessess the relevance of these semantics to the context of the target word, e.g., evaluating the semantics of the context word and the semantic connections between the target word. Finally, the semantics with the strongest contextual relevance is determined as the exact meaning of the target word by maximum semantic correlation calculation. The semantic correlation calculation formula is shown in Equation (9).

$$\hat{s} = \arg\max_{s \in S} \sum_{w' \in context(w)} sim(s, s') \tag{9}$$

The algorithm for identifying appropriate word semanticss extends beyond mere contextual alignment. It incorporates the frequency of each semantic's usage, rendering the approach both pragmatic and precise. To augment precision further, a heuristic formula is employed in Equation (10).

 doi:10.20944/preprints202403.0316.v1

$$h(s) = \begin{cases} \mathrm{P(s \mid w)}, & if\ s \in S_w \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

In addition, the algorithm takes into account the influence of the entire document context C on each word semantic s when selecting the final word semantic. It assesses the weights of the word semanticss using the following Equation (11).

$$weight(s|C) = \frac{1}{|C|} \times \sum_{c \in C} correlation(s, c) \tag{11}$$

Disambiguation using WordNet Knowledge Graph can provide important correlations between word meanings and can also help to distinguish semantic differences in words. However, the efficiency of this approach is limited by the depth and breadth of WordNet's semantic coverage. This drawback is especially evident for less common semantics or newly coined terms.

3. Methodology

In this research, we develop a novel suite of algorithms for natural language disambiguation by leveraging the Bert model's fine-tuning capabilities [30] in conjunction with WordNet, serving as an external knowledge source for disambiguation purposes [15]. Our primary data repository is the SemCor 3.0 [31] semantic annotation corpus. This section will illustrate the pre-train process of Bert in 3.1, structure and feature of *WordNet* in 3.2 and proposed model for disambiguation from 3.3 to the end of this section.

3.1. Bert Model and its Features

This research's foundation and initial pre-training model are based on the Bert model(see in Figure 2), which has a distinctive training regimen that provides robust support for our research. The Bert model's training regimen is divided into pre-training [32] and fine-tuning phases, highlighting the pivotal role of Embedding in the Bert model's training process.

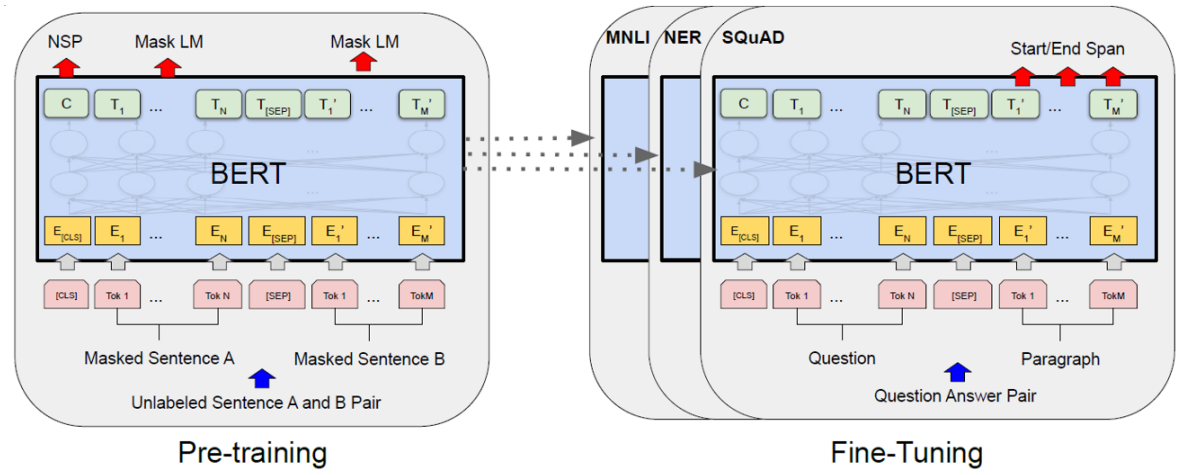

Figure 2. Structure of Bert model [10].

3.1.1. Embedding Process

Bert's (Bidirectional Encoder Representations from Transformers) architecture relies on three distinct embeddings to interpret and manage textual information: Token Embeddings [33,34], Segment Embeddings [35], and Positional Embeddings [36]. These embeddings are integrated into the input representation(shown in Figure 3).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 March 2024 doi:10.20944/preprints202403.0316.v1

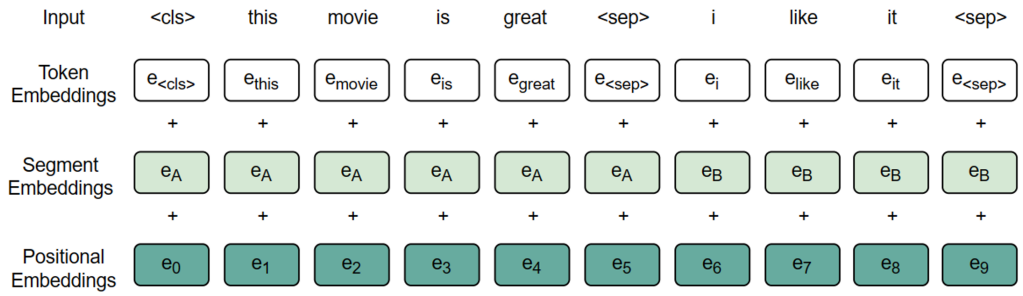

Figure 3. Embedding Process of Bert.

3.1.2. Pre-Training Process of Bert Model

Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). During Bert's pre-training stage [32], two tasks were instrumental in enhancing the model's grasp of linguistic nuances and fostering significant insights into comprehension language structures and relationships between sentences [39]. These tasks have been highly influential for this research.

Bert's Masked Language Modeling (MLM) [37] component aims to predict intentionally obscured words in a given context. This involves randomly selecting words to conceal, substituting them with a mask symbol, and tasking the model with accurately inferring these concealed elements. The corresponding equations are shown in Equations (12) and (13).

$$Bert_{input} = [CLS], X_1, X_2, ..., X_n, [SEP] \tag{12}$$

$$Input_{topredict} = [CLS], X_1, [MASK], ..., X_n, [SEP] \tag{13}$$

Among them, $X_1, X_2 \ldots X_n$ represent words in the sequence. $[CLS]$and$[SEP]$ are special token in Bert model.

The primary pre-training approach used in this research is the Next Sentence Prediction (NSP) task. The NSP task involves determining whether two sentences in the original text are related to each other through binary classification supervised learning (shown in Equation (14)). For this task, two sentences are combined using the embedding rules of the Bert model through data preprocessing. The relationship between the two sentences is then learned and judged based on the annotated data.

$$Bert_{NSP} = [CLS], Sentence_1, [SEP], Sentence_2, [SEP] \tag{14}$$

The design specific to the task significantly enhances the Bert model's capacity to comprehend the connection between sentences and reinforces the model's acquisition of the entire sequence's meaning.

The '[CLS]' token is critical for the NSP task [40]. In the Bert model, the '[CLS]' token functions as the initial token of the sequence, consolidating the meaning of all tokens in the entire sequence for the NSP task. The NSP task's binary classification algorithm predicts based on the vector in the '[CLS]' token. This research utilizes the special role of the '[CLS]' token in Bert to extract the semantic information of the entire sequence for the subsequent operation.

3.2. Exterior Knowledge Base-WordNet

WordNet is a widely used semantic knowledge base of English words in the field of natural language processing. Its special semantic structure [11–13] makes it outstanding performance of semantic annotation. WordNet is structured as a tree, where each synset node is a hyponym of its parent node. For example, 'bank.n.01' is a hyponym of 'Financial_institutions.n.01', meaning that the concept of 'bank' is part of the concept of 'financial institutions'. Each node in the WordNet model represents a synset, which contains a synset ID, a gloss, and an example. The model also includes antonyms and near-synonyms [11,41–43]. The disambiguation algorithm proposed by this research is

 doi:10.20944/preprints202403.0316.v1

based on the synset ID and gloss contributed by the WordNet. The strucure of WordNet is shown in Figure 4.

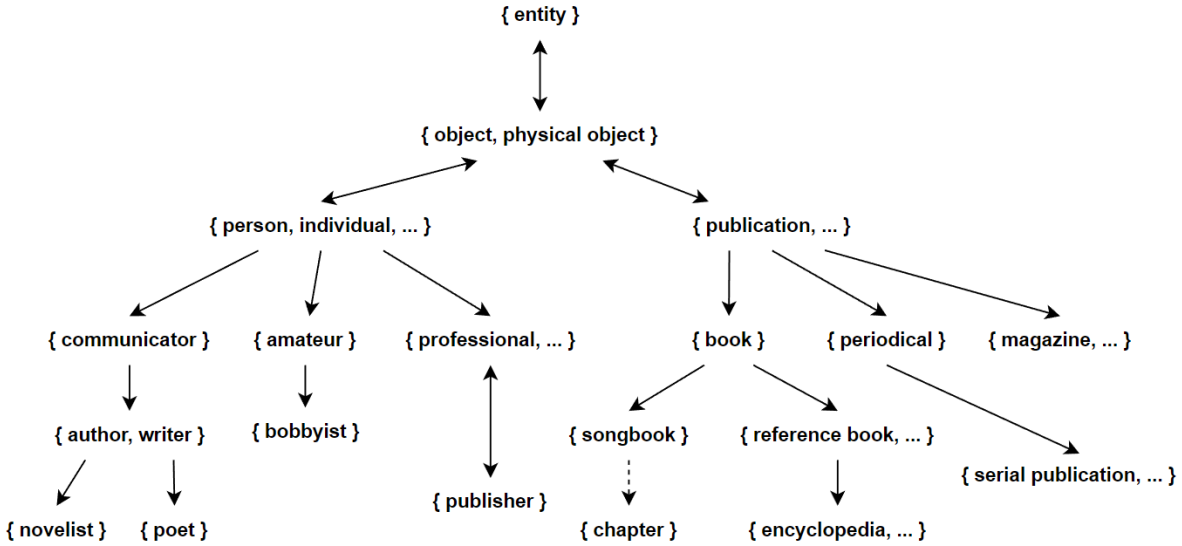

Figure 4. Structure of WordNet.

The following is the example of two Synsets.

Word:dog

Synset ID:'dog.n.01'

Gloss:*a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds*

Example:the dog barked all night

Word:car

Synset ID:'car.n.01'

Gloss:*a motor vehicle with four wheels; usually propelled by an internal combustion engine*

Example:he needs a car to get to work

Figure 5. Layout of Two synsets node includes synset ID, Gloss and example. [10].

3.3. Proposed Model

This research presents a disambiguation model based on the Poly Encoder framework, as illustrated in Figure 6. The model's objective is to select the appropriate semantics of the target word by learning the relationship between the target word's context and the semantic annotations of the correct semantics of the target word in its context. The model achieves the selection of the correct semantics of the target word through specialized algorithmic strategies. The model combines the characteristics of Poly Encoder and Bert to extract semantic details, particularly fine-grained semantics.

The attention dot product algorithm [44] is the core mechanism of the model, significantly improving its disambiguation ability. The algorithm assists the model in acquiring significant semantic information from the data by continuously adjusting the weights of the attention matrix during the iterative training of the neural network.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 March 2024 doi:10.20944/preprints202403.0316.v1

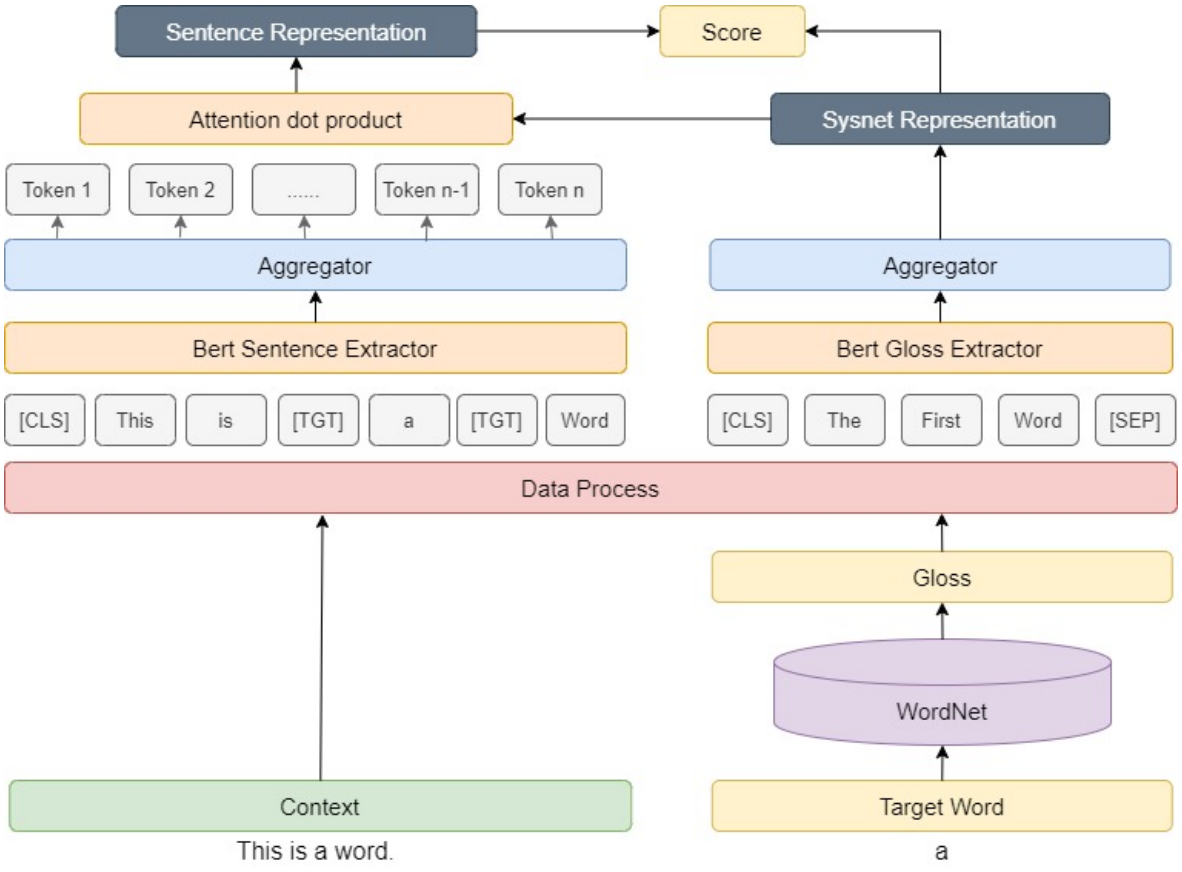

Figure 6. Structure of proposed model.

3.4. Data Preprocessing

The proposed model is illustrated in Figure 6. During the data processing stage, the model utilises a data loader in the data preprocessing step to convert the data into a 'context-correct semantics' one-to-one format. Then, processed data is then fed into the neural network model for the subsequent step. The processing results in the correct semantics of a target word being its true label in a specific contextual target sentence. However, in the context of other target sentences, the correct semantic of the target word is the context of the specific target sentence. This data processing approach simplifies the task and implements comparative learning in batches, which helps the model better distinguish between correct and incorrect semantics. The result of data preprocessing is outlined in Table 2.

Table 2. Results of data preprocessing ([TGT] is employed to help algorithm identifies target word).

Context	Gloss
In some instances a seventh question can be [TGT] added [TGT] :	state or say further
The latter [TGT] is [TGT] what concerns us all .	be identical to
Each family line can be [TGT] sonsidered [TGT] a substructure .	look at attentively
This tax was [TGT] discontinued [TGT] in 1936 .	put an end to an activity

3.5. Extraction of Sentimental Information

This research proposes a model that uses two unfine-tuned large-scale Bert models to independently extract semantic information for correct semantic from target sentences and target words. The model also adopts a differentiated extraction strategy to achieve deeper mining of this semantic information.

 doi:10.20944/preprints202403.0316.v1

3.5.1. Extraction of Gloss Sentimental Information

During the pre-training phase of the Bert model, particularly when performing the Next Sentence Prediction (NSP) [39,40] task, the [CLS] marker plays a crucial role in extracting global semantic information about the sequence. The vector representation of this token can reflect the depth of comprehension of the entire sentence by the Bert model. This research utilises the core property by utilising the [CLS] tokens from the Large-scale Bert model output to represent the accurate semantic information of the target word. Consequently, the model's output is a matrix of dimensions $[batch\ size, 1024]$, where $batch\ size$ represents the number of sentences fed into the model in a batch.

3.5.2. Extraction of Target Sentence Sentimental Information

To extract the context semantics of the target word, we use a similar approach to extract the semantic information of the target word. Compared to the target word meaning extraction, the context semantic extraction not only focuses on the [CLS] token embedded in the whole sequence semantic vector, but also extracts the semantic vectors of multiple tokens including the [CLS] token from the output sequence of the Bert model to form a new three-dimensional vector. To implement this strategy, we introduce a hyperparameter called $poly_m$. This hyperparameter determines the number of tokens extracted in the process of extracting semantic information from the target sentence. Through this approach, the model can learn the information of the whole semantic sequence more comprehensively, especially improving the ability of the model to extract fine-grained semantics.

By extracting the correct semantic information of the target word and the semantic information of the target data, we get two matrices(shown in Equations (15) and (16).

$$Context_{output} : [Batch\ size, poly_m, 1024] \tag{15}$$

$$Gloss_{output} : [Batch\ size, 1024] \tag{16}$$

Where 1024 is the dimension of Large-scale Bert's output vector.

3.6. Attention Dot Product

We construct two semantic matrices, $Context_{output}$ and $Gloss_{output}$, by extracting the correct semantic of the target word in the context and the semantic of the context in which the target word is located. To allow the model to pay attention to the important semantic information in the two semantic matrices, we introduce the attention dot product mechanism. The attention dot product mechanism is used to extract the important semantic information in $Context_{output}$ and $Gloss_{output}$ respectively, and the new semantic matrices $Context_{new}$ and $Gloss_{new}$ are obtained, whose dimensions are $[Batch\ size, poly_m, 1024]$,$[Batch\ size, 1024]$ respectively.

For ease of later computation, we construct a new dimension in $Gloss_{new}$, resulting in a new matrix dimension as Equation (17).

$$Gloss_{new} : [Batch\ size, poly_m, 1024] \tag{17}$$

To implement the attention dot product, we construct two trainable query matrices of dimension $[batchsize, poly_m, 1024]$. The initial values of these matrices are randomly generated, and their trainable specificity allows the model to dynamically adjust the attention weights during the learning process. This allows the model to continuously optimise its comprehension of semantics by learning important semantic information during the training process.

The two constructed query matrices are taken as $Query$, the semantic information vector matrix is taken as $Value$ and Key, and the extracted semantic information matrix is constructed by the attention dot product operation. The attention mechanism [45] is as follows Figure 7.

 doi:10.20944/preprints202403.0316.v1

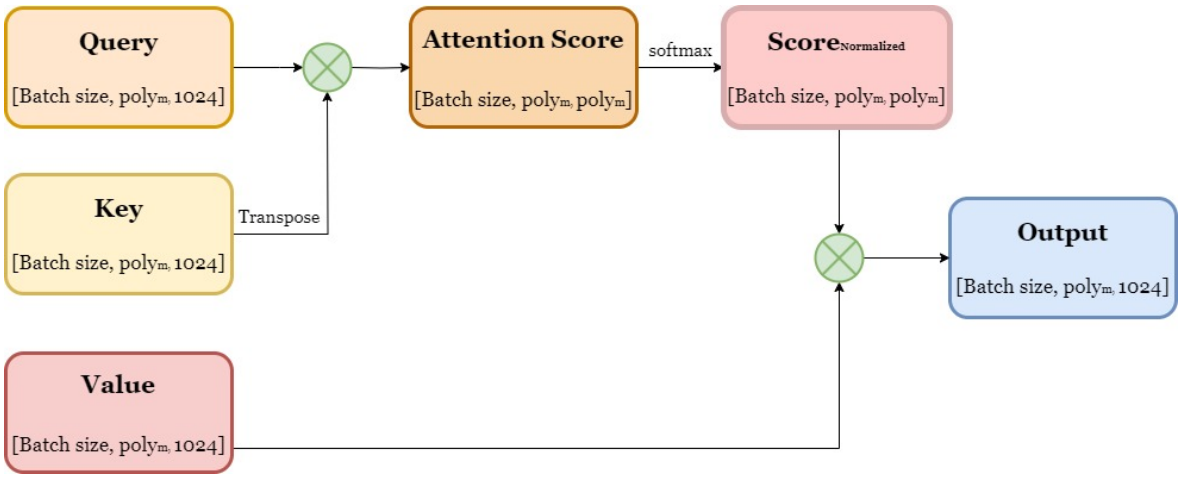

Figure 7. Process of attention product proposed by this research.

By attention dot product, we get two new matrices, $Context_{Attention}$ and $Gloss_{Attention}$, whose dimension are $[Batch\ size, poly_m, 1024]$.

3.7. Proposed Loss Function and Output Layer

After completing the attention dot product, this research proposes a dot product algorithm based on the Poly-Encoder architecture, which computes the new two matrices Context and Gloss by dot product. The core idea of the algorithm is to convert two matrices of dimension $[Batch\ size, poly_m, 1024]$ into one matrix of dimension $[Batch\ size, poly_m, Batch\ size]$ by dot product. And the matrix of dimension $[Batch\ size, poly_m, Batch\ size]$ named *semantic correlation matrix*.

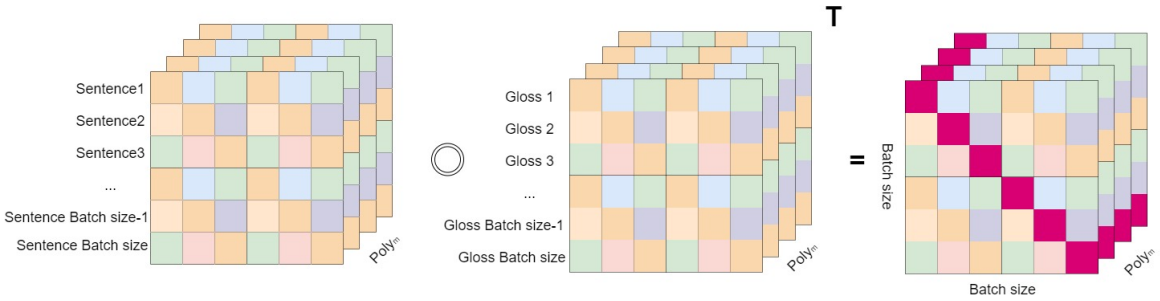

Figure 8. Matrix dot product algorithm flow chart.

The dot product operation was applied to the resultant matrix, which was then subjected to a summation operation through the third dimension which dimension is $poly_m$ to reduce its dimensionality. The resulting matrix has a size of $[Batch\ size, Batch\ size]$. The process is illustrated in Equation (18).

$$Matrix_{Result} = \sum_{k=1}^{poly_m} Dot\ Product_k \tag{18}$$

Where $Dot\ Product_k$ is every sub matrix of *semantics correlation matrix* which dimension is $[Batch\ size, Batch\ size]$.

Upon careful analysis of this matrix, a key observation is revealed: only the elements at the diagonal positions of the matrix display the interactions between the context of each target word within the sentence and its corresponding semantic annotation. The matrix's diagonal elements represent the degree of association between the target word context and the matching semantic annotation, reflecting the corresponding level of correlation. The other elements of the matrix reveal

 doi:10.20944/preprints202403.0316.v1

the association between each sentence context and the mismatched semantic annotations, indicating that these elements correspond to incorrect semantic interpretations.

Based on this analysis, the matrix is used to guide the model's training strategy. The strategy is optimized by enhancing the correlation values of diagonal elements, which represent correct semantic matches, and decreasing the values of non-diagonal elements, which represent incorrect semantic matches. This results in a more accurate distinction between correct and incorrect semantic annotations.

Based on the characteristics of the semantic correlation matrix, we propose a loss function that is specifically designed for this purpose. Firstly, we use the Softmax function to map each value in the semantic information data to the range of [0,1]. The closer the value is to 1, the more semantically relevant it is, and the closer it is to 0, the less semantically relevant it is. The diagonal values of the matrix indicate the correlation between the correct semantic and the context of the target word. The remaining values indicate the correlation between the incorrect semantics and the context of the target word. To extract the semantic correlation on the diagonal, we use an extraction matrix with 1 only on the diagonal and 0 elsewhere. The program extracts values and calculates their average in a batch. The neural network iteration optimizes disambiguation results by minimizing the inverse of the mean correlation between the correct semantic of target words and their context semantics. Equation (19),−21 provides the expression.

$$Mask = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{19}$$

$$Softmax(x_i) = \frac{e^{e_i^x}}{\sum_{j=1}^{n} e^{x_j}} \tag{20}$$

$$Loss = -\frac{Mask \cdot \log(Softmax(x_i))}{Batch\ size} \tag{21}$$

Using a specially designed loss function like Equation (21), the model can penalise incorrect meanings' correlation and achieve disambiguation.

4. Experiments and Performance Evaluation

In this section, three experiments are conducted to assess the performance and optimize the hyper-parameters. In 4.1, experiment of optimization on model illustrate that modifying *batch size* and $poly_m$ will improve the performance of disambiguation task. In 4.2, experiment of sensitivity analysis is conducted has proven the robust of model. In 4.3, we train a optimized model which hyper-parameters are decided by experiment 1 in 4.1 and optimized noise level is employed. We assess the performance of optimized model and compared with relative algorithm from relative work.

The following is the information of our device used in both two experiments.

Device of Experiments

Device information is below:

GPU	RTX 2080Ti from NVIDIA
CPU	Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz
System	Windows 10
GPU memory	24GB
Python version	3.10.1
Transformers version	4.38.0

4.1. Performance Evaluation and Optimization

In this experiment, Semcor 3.0 dataset is employed in the training process, and Large-scale Bert model is taken as basic model to be fine-tuned. In this experiment we aim to modify hyper-parameters

 doi:10.20944/preprints202403.0316.v1

such as *Batch size* and $poly_m$ to optimize the model's performance. Two metrics are applied in this experiment include F-1 Score and Accuracy [46].

To achieve this targets we construct a serious of hyper-parameters to train models and assess their performance, the hyper-parameters are from 1 to 20. And we conduct combination of them to finish all the combination's experiments. Because of limitation of computing resource we use randomly selected sub dataset to assess performance.

We conduct this experiment and record the result of Accuracy and F1 Score, the result are shown in Figures 9 and 10.

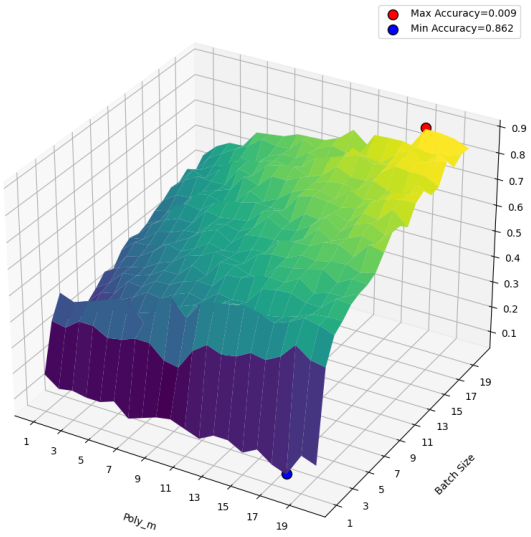

Figure 9. This chart illustrate the distribution of Accuracy of models trained by different hyper-parameters.

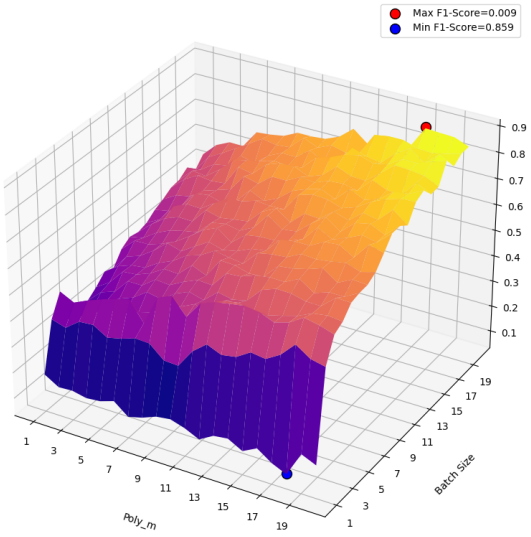

Figure 10. This chart illustrate the distribution of F1 Score of models trained by different hyper-parameters.

According to the result of this experiment, we can conclude that by increasing $poly_m$ and *Batch size*, the performance of the model improves dramatically. What's more, it is a significant phenomenon that when *Batch size* are nearly to 1, the Accuracy and F1 score are nearly to zero, which means that under this situation, the model has no ability to disambiguate.

 doi:10.20944/preprints202403.0316.v1

The reason lies in the loss function and output layer of the proposed model (mentioned in subsection 3.7). Because this model uses comparative learning as a training strategy, where we maximise the sentimental correlation between the context and the true semantic of the target word and minimise the correlation between the context and its false sentimental label. If *batchsize* is equal to 1, it means that only one true label can be learned and no false label will be considered, so the model will lose disambiguation ability.

As for $poly_m$ mentioned in subsection 3.5, it represents the number of tokens taken in the sentiment extraction process. The more tokens are selected, the more complete sentiment information is extracted. By increasing $poly_m$ the performance of the model will be improved.

4.2. Sensitivity Analysis

This experiment is conducted to assess the robustness of the proposed model in this research. We add noise to the training process by modifying the label of the training set. We set different groups of noise level to assess the robust of model [47], we apply 1%, 3%, 5% and 10% modification in data set as experimental group and original model as control group where there is no noise in training data.

After modifying the training set and training process, we get 4 models. Evaluating their performance we get the result shown in the Figure 11.

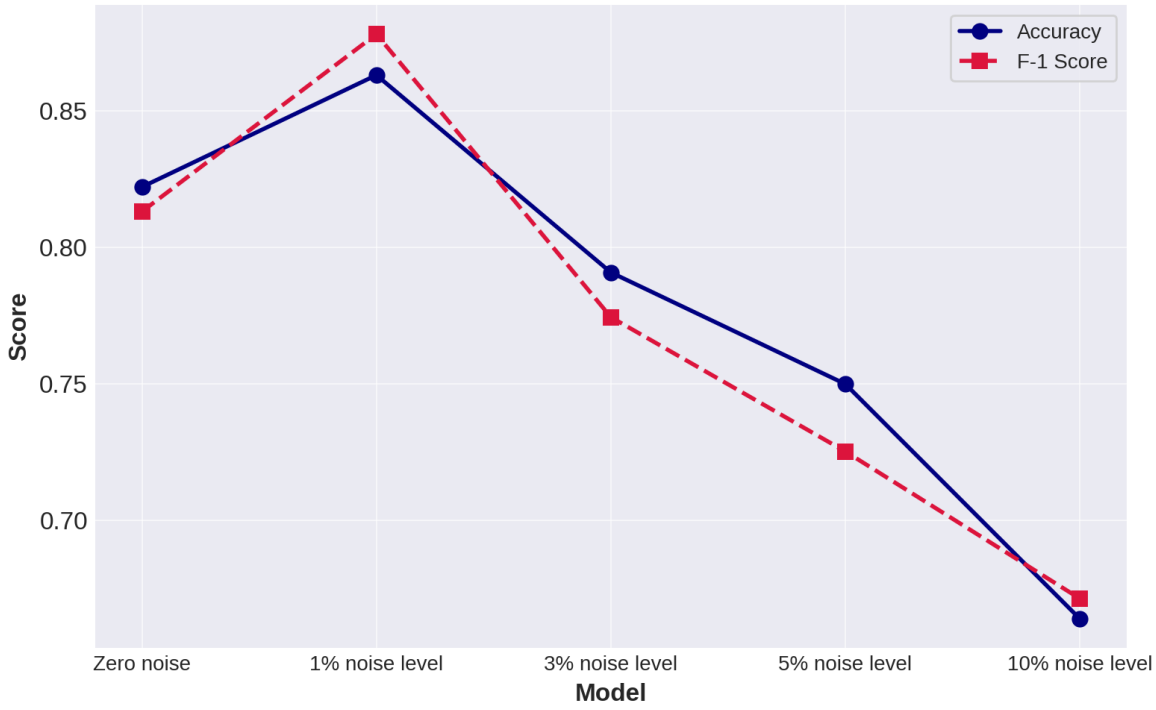

Figure 11. This figure illustrates different noise levels in the training set and their effect on Accuracy and F-1 Score.

According to the result, a conclusion can be drawn that with a limited noise level applied in the dataset, the performance of the model will be improved, but too high noise level will cause the decrease of the performance [48]. What's more, the sensitivity analysis experiment illustrates that the proposed model has strong robustness.

4.3. Performance Evaluation of Optimized model and Comparison

In this experiment, we have developed an optimized model by adjusting hyperparameters based on the outcomes of our experimental findings. The hyperparameters were set according to the specifications detailed in the following Table 3.

 doi:10.20944/preprints202403.0316.v1

Table 3. Optimized parameters conducted in this expirement.

Parameters	Value
$poly_m$	19
Batch size	19
Noise level	0.01

In this experiment, we utilize Accuracy and F1-Score as metrics to assess performance. We examine the algorithms discussed in the previous section and designate them as the control group. These algorithms include the Lesk [23] algorithm, Word2Vec approach [24], Dependence adaptability approach [26], Bi-LSTM approach [27], Gloss-Bert model [28], and the WordNet knowledge graph word semantics disambiguation algorithm [29]. Our proposed model is considered as the experimental group for comparison.

The result of experiment is shown in the following Figure 12.

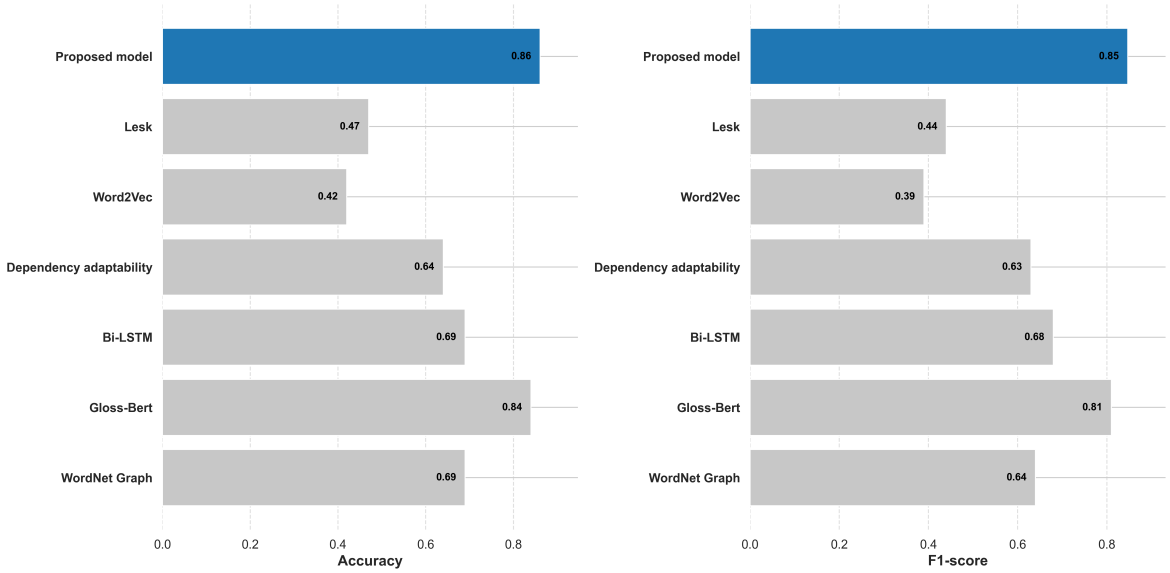

Figure 12. Result of comparison between experimental group and control group.

Based on the results of this experiment, it is evident that the proposed model outperforms the control group. And the best model's Accuracy is 86.1% while its F1-Score is 0.847. That means in most situation, our model can locate true semantic of target sentence.

5. Conclusions

This research proposes a novel word semantics disambiguation model based on Poly-Encoder Bert. We combine the outstanding inference capability of large-scale Bert model and the framework of Poly-Encoder, as a result we achieve extremely high performance, the Accuracy is 86.1% and F1-Score is 0.847. Compared with previous, our model achieves relatively better performance. This research provides a novel approach to solve the word semantics disambiguation task. In the future, we will continue to optimise the generalisation ability. What's more, in the next step of our research, we will apply this model to machine-to-machine communication [49–51] to improve the quality of sentence comprehension between different terminals.

Appendix A

This section show the demo result of proposed model's generalization.

1. **Context sentence:** The principle is commendable but we suspect that in the practice somebody is going to get gulled .

 doi:10.20944/preprints202403.0316.v1

Target word: gulled
Data ID in Semcor: d291.s073.t002
WSD result: gull%2:32:00::
Gloss of true semantic: fool or hoax

2. **Context sentence:** Since 1953 California has led the nation in enacting guarantees that public business shall be publicly conducted , but not until this year did the lawmakers in Sacramento plug the remaining loopholes in the Brown Act .
Target word: plug
Data ID in Semcor: d291.s100.t003
WSD result: plug%2:35:01::
Gloss of true semantic: fill or close tightly with or as if with a plug

3. **Context sentence:** In the tower , five men and women pull rhythmically on ropes attached to the same five bells that first sounded here in 1614 .
Target word: woman
Data ID in Semcor: l000.s005.t002
WSD result: woman%1:18:00::
Gloss of true semantic: an adult female person (as opposed to a man)

4. **Context sentence:** They belong to a group of 15 ringers – including two octogenarians and four youngsters in training – who drive every Sunday from church to church in a sometimes-exhausting effort to keep the bells sounding in the many belfries of East Anglia .
Target word: drive
Data ID in Semcor: l000.s010.t008
WSD result: drive%2:38:00::
Gloss of true semantic: move by being propelled by a force

5. **Context sentence:** In a well-known detective-story involving church bells , English novelist Dorothy L. Sayers described ringing as a " passion that finds its satisfaction in mathematical completeness and mechanical perfection . "
Target word: church
Data ID in Semcor: d000.s030.t003
WSD result: church%1:06:00::
Gloss of true semantic: a place for public (especially Christian) worship

6. **Context sentence:** It is a passion that usually stays in the tower, however .
Target word: tower
Data ID in Semcor: d000.s033.t003
WSD result: tower%1:06:00::
Gloss of true semantic: a structure taller than its diameter; can stand alone or be attached to a larger building

7. **Context sentence:** Since 1953 California has led the nation in enacting guarantees that public business shall be publicly conducted , but not until this year did the lawmakers in Sacramento plug the remaining loopholes in the Brown Act .
Target word: plug
Data ID in Semcor: d291.s100.t003
WSD result: plug%2:35:01::
Gloss of true semantic: fill or close tightly with or as if with a plug

 doi:10.20944/preprints202403.0316.v1

8. **Context sentence:** Scientists say the discovery of these genes in recent months is painting a new and startling picture of how cancer develops .
   Target word: cancer
   Data ID in Semcor: d001.s001.t009
   WSD result: cancer%1:26:00::
   Gloss of true semantic: any malignant growth or tumor caused by abnormal and uncontrolled cell division; it may spread to other parts of the body through the lymphatic system or the blood stream', 'type genus of the family Cancridae', 'a small zodiacal constellation in the northern hemisphere; between Leo and Gemini

9. **Context sentence:** The newly identified genes differ from a family of genes discovered in the early 1980s called oncogenes .
   Target word: genes
   Data ID in Semcor: d001.s011.t002
   WSD result: gene%1:08:00::
   Gloss of true semantic: (genetics) a segment of DNA that is involved in producing a polypeptide chain; it can include regions preceding and following the coding DNA as well as introns between the exons; it is considered a unit of heredity

10. **Context sentence:** Because of the isolation of the retinoblastoma tumor-suppressor gene , it became possible last January to find out what threat the Quinlan baby faced .
    Target word: became
    Data ID in Semcor: d001.s021.t003
    WSD result: become%2:30:00::
    Gloss of true semantic: enter or assume a certain state or condition

11. **Context sentence:** " All this may not be obvious to the public , which is concerned about advances in treatment , but I am convinced this basic research will begin showing results there soon . "
    Target word: obvious
    Data ID in Semcor: d001.s027.t000
    WSD result: obvious%3:00:00::
    Gloss of true semantic: easily perceived by the semanticss or grasped by the mind

12. **Context sentence:** The story of tumor-suppressor genes goes back to the 1970s , when a pediatrician named Alfred G. Knudson Jr. proposed that retinoblastoma stemmed from two separate genetic defects .
    Target word: genetic
    Data ID in Semcor: d001.s037.t010
    WSD result: genetic%3:01:02::
    Gloss of true semantic: of or relating to the science of genetics

13. **Context sentence:** The result is a generation of young people whose ignorance and intellectual incompetence is matched only by their good opinion of themselves .
    Target word: ignorance
    Data ID in Semcor: d002.s010.t004
    WSD result: ignorance%1:09:00::
    Gloss of true semantic: the lack of knowledge or education

14. **Context sentence:** Already two major pharmaceutical companies , the Squibb unit of Bristol-Myers Squibb Co. and Hoffmann-La Roche Inc. , are collaborating with gene hunters to turn the

 doi:10.20944/preprints202403.0316.v1

anticipated cascade of discoveries into predictive tests and , maybe , new therapies .
Target word: predictive
Data ID in Semcor: d001.s090.t011
WSD result: predictive%5:00:00:prophetic:00
Gloss of true semantic: of or relating to prediction; having value for making predictions

References

1. Tanenhaus M K, Trueswell J C. Sentence comprehension[J]. 1995.
2. Bedny M, HulBert J C, Thompson-Schill S L. comprehension words in context: The role of Broca's area in word comprehension[J]. Brain research, 2007, 1146: 101-114.
3. Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case research. In Proceedings of the 40th International Conference on Machine Learning (ICML'23), Vol. 202. JMLR.org, Article 1722, 41092–41110.
4. Sagduyu Y E, Ulukus S, Yener A. Task-oriented communications for nextG: End-to-end deep learning and AI security aspects[J]. IEEE Wireless Communications, 2023, 30(3): 52-60.
5. Alzubaidi L, Bai J, Al-Sabaawi A, et al. A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications[J]. Journal of Big Data, 2023, 10(1): 46.
6. Yi K, Zhang Q, Cao L, et al. A survey on deep learning based time series analysis with frequency transformation[J]. CoRR, abs/2302.02173, 2023.
7. Navigli R. Word sense disambiguation: A survey[J]. ACM computing surveys (CSUR), 2009, 41(2): 1-69.
8. Loureiro, D. Rezaee, K Pilevar, M. Camacho-Collados, J. (2020). Language Models and Word Sense Disambiguation: An Overview and Analysis.
9. Han X, Zhang Z, Ding N, et al. Pre-trained models: Past, present and future[J]. AI Open, 2021, 2: 225-250.
10. Kenton J D M W C, Toutanova L K. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of naacL-HLT. 2019, 1: 2.
11. Fellbaum C. WordNet[M]//Theory and applications of ontology: Computer applications. Dordrecht: Springer Netherlands, 2010: 231-243.
12. Miller G A. WordNet: A lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.
13. Morato J, Marzal M A, Lloréns J, et al. Wordnet applications[C]//Proceedings of GWC. 2004: 20-23.
14. Humeau, S., Shuster, K., Lachaux, M., Weston, J. (2019). Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. International Conference on Learning Representations.
15. Samhith K, Tilak S A, Panda G. Word sense disambiguation using WordNet lexical categories[C]//2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, 2016: 1664-1666.
16. Sun S, Luo C, Chen J. A review of natural language processing techniques for opinion mining systems[J]. Information fusion, 2017, 36: 10-25.
17. Bharadiya J. A Comprehensive Survey of Deep Learning Techniques Natural Language Processing[J]. European Journal of Technology, 2023, 7(1): 58-66.
18. Wu T, He S, Liu J, et al. A brief overview of ChatGPT: The history, status quo and potential future development[J]. IEEE/CAA Journal of Automatica Sinica, 2023, 10(5): 1122-1136.
19. Zhou J, Ke P, Qiu X, et al. ChatGPT: Potential, prospects, and limitations[J]. Frontiers of Information Technology Electronic Engineering, 2023: 1-6.
20. Hadi M U, Qureshi R, Shah A, et al. A survey on large language models: Applications, challenges, limitations, and practical usage[J]. Authorea Preprints, 2023.
21. Samsi S, Zhao D, McDonald J, et al. From words to watts: Benchmarking the energy costs of large language model inference[C]//2023 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2023: 1-9.
22. Savelka J, Ashley K D. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts[J]. Frontiers in Artificial Intelligence, 2023, 6.
23. Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone[C]//Proceedings of the 5th annual international conference on Systems documentation. 1986: 24-26.

 doi:10.20944/preprints202403.0316.v1

24. Orkphol K, Yang W. Word sense disambiguation using cosine correlation collaborates with Word2vec and WordNet[J]. Future Internet, 2019, 11(5): 114.
25. RoBertson, S. (2004). "Comprehension inverse document frequency: On theoretical arguments for IDF." Journal of Documentation, Vol. 60 No. 5, pp. 503-520.
26. Lu Wenpeng, Huang Heyan. Knowledge Automatic Acquisition Approach for Word Sense Disambiguation Based on Dependency Adaptability [J]. Journal of Software, 2013, 24(10): 2300-2311.
27. Mikael Kågebäck and Hans Salomonsson. 2016. Word Sense Disambiguation using a Bidirectional LSTM. In Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V), pages 51–56, Osaka, Japan. The COLING 2016 Organizing Committee.
28. Huang, L., Sun, C., Qiu, X., Huang, X. (2019). GlossBert: Bert for Word Sense Disambiguation with Gloss Knowledge. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3507–3512). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1355
29. Sakae Mizuki and Naoaki Okazaki. 2023. Semantic Specialization for Knowledge-based Word Sense Disambiguation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3457–3470, Dubrovnik, Croatia. Association for Computational Linguistics.
30. Bilal M, Almazroi A A. Effectiveness of fine-tuned Bert model in classification of helpful and unhelpful online customer reviews[J]. Electronic Commerce Research, 2023, 23(4): 2737-2757.
31. De Luca E W. A corpus for evaluating semantic multilingual web retrieval systems: The sense folder corpus[J]. argument, 2010, 48: 5.
32. Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., … Sun, L. (2023). A Comprehensive Survey on Pretrained Foundation Models: A History from Bert to ChatGPT.
33. Kanade A, Maniatis P, Balakrishnan G, et al. Learning and evaluating contextual embedding of source code[C]//International conference on machine learning. PMLR, 2020: 5110-5121.
34. Mehta, S., Koncel-Kedziorski, R., Rastegari, M., Hajishirzi, H. (2020). Define: Deep factorized input token embeddings for neural sequence modeling. In International Conference on Learning Representations.2019.
35. van der Goot R, Müller-Eberstein M, Plank B. Frustratingly Easy Performance Improvements for Low-resource Setups: A Tale on Bert and Segment Embeddings[C]//Proceedings of the Thirteenth Language Resources and Evaluation Conference. 2022: 1418-1427.
36. Yu-An Wang and Yun-Nung Chen. 2020. What Do Position Embeddings Learn? An Empirical research of Pre-Trained Language Model Positional Encoding. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6840–6849, Online. Association for Computational Linguistics.
37. Koren Lazar, Benny Saret, Asaf Yehudai, Wayne Horowitz, Nathan Wasserman, and Gabriel Stanovsky. 2021. Filling the Gaps in Ancient Akkadian Texts: A Masked Language Modelling Approach. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4682–4691, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
38. Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. NSP-Bert: A Prompt-based Few-Shot Learner through an Original Pre-training Task —— Next Sentence Prediction. In Proceedings of the 29th International Conference on Computational Linguistics, pages 3233–3250, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
39. Lee K, Choi G, Choi C. Use all tokens method to improve semantic relationship learning[J]. Expert Systems with Applications, 2023, 233: 120911.
40. Wu, L. [CLS] Token is All You Need for Zero-Shot Semantic Segmentation.
41. Zhu X, Yang X, Huang Y, et al. Measuring correlation and relatedness using multiple semantic relations in WordNet[J]. Knowledge and Information Systems, 2020, 62: 1539-1569.
42. Moldovan D, Novischi A. Word sense disambiguation of WordNet glosses[J]. Computer Speech Language, 2004, 18(3): 301-317.
43. Ercan G, Haziyev F. Synset expansion on translation graph for automatic wordnet construction[J]. Information Processing Management, 2019, 56(1): 130-150.
44. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning[J]. Neurocomputing, 2021, 452: 48-62.

 doi:10.20944/preprints202403.0316.v1

45. Namazifar M, Hazarika D, Hakkani-Tür D. Role of bias terms in dot-product attention[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.
46. Sokolova M, Japkowicz N, Szpakowicz S. Beyond Accuracy, F-score and ROC: A family of discriminant measures for performance evaluation[C]//Australasian joint conference on artificial intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 1015-1021.
47. Tortorelli D A, Michaleris P. Design sensitivity analysis: Overview and review[J]. Inverse problems in Engineering, 1994, 1(1): 71-105.
48. Ghosh A, Lan A. Contrastive learning improves model robustness under label noise[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 2703-2708.
49. Sauter T, Lobashov M. End-to-end communication architecture for smart grids[J]. IEEE Transactions on Industrial Electronics, 2010, 58(4): 1218-1228.
50. Aoudia F A, Hoydis J. Model-free training of end-to-end communication systems[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(11): 2503-2516.
51. Sowa J F. Semantic networks[J]. Encyclopedia of artificial intelligence, 1992, 2: 1493-1511.
52. Voorhees, E. M. (1993). Using WordNetTM to Disambiguate Word Senses for Text Retrieval. In Proceedings of the ACM-SIGIR'93 (pp. 171-179), Pittsburgh, PA, USA.
53. Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word correlation Tasks. In Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, pages 30–35, Berlin, Germany. Association for Computational Linguistics.
54. Corley, C., Mihalcea, R. (2005). Measuring the Semantic correlation of Texts. In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment (pp. 13-18), Ann Arbor, MI.