Effectiveness Analysis of Word Sense Disambiguation Using Examples of Word Senses from WordNet

Hiroshi Sekiya
Graduate School of Sci. and Eng., Ibaraki Univ. / 4-12-1 Nakanarisawacho, Hitachi City, Ibaraki Prefecture, 316-8511, Japan
22nm727x@vc.ibaraki.ac.jp

Minoru Sasaki
Graduate School of Sci. and Eng., Ibaraki Univ. / 4-12-1 Nakanarisawacho, Hitachi City, Ibaraki Prefecture, 316-8511, Japan
minoru.sasaki.01@vc.ibaraki.ac.jp

Abstract

In Word Sense Disambiguation (WSD), several studies have proposed systems that incorporate dictionary glosses. The use of glosses expressed in short sentences can improve the accuracy of the WSD system. However, since many glosses are short sentences, additional lexical information could improve accuracy. In this study, we propose a method to incorporate examples of word senses described in WordNet 3.0 as new lexical information into BEM, a WSD system, and analyze the effectiveness of the examples of word senses. Specifically, examples of word senses are input into BEM, and the [CLS] vector or target word vector is taken from the output word embedding representation sequence and used as the sense embedding representation. In the experiment, out of six evaluation sets, including the development set, the F1 score decreased in five evaluation sets and improved in one evaluation set. Since the F1 score improved in one evaluation set, we expect that the use of examples of word senses would be effective.

1 Introduction

Word sense disambiguation (WSD) is one of the major tasks in natural language processing (NLP). WSD is the process of identifying the most appropriate sense for a polysemous word in context. This technique is crucial in many applications in other areas of NLP, such as machine translation (Nguyen et al., 2018), information extraction (Chai and Biermann, 1999), text summarization (Rahman and Borah, 2020), and so on.

Previous research has shown that incorporating lexical information, such as glosses, into a WSD system improves accuracy (Luo et al., 2018a, b; Blevins and Zettlemoyer, 2020). Glosses have been found to be effective for both most frequent sense (MFS) and less frequent sense (LFS). However, many of the glosses are written in short sentences, and it is not clear whether the glosses effectively capture the information in the sense. We expect that the use of information on many senses, rather than just glosses, will capture the characteristics of senses.

To solve this problem, we propose a WSD method to use examples of word senses from WordNet 3.0 (Miller, 1995) as additional lexical information in BEM (Blevins and Zettlemoyer, 2020). In the proposed system, the target word vector or [CLS] vector of examples of word senses is used as the sense embedding representations. We expect that the use of examples of word senses as well as glosses can effectively capture the characteristics of word senses. We compare the performance of this proposed method with that of BEM to test the effectiveness of examples of word senses in WSD.

2 Related Work

There are two types of word sense disambiguation (WSD) tasks: the Lexical Sample Task, in which the target words for WSD are predefined, and the All-words WSD, in which all polysemous words in a

sentence are target words. This study is categorized as an All-words WSD.

The BEM by Blevins et al. consists of a context encoder that represents the target word and its surrounding context and a gloss encoder that represents the sense glosses, representing the target words and senses in the same embedding space. These two encoders are initialized with a pre-trained model, BERT (Devlin et al., 2019), and are jointly fine-tuning. This method outperforms the results for All-words WSD in English presented in the previous study by Raganato et al. (2017b). In this study, the model was trained in BEM by creating a new sense embedding representation of examples of word senses.

It has been shown that lexical information such as glosses is a valuable resource for improving the accuracy of WSD. Lesk (1986) used overlap between sense glosses and the context of the target word to estimate the sense of the target word. This method was later extended to incorporate WordNet graph structure (Banerjee and Pedersen, 2003). It has also been extended to incorporate word embeddings (Basile et al., 2014). In a recent study, Luo et al. (2018a, b) input sense glosses into a neural WSD system and significantly improved accuracy.

3 BEM

In this section, we introduce the BEM proposed by Blevins et al. The model structure of the BEM is shown in Figure 1. BEM is a supervised WSD system designed to efficiently utilize sense glosses that define a less frequent sense. BEM is composed of two independent encoders: a context encoder that represents the target word and surrounding context, and a gloss encoder that embeds the sense glosses. Each encoder is a deep transformer network initialized with BERT to take advantage of the word sense information obtained from prior training (Coenen et al., 2019; Hadiwinoto et al., 2019). Thus, the input to each encoder is padded with BERT-specific start [CLS] and end [SEP] symbols.

BEM is designed to encode contextualized target words and sense glosses independently (Bromley et al., 1994; Humeau et al., 2019), and each of these models is initialized with a BERT-base.

The context encoder T_c takes as input a context sentence c containing the target words w for WSD. c is represented by $c = c_0, c_1, c_2, \cdots, w_i, \cdots, c_n$,

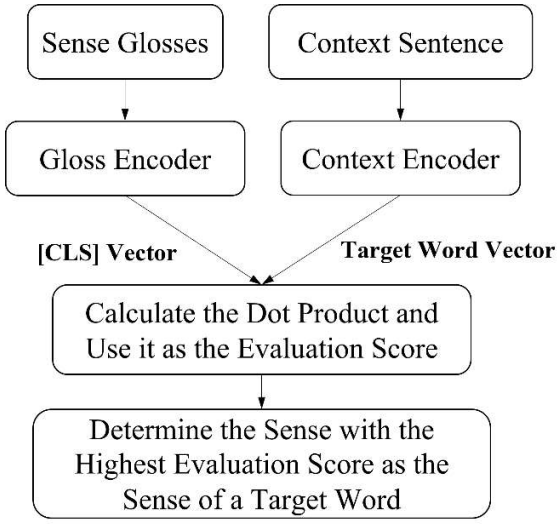

Figure 1. Model Structure of BEM

where w_i is the i^{th} target word of the context sentence. The context encoder outputs a context-aware word embedding representation sequence r. The target word vector in the word embedding sequence r is denoted by r_{w_i}, where r_{w_i} is the i^{th} representation output by T_c. r_{w_i} is given by

$$r_{w_i} = T_c(c)[i]$$

For words tokenized into multiple subword by the BERT tokenizer, the word is represented by the average representation of the subword parts. For example, if the j^{th} through k^{th} tokens correspond to the subword of the i^{th} word, r_{w_i} is given by

$$r_{w_i} = \frac{1}{k-j}\sum_{l=j}^{k}(T_c(c)[l])$$

The gloss encoder T_g takes as input the gloss $g_s = g_0, g_1, \cdots, g_m$ that define the sense s. The [CLS] vector, which is the first representation in the word embedding representation sequence output by gloss encoder, is the global representation of s. The global representation of s is denoted by r_s, and r_s is given by

$$r_s = T_g(g_s)[0]$$

As shown in the following equation, each candidate sense $s \in S_w$ of the target word w is given a score by taking the dot product of r_w and every r_s.

$$\phi(w, s_i) = r_w \cdot r_{s_i}$$

where i is $i = 0, \cdots, |S_w|$. When evaluating, the meaning $\hat{s}$ of the target word w is predicted to be

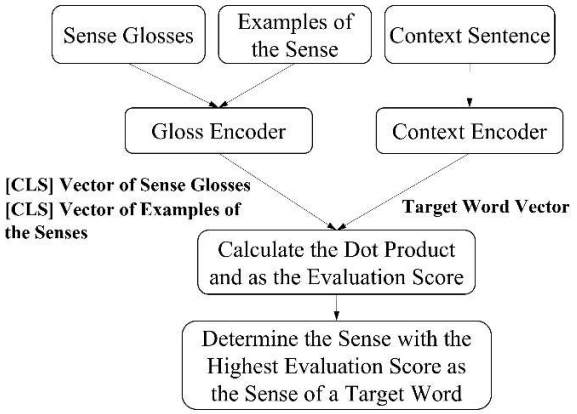

Figure 2. Structure of the Model Using the [CLS] Vector of Examples of Word Senses

the $s_i \in S_w$ with the highest score for the dot product of r_{s_i} and r_w.

For the score of each candidate sense of the target word w, the BEM is trained using a loss function, cross-entropy loss. Given a word-sense pair (w, s_i), the loss function of the system is given by

$$\mathcal{L}(w, s_i) = -\phi(w, s_i) + \log \sum_{j=0}^{|S_w|} \exp\left(\phi(w, s_i)\right)$$

4 Method

In this study, we input examples of word senses into the BEM to estimate the sense of the target word. Specifically, we use the [CLS] vector and the target word vector obtained from the examples of word senses as the sense embedding representations. Three main types of examples of word senses representations are used to train the BEM, and three types of models are created using the examples of word senses representations. The method of representation of examples of word senses is explained in detail in Sections 4.1, 4.2, and 4.3. The examples of word senses are obtained from WordNet3.0.

4.1 Word Sense Disambiguation Using the [CLS] Vector of Examples of Word Senses

In this section, we present an approach using [CLS] vector of examples of word senses. The structure of the model using the [CLS] vector of examples of word senses is shown in Figure 2. If there are multiple examples of the sense for a sense, one of them is obtained. The obtained examples of the sense are input to the gloss encoder, and the [CLS] vector is extracted from the word embedding representation sequence output from the gloss encoder. We use the extracted [CLS] vector as the sense embedding representation. We take the dot product of the target word vector of the context sentence output by the context encoder and the [CLS] vector of the sense glosses output by the gloss encoder. In addition, we take the dot product of the target word vector of the context sentence output by the context encoder and the [CLS] vector of the examples of the sense output by the gloss encoder. The results of the dot product calculation are used as the score of each candidate sense. Then, the sense with the highest score among each candidate sense is estimated as the sense of target word.

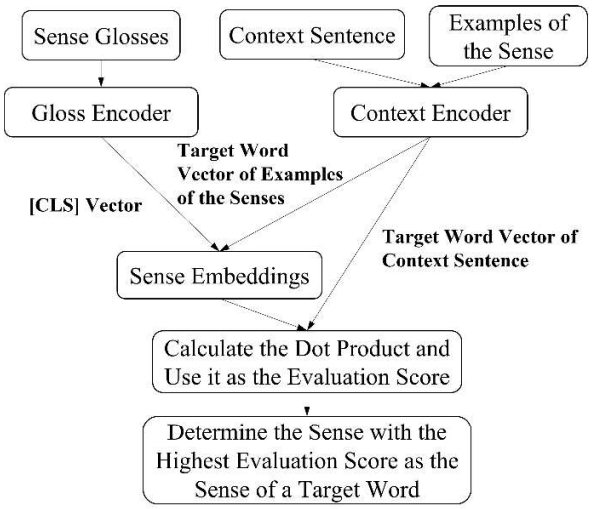

Figure 3. Structure of the Model Using the Target Word Vector of Examples of Word Senses

4.2 Word Sense Disambiguation Using the Target Word Vector of Examples of Word Senses

In this section, we present an approach using the target word vector of examples of word senses. The structure of the model using the target word vector of examples of word senses is shown in Figure 3. If there are multiple examples of the sense containing the target word for a sense, one of them is obtained. For example, if the target word is "review," the examples of the sense with words of the same type as "review" in the sentence are obtained. The

Dataset	Part-of-speech of the Target Word	Number of Senses	Number of Annotated Example
SE07	Nouns, Verbs	375	455
SE2	Nouns, Verbs, Adj., Adv.	1335	2282
SE3	Nouns, Verbs, Adj., Adv.	1167	1850
SE13	Nouns	827	1644
SE15	Nouns, Verbs, Adj., Adv.	659	1022

Table 1. Characteristics of the Senseval/SemEval Dataset

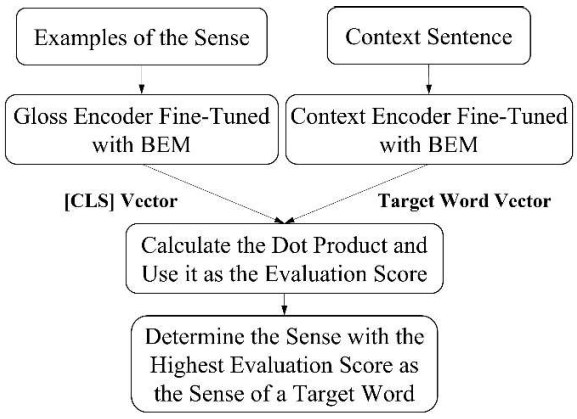

Figure 4. Structure of the Model with Vector of Sense Glosses Updated with Examples of Word Senses

obtained examples of the sense are input to the context encoder, and the target word vector is extracted from the word embedding representation sequence output from the context encoder. We use the extracted target word vector as the sense embedding representation. We take the dot product of the target word vector of the context sentence output by the context encoder and the [CLS] vector of the sense glosses output by the gloss encoder. In addition, we take the dot product of the target word vector of the context sentence output by the context encoder and the target word vector of the examples of the sense output by the context encoder. The results of the dot product calculation are used as the score of each candidate sense. Then, the sense with the highest score among each candidate sense is estimated as the sense of target word.

4.3 Word Sense Disambiguation with the vector of sense glosses updated with examples of word senses

In this section, we present an approach that updates the vector of sense glosses with examples of word senses. The structure of the model with the vector of the sense glosses updated with the examples of word senses is shown in Figure 4. If there are multiple examples of the sense for a sense, one of them is obtained. If no examples of the sense exist for a sense, the sense gloss is used as the examples of the sense as a substitute. The obtained examples of the sense are input to the gloss encoder, and the [CLS] vector is extracted from the word embedding representation sequence output from the gloss encoder. We use the extracted [CLS] vector as the sense embedding representation. We then update the word sense vector of the sense glosses by fine tuning each encoder with the sense glosses, followed by fine tuning with the examples of the sense.

5 Experiments

5.1 Datasets

We use the WSD framework created by Raganato et al. (2017b) to evaluate model performance. As training data, we use the SemCor corpus (Miller et al., 1993), a large dataset of manually annotated word senses from WordNet. SemCor contains 226,036 annotated examples covering 33,362 senses. We use the SemEval-2007 (SE07) dataset (Pradhan et al., 2007) as the development set. As evaluation sets, we use the Senseval-2 (SE2; Palmer et al. (2001)), Senseval-3 (SE3; Snyder and Palmer (2004)), SemEval-2013 (SE13; Navigli et al. (2013)), SemEval- 2015 (SE15; Moro and Navigli (2015)), and ALL datasets. The ALL dataset is a dataset that concatenates all development and evaluation sets. In addition, all sense glosses and examples of word senses used in this system are taken from WordNet 3.0.

The Senseval/SemEval dataset is a dataset focused on the WSD task. The characteristics of each dataset are shown in Table 1.

	SE07	SE2	SE3	SE13	SE15	ALL
BEM	**74.5**	**79.4**	77.4	**79.7**	**81.7**	**79.0**
BEM1	73.8	78.6	76.7	77.4	80.6	77.8
BEM2	73.6	79.3	**78.1**	78.2	80.3	78.5
BEM3	73.0	77.6	76.2	76.5	80.1	77.0

Table 3. F1 Score for Each Model

Model	Explanation
BEM1	Use [CLS] vector of examples of word senses (Section 4.1)
BEM2	Use target word vector of examples of word senses (Section 4.2)
BEM3	Vector of sense gloss updated with examples of word senses (Section 4.3)

Table 2. The Model Proposed in this Study

5.2 Experimental Setup

The models presented in Sections 4.1, 4.2, and 4.3 are shown in Table 2. We compare the F1 score of these models with the F1 score of the BEM to analyze the effectiveness of the examples of word senses. BEM1 and BEM3 are trained in the Google Colaboratory Pro+ and BEM2 are trained in the Google Colaboratory Pro environment.

Each model proposed in this study is trained in the Google Colaboratory environment, which has a limited runtime, and therefore epochs that can be done in a single run is limited. Also, each model saves only the model with the highest F1 score and resumes training. Therefore, if current run does not obtain a higher F1 score than the previous run, the model training is terminated at that point. As a result, epochs vary from model to model. BEM1 is trained 14 epochs, BEM2 is trained 6 epochs, and BEM3 is trained 17 epochs. We use context batch size 4; BEM1 and BEM2 use gloss batch size 128 and BEM3 uses gloss batch size 256.

5.3 Evaluation Method

We use the development and evaluation sets presented in Section 5.1 to evaluate the performance of our models by determining the F1 score of each model. We used the model with the highest F1 score in the development set to obtain the F1 score in the evaluation set.

	Zero-shot Words
BEM	91.2
BEM1	92.2
BEM2	**92.9**
BEM3	90.6

Table 4. F1 Score of Zero-shot Words

5.4 Results

The F1 score for BEM, BEM1, BEM2, and BEM3 on the development set and the five evaluation sets are shown in Table 3. BEM1, BEM2, and BEM3 with the use of examples of word senses resulted in lower F1 score than BEM in all development and evaluation sets except the Senseval-3 evaluation set. For the Senseval-3 evaluation set, the highest score is 78.1% for BEM2, which is a 0.7% improvement over BEM.

The F1 score for zero-shot words in the ALL evaluation set are shown in Table 4. zero-shot words are words that did not appear in the training data. BEM3 results in an F1 score below that of BEM, while BEM1 and BEM2 outperform the F1 score of BEM by up to 1.7%.

6 Discussion

The model using the examples of word senses results in lower F1 score than the original BEM, except for the senseval-3 evaluation set. This indicates that the examples of word senses are noise and have a negative impact on the system. However, since the senseval-3 evaluation set improves the F1 score, we expect that the use of examples of word senses may be effective if we devise a way to use them.

Among the models using examples of word senses, BEM2 shows the best results. therefore, we can say that the most effective way to incorporate examples of word senses into a model is to represent the examples of word senses as the target word vector. The reason for the best results with the target word vector is thought to be that unlike the sense glosses, the examples of word senses the usage of

	BEM-correct	BEM-wrong
BEM1-correct	-	249
BEM1-wrong	333	-
BEM2-correct	-	215
BEM2-wrong	249	-
BEM3-correct	-	251
BEM3-wrong	391	-

Table 5. Comparison of the number of correct and incorrect senses between BEM and BEM1~3. For example, 249 is the number of senses that are wrong in BEM but correctly estimated in BEM1.

the sense, not the meaning of the sense. This may have resulted in the [CLS] vector having a low F1 score because the [CLS] vector could not effectively express the features of the sense meaning due to the presence of many words in the examples of word senses that have a low similarity to the sense meaning. BEM2 also has a higher F1 score than the original BEM on the senseval-3 evaluation set. Therefore, we expect that the F1 score will be improved by devising the use of examples of word senses. For example, if the target word is "review," we are considering using "reviewed," the past tense of review, or "reviews," the plural of review, as the target word.

We find that BEM1 and BEM2 have a higher F1 score for zero-shot words than the original BEM. Therefore, we think that using the examples of word senses as sense embedding representations can effectively represent the features of words that do not appear in the training data.

To analyze the experimental results of this study in detail, we examined the number of senses that were wrong in the BEM but correct in the model using examples of word senses, and the number of senses that were correct in the BEM but wrong in the model using examples of word senses, based on the estimation results of the ALL evaluation set. The results of the survey are shown in Table 5. The survey results show that the proposed system can correctly estimate senses that are incorrectly estimated by BEM. However, more than that number, the proposed system incorrectly estimates senses that are correctly estimated by BEM. In particular, BEM1 and BEM3 incorrectly estimate 84 and 140 more than the original BEM, respectively. In contrast, BEM2 is estimated with 34 more errors than the original BEM. This indicates that although BEM2 has fewer correctly estimated senses than BEM1 and BEM3, it has less negative impact on the system than BEM1 and BEM3. These results indicate that although the examples of word senses have a negative impact on the system, there are many cases where a sense that is wrong in the original BEM is correctly estimated by the BEM using the examples of word senses. Therefore, we anticipate that the system's accuracy could be improved by devising ways to use examples of word senses.

	Nouns	Verbs	Adj.	Adv.
BEM1	4.49%	**5.39%**	4.40%	2.60%
BEM2	3.26%	**4.30%**	3.14%	2.31%
BEM3	5.02%	**7.20%**	4.50%	3.76%

Table 6. Percentage of Parts of Speech of Newly Mistaken Word Senses When Examples of Word Senses are Used

	Nouns	Verbs	Adj.	Adv.
SE07	2.52%	**6.42%**	-	-
SE2	1.97%	**5.80%**	2.47%	2.36%
SE3	2.78%	1.70%	**3.71%**	0%
SE13	**4.38%**	-	-	-
SE15	3.39%	**4.78%**	3.75%	2.50%
ALL	3.26%	**4.30%**	3.14%	2.31%

Table 7. Percentage of Parts of Speech of Newly Mistaken Word Senses in Each Evaluation Set (BEM2)

To analyze in detail the findings presented in Table 5, we investigated the percentage of newly mistaken word senses parts of speech in the ALL evaluation set when using examples of word senses. The results of the survey are shown in Table 6. The survey results show that the use of example word senses increases the number of mistakes most frequently in the identification of verb senses. Therefore, we consider that to improve the accuracy of the WSD, it is necessary to reconsider the method of extracting examples of word senses related to verbs.

To analyze the cause of the decrease in accuracy in the evaluation sets other than senseval3, we examined the proportion of parts of speech of newly mistaken word senses in BEM2 in each evaluation set. The results of the survey are shown in Table 7. The survey results show that the senseval3 evaluation set with improved accuracy has fewer errors in verb senses, while the other evaluation sets have more errors in verb senses. This suggests that the accuracy of identifying verb senses contributes

to the difference in accuracy of each evaluation set. One possible reason for the high number of errors in verb senses could be that only examples of word senses containing words of the same type as the target word were used. Therefore, we consider that accuracy could be improved by increasing the number of examples of word senses used by extracting examples of word senses that include words converted to the past tense, plural, etc.

7 Conclusion and Future Work

In this study, we analyzed the effectiveness of examples of word senses in WSD by using examples of word senses retrieved from WordNet 3.0 as sense embedding representations and incorporating them into BEM. The results showed that the use of examples of word senses decreased the overall performance, but the model using the target word vector of the examples of word senses slightly improved the F1 score on the Senseval-3 evaluation set. Additionally, the F1 score of zero-shot words was improved. Thus, we expect that although examples of word senses have a negative impact on BEM, they can be effective if examples of word senses are used in a different way.

For future work, we are considering reexamining the extraction method when using the target word vector of examples of word senses, such as targeting not only those that are isomorphic to the target word, but also those that have been transformed into plural or past tense forms. We are also considering other ways to use examples of word senses to mitigate data bias, such as using examples of word senses only in the case of LFS without using examples of word senses in the case of MFS.

References

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017b. Word sense disambiguation: A unified evaluation framework and empirical comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 99–110.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pages 288–297.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viegas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. arXiv preprint arXiv:1906.02715.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. 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), pages 5300–5309.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In Proceedings of the workshop on Human Language Technology, pages 303–308. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sackinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In Advances in neural information processing systems, pages 737–744.

Joyce Yue Chai and Alan W Biermann. 1999. The use of word sense disambiguation in an information

extraction system. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence (AAAI '99/IAAI '99), pages 850-855.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, pages 21 – 24.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In Proceedings of the 5th annual international conference on Systems documentation, pages 24–26. ACM.

Nazreena Rahman and Bhogeswar Borah. 2020. Improvement of query-based text summarization using word sense disambiguation. Complex Intelligent Systems, vol. 6 No.7, pages 75-85.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1591–1600.

Quang-Phuoc Nguyen, Anh-Dung Vo, Joon-Choul Shin and Cheol-Young Ock. 2018. Effect of Word Sense Disambiguation on Neural Machine Translation: A Case Study in Korean. in IEEE Access, vol. 6, pages 38512-38523.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 222–231.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. In Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007), pages 87–92.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. arXiv preprint arXiv:1905.01969.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In IJCAI.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1006-1017.