Article

Word Sense Disambiguation Using Clustered Sense Labels

Jeong Yeon Park, Hyeong Jin Shin and Jae Sung Lee *

Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea;
parkjeongyeon@chungbuk.ac.kr (J.Y.P.); shinhj@chungbuk.ac.kr (H.J.S.)

* Correspondence: jasonlee@chungbuk.ac.kr

Abstract: Sequence labeling models for word sense disambiguation have proven highly effective when the sense vocabulary is compressed based on the thesaurus hierarchy. In this paper, we propose a method for compressing the sense vocabulary without using a thesaurus. For this, sense definitions in a dictionary are converted into sentence vectors and clustered into the compressed senses. First, the very large set of sense vectors is partitioned for less computational complexity, and then it is clustered hierarchically with awareness of homographs. The experiment was done on the English Senseval and Semeval datasets and the Korean Sejong sense annotated corpus. This process demonstrated that the performance greatly increased compared to that of the uncompressed sense model and is comparable to that of the thesaurus-based model.

Keywords: word sense disambiguation; clustering; deep learning; sense vocabulary

1. Introduction

In computational semantic analysis, word sense disambiguation (WSD), i.e., finding the correct sense of a word in a given context, has long been a challenge in natural language understanding. Recently, it has mostly been studied with machine learning models typically using the supervised, unsupervised, and knowledge-based approaches [1–4]. The supervised approach [5,6] uses a sense annotated corpus to train the models and usually suffers from data sparseness. The unsupervised approach [7,8] uses raw text to find a word cluster with the same sense, discriminating it from others. However, these senses often do not entirely match those defined in a dictionary or a thesaurus; therefore, clustering such senses for explainable applications is not trivial. The knowledge-based approach [9,10] uses glossary information, usually from a dictionary or thesaurus, to match with the context of a target word. The glossary information is usually insufficient to cover all the contexts of a target word.

To date, supervised learning becomes popular and is considered an effective approach, but suffers from processing unknown senses in the training data. To alleviate this problem, the supervised approach combined with the knowledge-based approach has recently been researched. Advances in computing power enables the various WSD algorithms to utilize the large size of dictionary and corpus [11]. Word embeddings and contextual word vectors are especially used widely based on neural network models in a pre-trained manner [12–14]. Huang et al. [15] used the sense definition in a glossary to match the context of a target word by using the sentence pair comparison function of BERT [16]. The context vector and the glossary vector are trained in a supervised way for fine-tuning of a generic BERT system. Kumar et al. [17] extended this idea to include thesaurus information into the sense definition and merge them for continuous vector representation. However, this approach requires a large amount of memory to process all the senses of homographs and their related glossaries. Moreover, when decoding it requires multiple scans for the homographs' sense glossaries.

The sense vocabulary compression (SVC) method tackles the unknown sense problem differently by compressing sense tags [18]. It finds the optimal set of sense tags by

Citation: Park, J.Y.; Shin, H.J.; Lee, J.S. Word Sense Disambiguation Using Clustered Sense Labels. *Appl. Sci.* **2022**, *12*, 1857. <https://doi.org/10.3390/app12041857>

Academic Editor: Valentino Santucci

Received: 7 January 2022

Accepted: 8 February 2022

Published: 11 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

decreasing the number of sense tags using sense relations, such as synonyms, hypernyms, and hyponyms in a thesaurus. The synonyms are simply compressed to one representative sense, known as the synset tag. The hypernym and hyponym sense tags are compressed by using the highest common ancestor tag not shared with the other senses in the homograph. Figure 1 shows an example of compressed sense vocabularies in hypernym hierarchical relations. There are two kinds of homographs in the figure: *fund* and *bank*. The dotted rectangles show the maximum subtrees not shared with the other senses in the homographs. Each root in the subtrees is a representative sense. For example, *nature#1* is a representative sense of *bank#7* and *river#1*; *financial institution#1* is that of *fund#1* and *bank#4*.

The compressed sense vocabulary achieves engineering efficiency in processing smaller output parameters and learning efficiency for sparse or unknown sense tags to be included implicitly during the learning phase. Once the compressed sense vocabulary is obtained, no additional memory or multiple scans are required for decoding; therefore, it is extremely compact and fast.

Figure 1. Sense vocabulary compression using hypernym and hyponym relations [18].

However, for sense compression, the SVC method requires a superior quality thesaurus. Although there are thesauruses for many languages, most are not rich enough or freely available [19]. In this paper, we propose an alternative sense vocabulary compression method without using a thesaurus. Instead, the sense definitions in a dictionary become the embedding vectors and are clustered to find representative sense vocabularies, which are the compressed sense vocabularies. Our paper's contributions are the following: (1) a novel method is proposed to compress sense vocabularies without using a thesaurus and (2) a practically effective method is proposed for sense vocabulary clustering, which is required to handle the vast number of senses with modern clustering algorithms.

2. Sense Definition Clustering

Word sense definitions have been utilized for WSD tasks either as a surface form or a vector form [9,15,17,20]. In this paper, we also use the word sense definition in a dictionary to calculate the vectors for the senses, assuming that the sense is represented by the vector of the sense definition. To produce the sense definition vector (SDV), we used Universal Sentence Encoder [21], which, at the time, was the best sentence vector generation program available. SDVs are clustered into groups of similar vectors. The similarity is measured using normalized Euclidean distance, which ranges from 0 to 1 as defined in the following:

$$ndist(x - y) = \frac{1}{M} \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

where M is the maximum distance in all the SDV pairs, and n is the dimension of x, y vectors.

A pair of vectors are clustered as a group under two conditions: 1. the similarity value is above a threshold, and 2. the new group does not include more than one sense in the same homograph set. This is accomplished by the sense definition clustering algorithm (SDC) as shown in Algorithm 1, which is a slight modification of the hierarchical agglomerative clustering (HAC) algorithm [22].

Algorithm 1 Sense Definition Clustering (SDC)

- 1: Input: SDVs represented as w_i ($1 \leq i \leq N$)
 - 2: Output: clustered SDVs
 - 3: Place each w_i in a group, g_i , as a set with one element
 - 4: Let $G = \{x \mid x = g_i \ (1 \leq i \leq N)\}$
 - 5: Calculate the similarity matrix for each pair of (g_j, g_k) where $j \neq k$.
 - 6: **repeat**
 - 7: Find the most similar pair (g_j, g_k) with their similarity above the threshold
 - 8: Make a new group g_{jk} with union (g_j, g_k) where $j \neq k$ and only if any pair of the
 padd group is not in the same homograph set
 - 9: If g_{jk} equals null, then terminate
 - 10: Let $G = G - g_j - g_k + g_{jk}$
 - 11: Recalculate the similarity matrix
 - 12: **until** $|G| \leq 1$
-

The time complexity of the SDC algorithm is $O(n^3)$, the same as HAC. Moreover, the number of SDVs is usually too large to be run by the complex clustering algorithm; in the case of WordNet the size is 207 K. Therefore, for practical reasons, we divided the initial large set of SDVs into smaller partitions to start the hierarchical agglomeration SDC with efficiently sized partitions rather than the set of singleton clusters [23].

We first divide the data by parts of speech (noun, verb, adjective, adverb) and then utilize a flat clustering program, such as the affinity propagation program (AP) [24] or the k -means program, Ref. [22] for further partitioning. Affinity propagation is good for partitioning with an appropriate number of clusters. However, it fails to partition very large datasets such as noun partition, because its algorithm needs considerable space to process various matrix calculations. In those cases, AP is run to determine the cluster size k for a randomly chosen subset of the dataset, and for partitioning, the k -means program is run using the k size determined on the original dataset.

Once we have partitions, we apply the SDC algorithm to each partition to build a compressed sense vocabulary group within the partition. We then apply the SDC again to the newly grouped senses externally to find a higher common ancestor. The new clusters are given cluster numbers, which are used as compressed sense numbers. Figure 2a shows the overall flow of sense clustering. Figure 2b shows the clustering example in detail. In Figure 2b, the three red dots represent the different senses of a homograph. In the partitioning process at (2), more than one red dot can be included in a group, but in the clustering process at (3) and (4), the red dots cannot be located in the same group even though they are closer than others in Euclidean distance.

Figure 2. Process of sense clustering.

3. Deep-Learning Model for Word Sense Disambiguation

A sequence labeling model is used for WSD. We chose a BERT-based model, which performed well, especially in WSD [15]. A transformer [25] encoder is added above the BERT to obtain the overall sentence information. We chose a single-task model [20], because, in our case, a multi-task model would not work well [26]. Figure 3 shows the architecture processing a sample sentence with clustered sense tags. Every word is divided into word pieces for input and the sense number is given to the first token of the word piece in the output.

Figure 3. BERT-WSD: BERT-based model for word sense disambiguation.

4. Experiment and Result

4.1. Experiment Setting

We used the affinity propagation program and the k -means program provided in scikit-learn (2020) [27], and the values of the parameters used in our experiment are shown in Table 1a,b. For the BERT-WSD sequence model, we used the hyper-parameter values shown in Table 1c. The pre-trained BERT-base-cased model [28] was used for English and the KorBERT-morp model [29] for Korean.

Table 1. System hyper-parameter values.

(a) <i>k</i>-means	
init	<i>k</i> -means++
max iter	300
tol	1×10^{-4}
dist	Euclidean
(b) affinity propagation	
damping	0.7
max iter	50,000
converg iter	15
dist	Euclidean
(c) BERT-WSD model	
learning rate	1×10^{-4}
embed size	768
dropout rate	0.1
trans layer num	6
trans layer dim	2048
atten head num	8

The proposed model was evaluated on English and Korean data. For the English data, we used the sense definitions in WordNet 3.0 [30] for sense vector generation to compare with SVC using the same thesaurus [18]. For the same reason, we used SemCor 3.0 for training and Senseval and Semeval data for testing. For Korean data, we used the Standard Korean Language Dictionary [31] for sense vector generation and used the Sejong sense annotated corpus [32] for training and testing of the model.

Table 2 shows the statistics of each sense vocabulary in WordNet 3.0 [30] and in the Standard Korean Language Dictionary. Table 3 shows the statistics of the annotated dataset provided by [33] and that of the Sejong sense annotated corpus. The *Ambiguity* field represents the average number of senses per annotated word instance. Table 4 shows the compression rate at various threshold points. The lemma in the Standard Korean Language Dictionary does not always match the morpheme in the Sejong sense annotated corpus because of the different morpheme analysis principle. In these cases, we morphologically analyzed the lemma with Sejong morpheme annotation principle and tried to match it to the annotated units in Sejong sense annotated corpus. If the trial failed, we ignored the sentences containing the unmatched annotations in the evaluation. The sentences that exceeded 512 bytes after BERT tokenizing were also not used for the evaluation.

4.2. Experiment Result

For evaluating the English data, we followed the evaluation methods used in Semeval, where the SemCor corpus is used for training, and the Senseval and Semeval corpus are used for testing. For a fair comparison with the English data, the Korean Sejong sense annotated corpus was randomly sampled to be the same size as the English data and named the Small set. To see the effect of corpus size, we sampled the Medium and Large sets, respectively, to be twice and thrice the size of the Korean Small set. All the experiments for the Korean data were 3-fold cross validated.

The results are shown in Table 5. Generally, the score increases as the threshold value increases to a certain point, then decreases thereafter. Figure 4 shows the performance changes at each threshold point. The English data showed the best performance with a 69.1% average at threshold point 0.5 with an 89% compression ratio. The Korean data produced a 97.3% average at point 0.3 with a 33% compression ratio. The performances are much better than the baseline at threshold 0.0, which does not use a compressed sense vocabulary. Note that the baseline uses the first senses as default labels for the untrained

senses as done in [18]. The average performance increased from 62.9% to 69.1% on English data and from 95.9% to 97.3% on Korean data.

Table 2. Number of senses in the dictionaries: WordNet 3.0 and Standard Korean Language Dictionary.

POS	Noun	Verb	Adj	Adv	Total
Eng	146,312	25,047	30,002	5580	206,941
Kor	91,457	16,381	4317	2667	114,822

Table 3. Statistics of annotations in all-words WSD datasets (Kor uses Sejong sense annotated corpus).

	Corpus	Sentences	Tokens	Annotations	Sense Types	Word Types	Ambiguity
Eng	SemCor (train)	37,176	802,443	226,036	33,316	20,399	8.9
	Senseval2 (test)	242	5766	2282	1271	1057	7.2
	Senseval3 (test)	352	5541	1850	1144	948	9.3
	Semeval07 (test)	135	3201	455	371	327	10.9
	Semeval13 (test)	306	8391	1644	822	751	6.9
	Semeval15 (test)	138	2604	1022	545	496	7.6
Kor	Small (train)	37,940	881,549	146,767	9393	7684	5.3
	Small (test)	1060	24,533	4108	1671	1587	5.2
	Medium (train)	75,880	1,765,792	294,045	11,805	9249	5.3
	Medium (test)	2120	47,337	7816	2489	2305	5.3
	Large (train)	113,820	2,649,071	440,157	13,299	10,126	5.3
	Large (test)	3180	72,821	12,060	3163	2901	5.3

Table 4. Sense compression ratio at different thresholds.

	Threshold	0.0	0.1	0.3	0.5	0.7	0.9	1.0
Eng	Number of senses	206,941	93,399	44,157	22,052	19,950	14,397	1520
	Compression ratio (%)	0	55	79	89	90	93	99
Kor	Number of senses	114,822	104,183	76,427	59,864	39,179	15,159	2410
	Compression ratio (%)	0	9	33	48	66	87	98

Table 5. WSD F1-score with standard deviation of various thresholds and datasets.

	Threshold	0.0	0.1	0.3	0.5	0.7	0.9	1.0
Eng	Senseval2	66.3	70.9	71.1	70.6	70.6	69.8	67.4
	Senseval3	65.6	71.2	72.7	71.6	71.4	72.5	66.8
	Semeval07	56.0	62.2	63.3	62.9	63.1	61.8	54.7
	Semeval13	64.2	71.2	71.0	71.8	71.5	71.7	70.3
	Semeval15	62.2	66.5	66.8	68.1	67.3	66.6	64.7
	Avg.	62.9 ± 3.4	68.4 ± 3.3	69.0 ± 3.1	69.1 ± 3.0	68.8 ± 2.9	68.5 ± 3.6	64.8 ± 4.9
Kor	Small	95.7 ± 0.2	97.2 ± 0.1	97.1 ± 0.1	96.7 ± 0.0	96.1 ± 0.1	95.4 ± 0.2	94.1 ± 0.0
	Medium	96.0 ± 0.1	97.0 ± 0.1	97.3 ± 0.0	96.9 ± 0.1	96.5 ± 0.1	95.9 ± 0.1	95.0 ± 0.1
	Large	96.1 ± 0.1	97.1 ± 0.1	97.5 ± 0.0	96.9 ± 0.2	96.8 ± 0.1	96.1 ± 0.0	95.1 ± 0.1
	Avg.	95.9 ± 0.1	97.1 ± 0.1	97.3 ± 0.1	96.8 ± 0.1	96.5 ± 0.2	95.8 ± 0.3	94.7 ± 0.4

Figure 4. F1 performance at various threshold values of English Avg., Korean Small, Medium, and Large. Note that Korean performance and English performance are scaled differently on the left-hand and right-hand sides of the Y axis.

5. Discussion

Training size effect: We tested Korean training datasets of three different sizes. As the size increased, the performance increased both in the uncompressed vocabulary (threshold 0.0) and in the compressed vocabulary. Table 5 shows the change in performance on the Small, Medium, and Large datasets: 95.7%, 96.0%, and 96.1% for the uncompressed vocabulary at threshold 0.0, and 97.2%, 97.3%, and 97.5% for the compressed vocabulary. This is because more training data increases the performance by decreasing the quantity of unseen data on testing. Table 6 shows that the percentage of untrained data decreases from 3.3% to 1.8% and 1.2% as the training data size increases. This confirms that more training data are also useful for the compressed vocabulary.

Table 6. Number of sense vocabularies in the trained and untrained test data.

Data	Eng All	Kor Small	Kor Medium	Kor Large
Trained	6072 (83.7%)	3972 (96.7%)	7674 (98.2%)	11,918 (98.8%)
Untrained	1181 (16.3%)	136 (3.3%)	142 (1.8%)	142 (1.2%)
Total	7253 (100.0%)	4108 (100.0%)	7816 (100.0%)	12,060 (100.0%)

Performance difference between English and Korean data: The performance on Korean data is much higher than on the English in the highest average F1 value: 97.3% versus 69.1%. We conjecture that this difference is caused by the following:

1. The ambiguity of English is higher than Korean. It ranges from 6.9 to 10.9 in English and from 5.2 to 5.3 in Korean (shown in Table 3). This is because English senses are categorized more specifically than Korean, as shown in Table 2; 206 K versus 114 K senses. This means that English sense labeling prediction should be more precise than Korean, which makes the performance gain relatively difficult.

2. The configuration of test corpus is different. English test corpus used the data of different domains: SemCor for training, Senseval and Semeval for testing, whereas Korean used the shuffled data of the same corpus (Sejong) for 3-fold cross validation testing. It is verified by Table 6, showing that the ratio of untrained data in English test data is much higher than Korean, 16.3% versus 1.2% to 3.3%. This fact benefits the performance of Korean data.

3. For Korean BERT input, normalized words are used, whereas raw words are used for English input. English language is inflectional and does not have many variations in word form, but Korean language is agglutinative and has many variations in word ending [34,35].

Because these productive variations of Korean language degrade the performance of the language model, some Korean language models are trained with preprocessing the input words to avoid the degradation. We used KorBERT-morp model for Korean BERT, which is the Korean language model using normalized input words through the preprocessing of morphological analysis and part of speech tagging. We conjecture that this normalized word with part of speech tag rather benefits the performance of Korean dataset, because it gives more information to the BERT model than a simple word.

Performance increase ratio: The performance of English data increased by 6.2 percentage points, from 62.9% to 69.1%, whereas that of Korean data increased by 1.4 percentage points, changing from 95.9% to 97.3%. There is a much higher performance improvement on English data. This is because the baseline performance for English data is significantly lower than for Korean data, which allows greater improvements more easily than a high baseline performance.

Compression level: Table 7 shows the comparison of (a) the model with a thesaurus and (b) the model without a thesaurus. The best value of (a) is higher than (b), i.e., 75.6% versus 69.1%. This means that a manually constructed thesaurus is better than our clustering method for sense label compression. Nonetheless, both models perform better than the baseline models that do not use compressed labels and have the highest performance when compression is used with appropriate relations or threshold restrictions.

Table 7. Performance comparison of SVC model and proposed model at various compression levels in F1 measure.

(a) SVC model [18] using thesaurus.	
Comp. Level	F1 Value
Baseline (Ensemble)	63.4
Synonyms	75.1
Hypernyms	75.6
All relation	73.9
(b) Our model without using thesaurus.	
Comp. Level	F1 Value
Baseline	62.9
Threshold 0.1	68.4
Threshold 0.5	69.1
Threshold 1.0	64.8

Use of Korean thesaurus: The sense vocabulary compression method [18] must have both a thesaurus and the corresponding sense annotated corpus. However our method needs a sense annotated corpus and the corresponding sense definitions contained in a dictionary or a thesaurus. In the case of Korean language, most of the sense annotated corpus is developed based on a dictionary. Examples are the Sejong sense annotated corpus based on Standard Korean Language Dictionary and the Modu sense annotated corpus [36] based on Urimalsaem dictionary [37]. Korean thesauri that have been developed so far are KAIST's CoreNet [38], Pusan National University's KorLex [39], University of Ulsan's UWordMap [40], Postech's WordNet translation [41], and others. Among these, KorLex, UWordMap, and Postech's WordNet use the same sense tag as Sejong sense annotated corpus. However, KorLex and Postech's WordNet contain only some part of the whole senses in Standard Korean Language Dictionary; KorLex contains only the senses that exist in English WordNet, and Postech's WordNet is partially translated from English WordNet. UWordMap has included most of sense vocabulary of Standard Korean Language Dictionary more recently, which is quite later than the sense annotated corpus release [42].

Modu sense annotated corpus has been recently developed based on the senses defined in Urimalsaem dictionary, which contains more vocabulary and uses finer sense number hierarchy than Standard Korean Language Dictionary. The corresponding the-

sauros to Urimalsaem dictionary has not been developed yet. Urimalsaem dictionary is being constructed and updated with online crowd sourcing, which makes updating the corresponding thesaurus, if it exists, non-trivial. Our sense clustering method using sense definitions in a dictionary is very useful for this environment, where a corresponding thesaurus does not exist to the sense annotated corpus. For future work, we will apply our method to the Modu sense annotated corpus, for which a full version was not available at implementation time.

Further improvement of clustering: We can further improve our method using a clustering algorithm. For this we should consider the following points: 1. Because we are dealing with huge data points (146K in case of English nouns), the space and time complexity should be small enough to be executed in an appropriate time limit. 2. The clusters with appropriate size should be determined automatically, with the clustering criteria reflecting not only similarity measures but also homograph restrictions that avoid including the same homograph senses in the same cluster.

Most clustering algorithms such as AP [24], HAC [22], GMM [43], and DBSCAN [44] having high computational complexity could not execute our clustering data with huge data points under our workstation level computation environment (Intel i7 CPU and 48GB main memory). Although the k-means algorithm has relatively lower complexity than the other algorithms, it has to determine the optimal number of clusters automatically [45,46], which will cause the complexity to increase. What we have proposed in this paper is one of the practical solutions, and we leave for future work the task to find in complexity level more efficient clustering algorithms for sense compression.

Word embedding vectors retain multiple relations such as synonym, antonym, singular-plural, present-past tense, capital city-country, and so on [47,48]. This is same as SDVs we have used in this paper. As shown in Table 7a, the selective use of the relations increases the performance. Therefore, finding the useful relations from SDVs will improve the quality of clustering.

6. Conclusions

This paper proposes a clustering method to develop compressed sense vocabularies from sense definition vectors. The sense definition vectors are generated from the sense definition sentences in a dictionary, and they are clustered using affinity propagation clustering, *k*-means clustering, and modified hierarchical agglomerative clustering programs. The performance of the word sense disambiguation program based on the BERT sequence labeling program greatly increased from 62.9% to 69.1% in the English experiment, and 95.9% to 97.3% in the Korean experiment regarding the F1-score when compressed sense vocabularies were used. This is comparable to the performance of the thesaurus-based compression model, i.e., 75.6%, considering that only sense definitions were used without knowing the sense hierarchy in the thesaurus. Our proposed method will be useful for finding the compressed sense vocabulary without using a manually constructed thesaurus.

Author Contributions: Conceptualization, J.Y.P. and J.S.L.; Methodology, J.Y.P. and H.J.S.; Software, J.Y.P. and H.J.S.; Validation, J.Y.P.; Investigation, J.Y.P.; Writing-original draft, J.Y.P. and J.S.L.; Writing-review and editing, J.S.L.; Supervision, J.S.L.; funding acquisition, J.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R111A3059545).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

WSD	Word Sense Disambiguation
BERT	Bidirectional Encoder Representations from Transformers
SDV	Sense Definition Vector
SDC	Sense Definition Clustering
HAC	Hierarchical Agglomerative Clustering
AP	Affinity Propagation
POS	Part Of Speech
GMM	Gaussian Mixture Model
DBSCAN	Density Based Spatial Clustering of Applications with Noise

References

1. Navigli, R. Word sense disambiguation: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–69. [\[CrossRef\]](#)
2. McCarthy, D. Word sense disambiguation: An overview. *Lang. Linguist. Compass* **2009**, *3*, 537–558. [\[CrossRef\]](#)
3. Sreedhar, J.; Raju, S.V.; Babu, A.V.; Shaik, A.; Kumar, P.P. Word sense disambiguation: An empirical survey. *Int. J. Soft Comput. Eng. (IJSCE)* **2012**, *2*, 2231–2307.
4. Borah, P.P.; Talukdar, G.; Baruah, A. Approaches for word sense disambiguation—A survey. *Int. J. Recent Technol. Eng.* **2014**, *3*, 35–38.
5. Zhong, Z.; Ng, H.T. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In Proceedings of the ACL 2010 System Demonstrations, Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 78–83.
6. Iacobacci, I.; Pilehvar, M.T.; Navigli, R. Embeddings for word sense disambiguation: An evaluation study. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1; Long Papers, pp. 897–907.
7. Agirre, E.; Martínez, D.; De Lacalle, O.L.; Soroa, A. Two graph-based algorithms for state-of-the-art WSD. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 585–593.
8. Di Marco, A.; Navigli, R. Clustering and diversifying web search results with graph-based word sense induction. *Comput. Linguist.* **2013**, *39*, 709–754. [\[CrossRef\]](#)
9. Lesk, M.E. 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 SIGDOC '86, Toronto, ON, Canada, 8–11 June 1986.
10. Basile, P.; Caputo, A.; Semeraro, G. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014; Technical Papers; Dublin City University and Association for Computational Linguistics: Dublin, Ireland, 2014; pp. 1591–1600.
11. Bhala, R.V.; Abirami, S. Trends in word sense disambiguation. *Artif. Intell. Rev.* **2014**, *42*, 159–171. [\[CrossRef\]](#)
12. Popov, A. Neural network models for word sense disambiguation: An overview. *Cybern. Inf. Technol.* **2018**, *18*, 139–151. [\[CrossRef\]](#)
13. Loureiro, D.; Rezaee, K.; Pilehvar, M.T.; Camacho-Collados, J. Language models and word sense disambiguation: An overview and analysis. *arXiv* **2020**, arXiv:2008.11608.
14. Bevilacqua, M.; Pasini, T.; Raganato, A.; Navigli, R. Recent trends in word sense disambiguation: A survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, QC, Canada, 19–27 August 2021.
15. Huang, L.; Sun, C.; Qiu, X.; Huang, X. GlossBERT: BERT for word sense disambiguation with gloss knowledge. *arXiv* **2019**, arXiv:1908.07245.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Kumar, S.; Jat, S.; Saxena, K.; Talukdar, P. Zero-shot word sense disambiguation using sense definition embeddings. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5670–5681.
18. Vial, L.; Lecouteux, B.; Schwab, D. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *arXiv* **2019**, arXiv:1905.05677.
19. Global WordNet Association. Available online: <http://globalwordnet.org> (accessed on 13 May 2020).
20. Raganato, A.; Bovi, C.D.; Navigli, R. Neural sequence learning models for word sense disambiguation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1156–1167.
21. Cer, D.; Yang, Y.; Kong, S.Y.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Céspedes, M.; Yuan, S.; Tar, C. Universal sentence encoder. *arXiv* **2018**, arXiv:1803.11175.
22. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
23. Posse, C. Hierarchical Model-Based Clustering for Large Datasets. *J. Comput. Graph. Stat.* **2001**, *10*, 464–486. [\[CrossRef\]](#)
24. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [\[CrossRef\]](#)

25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
26. Bjerva, J. Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multitask learning. In Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden, 23–24 May 2017; pp. 216–220.
27. Scikit-Learn. Available online: <https://scikit-learn.org> (accessed on 11 April 2020).
28. BERT-Base-Cased. Available online: <https://github.com/huggingface/transformers> (accessed on 27 October 2021).
29. KorBERT. Available online: <http://aiopen.etri.re.kr> (accessed on 10 June 2019).
30. Fellbaum, C. WordNet. In *Theory and Applications of Ontology: Computer applications*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 231–243.
31. Standard Korean Language Dictionary. Available online: <https://stdict.korean.go.kr/> (accessed on 19 March 2020)
32. The National Institute of the Korean Language. *21st Century Sejong Project Final Result*, Revised ed.; The National Institute of the Korean Language: Seoul, Korea, 2011.
33. Raganato, A.; Camacho-Collados, J.; Navigli, R. 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, Valencia, Spain, 3–7 April 2017; Volume 1, Long Papers, pp. 99–110.
34. Hakkani-Tür, D.Z.; Oflazer, K.; Tür, G. Statistical morphological disambiguation for agglutinative languages. *Comput. Humanit.* **2002**, *36*, 381–410. [[CrossRef](#)]
35. Han, C.H.; Palmer, M. A morphological tagger for Korean: Statistical tagging combined with corpus-based morphological rule application. *Mach. Transl.* **2004**, *18*, 275–297. [[CrossRef](#)]
36. The National Institute of the Korean Language. *The Construction of a Disambiguated Tagged Corpus*; The National Institute of the Korean Language: Seoul, Korea, 2020.
37. Urimalsaem Dictionary. Available online: <https://opendict.korean.go.kr/> (accessed on 2 March 2021).
38. Choi, K.S.; Bae, H.S.; Kang, W.; Lee, J.; Kim, E.; Kim, H.; Kim, D.; Song, Y.; Shin, H. Korean-Chinese-Japanese Multilingual Wordnet with Shared Semantic Hierarchy. In Proceedings of the LREC2004, Lisbon, Portugal, 26–28 May 2004; pp. 1131–1134.
39. Yoon, A. Korean WordNet, KorLex 2.0-A Language Resource for Semantic Processing and Knowledge Engineering. *HAN-GEUL* **2012**, *295*, 163–201. [[CrossRef](#)]
40. Bae, Y.j.; Ock, C. Introduction to the Korean word map (UWordMap) and API. In Proceedings of the 26th Annual Conference on Human and Cognitive Language Technology, Chuncheon, Korea, 18–20 December 2014; pp. 27–31.
41. Lee, C.K.; Lee, G.B. Using WordNet for the automatic construction of Korean thesaurus. In Proceedings of the 11th Annual Conference on Human and Cognitive Language Technology, Jeonju, Korea, 8–9 October 1999; pp. 156–163.
42. Vu, V.H.; Nguyen, Q.P.; Shin, J.C.; Ock, C.Y. UPC: An Open Word-Sense Annotated Parallel Corpora for Machine Translation Study. *Appl. Sci.* **2020**, *10*, 3904. [[CrossRef](#)]
43. Xu, L.; Jordan, M.I. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Comput.* **1996**, *8*, 129–151. [[CrossRef](#)]
44. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **1996**, *96*, 226–231.
45. Märzinger, T.; Kotik, J.; Pfeifer, C. Application of Hierarchical Agglomerative Clustering (HAC) for Systemic Classification of Pop-Up Housing (PUH) Environments. *Appl. Sci.* **2021**, *11*, 11122. [[CrossRef](#)]
46. Dinh, D.T.; Fujinami, T.; Huynh, V.N. Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. In Proceedings of the International Symposium on Knowledge and Systems Sciences, Da Nang, Vietnam, 29 November–1 December 2019; pp. 1–17.
47. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
48. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 5–10 December 2013; Volume 2, pp. 3111–3119.