Density Measure in Context Clustering for Distributional Semantics of Word Sense Induction

Masood Ghayoomi
Faculty of Linguistics, Institute for Humanities and Cultural Studies, Tehran, Iran
m.ghayoomi@ihcs.ac.ir

Received: 22/Nov/2019 Revised: 17/Dec/2019 Accepted: 8/Feb/2020

Abstract
Word Sense Induction (WSI) aims at inducing word senses from data without using a prior knowledge. Utilizing no labeled data motivated researchers to use clustering techniques for this task. There exist two types of clustering algorithm: parametric or non-parametric. Although non-parametric clustering algorithms are more suitable for inducing word senses, their shortcomings make them useless. Meanwhile, parametric clustering algorithms show competitive results, but they suffer from a major problem that is requiring to set a predefined fixed number of clusters in advance.
The main contribution of this paper is to show that utilizing the silhouette score normally used as an internal evaluation metric to measure the clusters' density in a parametric clustering algorithm, such as K-means, in the WSI task captures words' senses better than the state-of-the-art models. To this end, word embedding approach is utilized to represent words' contextual information as vectors. To capture the context in the vectors, we propose two modes of experiments: either using the whole sentence, or limited number of surrounding words in the local context of the target word to build the vectors. The experimental results based on V-measure evaluation metric show that the two modes of our proposed model beat the state-of-the-art models by 4.48% and 5.39% improvement. Moreover, the average number of clusters and the maximum number of clusters in the outputs of our proposed models are relatively equal to the gold data.

Keywords: Word Sense Induction; Word Embedding; Clustering; Silhouette Score; Unsupervised Machine Learning; Distributional Semantic; Density.

1- Introduction

Language is a means of communication to transfer a concept from a producer (speaker) to a recipient (listener). de Saussure [1] believes that language is composed of 'form' and 'meaning'. 'Form' which is tangible and recordable can be represented by a phonological or orthographic system; and 'meaning' which is abstract is very difficult to capture. The general learning process of humanbeings is that they use inductive methods to cluster the information in the brain by finding similarities and dissimilarities of instances. Through this clustering approach, concepts are discovered and words' meanings (senses) are found out. For instance, the inductive process of human's brain puts the word '*bank*' in two clusters due to having two different meanings, which are '*the financial place*' and '*near the river*'. Artificial intelligence and natural language processing aim at simulating this ability of human on a machine to learn a natural language. Word Sense Induction (WSI) is a task that causes a machine to induce word senses automatically from a raw data without using a prior knowledge or annotated data.

Providing knowledge sources of words' senses for machines, such as the WordNet [2] and OntoNotes [3], is normally done manually. There are drawbacks for this approach. Providing such data is expensive in terms of both time and cost and due to the language change over time, updating the lexical resource and revising it require additional cost and time. The number of senses of an existing word is not consistent; therefore, a new sense may be added to an existing word, or the senses of an existing word may become outdated. Moreover, senses may change with respect to the domain. To overcome the drawbacks, WSI can be a solution to provide helpful information for different tasks, such as machine translation and information retrieval.

Context clustering is one of the well-known successful approaches for WSI [4, 5, 6]. One challenge of this approach is choosing the appropriate clustering algorithm. The proposed models in the literature have benefited two important clustering approaches, namely parametric and non-parametric. Parametric clustering algorithms, such as partitioning K-means algorithm, require a predefined fixed number of clusters as an input parameter; while non-parametric clustering algorithms, such as Chinese Restaurant Process (CRP) [7] and Density-Based Spatial Clustering of Applications with

* Corresponding Author

Noise [8], make decision to define new clusters. Although non-parametric clustering algorithms are more suitable for inducing word senses, the reported results in the literature show the superiority of parametric clustering in the field. It has to bear in mind that parametric clustering suffers from the problem of requiring a predefined number of clusters. In this research, we aim at addressing this problem by capturing the optimum number of clusters using a density measure. To reach the goal, we use the parametric clustering approach for the WSI task and try to solve its major problem by utilizing the internal evaluation score defined for measuring the density of clusters.

Another challenge of the context clustering approach is the method to be used to capture the contextual information of words to be able to decide about their senses and to achieve accurate results. Distributional semantic representation of words, such as Word2Vec [9], has achieved promising results in the area of natural language processing, such as syntactic parsing [10, 11], named entity recognition [12], sentiment analysis [13], and WSI [5, 6, 14]. Following the successful history of distributional semantic representation in the tasks, we also benefit from this approach to capture the information of the surrounding words in a context to induce word senses.

The structure of this paper is as follows: in Section 2, the distributional semantic representation of words and contexts is described. In Section 3, the previous studies on context clustering for WSI and the clustering algorithms used for this task are discussed. Section 4 explains our proposed models. The experimental results are reported and discussed in Section 5. Finally, Section 6 concludes the paper.

2- Distributional Semantic Representation

Distributional semantics is based upon the "distributional hypothesis". The distributional hypothesis roots at the idea of Harris [15] such that the words that occur in the same context tend to have similar meanings. Harris believes that the meaning of a word is reflected from the context that the word is used. This idea resembles the idea proposed by Wittgenstein [16] who says, "the meaning of words lies in their use". Firth [17] adds that "[y]ou shall know a word by the company it keeps". These ideas indicate that using the contextual information plays a very important role in determining the meaning of a word. As a result, Miller and Charles [16] have proposed a strong contextual hypothesis that expresses "two words are semantically similar to the extent that their contextual representations are similar". Based on their idea, Examples (1) to (3) nicely show that the words '*car*', '*automobile*', and '*auto*' are (relatively) semantically similar due to have (relatively) similar contexts.

(1) He parked his new car in the parking lot.
(2) He parked his new automobile in the parking lot.
(3) He parked his new auto in the parking lot.

To represent the contextual information of the distributional semantics, two general approaches are used [14]: (a) Bayesian methods using topic modeling approaches; and (b) feature-based methods using the vector representation of the contextual information. While topic modeling approaches, such as Latent Dirichlet Allocation [19] and Hierarchical Dirichlet Process [20], represented successful results in WSI, the flexibility of vector space models has received researchers' attention to capture multiple senses of words in the WSI framework.

The vector space model exploited in information retrieval [21] has a crucial contribution to distributional semantics to represent information of a word and its context. In other words, compressing the information about the words and their contexts in vectors explores the semantic distribution of the words. In the literature, this way of encoding and representing word information is known as 'word embedding' [9]. Computing the geometric distance between the vectors results in the similarity between the words. In Examples (1) to (3), the distance between the vectors of the words '*car*', '*automobile*', and '*auto*' is measured low; therefore, these words are assumed to have a similar meaning. There are several similarity measures to compute the vector distance, such as the Euclidean distance, the Cosine similarity, the Jaccard measure, and the Dice measure [22].

Precise coding of the word's contextual information has a direct impact on the quality of finding the most similar words. Since the context plays a very import role, Peirsman and Geeraerts [23] introduced three types of linguistic contexts: (a) document-based model: the words which are used in the same paragraph or in the same documents are similar [24, 25]; (b) syntax-based model: words are compared according to their syntactic relations, more precisely using the dependency relations [26, 27, 28, 10], or the combinatory categorial grammar [29]; and (c) word-based model: words are modeled based on their word-word co-occurrence within a window size. These word co-occurrences resemble the 'bag-of-words' model [25].

In recent studies, the word embedding approach has been taken into the consideration to build the words' vectors. The promising results that this approach obtained caused researchers to propose different techniques to achieve high quality vectors. As a result, two different approaches have been widely studied recently to model the contextual information: (a) using the matrix decomposition techniques, and (b) using the neural network-based techniques. GLObal VEctor representation (GloVe) [30] is an unsupervised learning method that follows the former approach to provide the distributional representation of

words. Continuous Skip gram (Skip-gram) and Continuous Bag Of Words (CBOW) models [9] use the latter approach to represent the contextual information of a word in a vector. Various toolkits are developed based on these approaches, such as the Word2Vec toolkit developed by Mikolov et al. [9]. In this paper, we use the Gensim library in Python [1] to create words' vectors in our model. To capture the context of each word for clustering, we propose two modes within our model. We use the whole sentence and extract the required information of the target word from the sentence, thereafter called the SentContext mode. Additionally, we limit the local context of the target word to the surrounding words and extract the contextual information of the target word with respect to the neighboring words, thereafter called the WinContext mode.

3- Studies on Context Clustering for WSI

The main focus of this paper is on the WSI task that is performed by context clustering to distinguish senses of the target polysemous word. In this approach, each cluster determines a sense of the target word.

Huang et al. [4] calculated TF-IDF[2] of each word and used it as a weighting value in the vectors of each word. The K-means algorithm was used to cluster the weighted words' contexts.

Neelakantan et al. [5] predicted each sense of a word as a context cluster assignment. To this end, they used the K-means algorithm in their model, such that a fixed number of clusters, namely 3 clusters, was defined to run the clustering algorithm.

Li and Jurafsky [6], however, proposed using CRP [7] as a non-parametric model to capture the senses dynamically. In their approach, the model decides either to generate a new sense for each context or to assign the context to an already generated sense.

Wang et al. [31] proposed a model to use weighted topic modeling for sense induction.

Amrami and Goldberg [32] extended a bidirectional recurrent neural network model proposed by Peters et al. [33] and used predicted word probabilities in the language model of their induction model.

Alagic et al. [34] proposed the idea that words belonging to a cluster should be able to be substituted in an appropriate context. Based on this idea, they implemented a model to induce word senses.

Correa and Amancio [35] used the complex network proposed by Contucci et al. [36] for context embedding and proposed a model to capture the structural relationship among contexts.

A large number of researches in this field use context clustering to address the problem. Both parametric and non-parametric methods have been studied in this field. The proposed models by Huang et al. [4], Neelakantan et al. [5], and Amrami and Goldberg [32] are the examples of parametric clustering; while the proposed model by Li and Jurafsky [6] is an example of non-parametric clustering. The main advantage of parametric clustering is that they can work with high data dimensionality; but its main disadvantage, as discussed in Section 1, is requiring a fixed number of clusters, which does not seem to meet the requirements of the WSI task. The advantage of non-parametric methods is that they do not require a fixed number of clusters; but the disadvantage of these methods is their poor performance to make a decision in order to assign a word to a new cluster.

As reported by Song et al. [14], a comparative study on parametric and non-parametric models on the SemEval2010 WSI task [37] shows that the K-means parametric model outperforms the CRP algorithm proposed by Li and Jurafsky [6]. As stated in Song et al. [14], the main reason for obtaining such results is the poor performance of CRP in making a decision to assign a word to a new cluster. While the best average number of clusters in the SemEval2010 gold data for the WSI task for both noun and verb categories is 5.04 clusters (senses), in the study of Neelakantan et al. [5] the K-means algorithm used 3 clusters as the fixed number of clusters and CRP ended to a lesser number of clusters on average. This result indicates that relaxing the pre-defined number of clusters in K-means can further improve the performance of the task.

4- Density Measure as a Clustering Criteria

The K-means algorithm [38] is one of the most popular unsupervised learning algorithms to be used in various tasks. This clustering algorithm works based on the similarity within the objects of a cluster, and the dissimilarities between the objects of different clusters. To make the decision about the similarities, the distance between the objects is approximated. To find the best number of clusters, we need to evaluate the results of the clustering algorithm. The clustering result can be measured externally or internally. In the former validation, gold data is required; while in the latter validation no gold data is required and it is done in an unsupervised fashion. In our case, since we have no access to gold data in the real application runtime, the latter validation has to be used.

Liu et al. [39] introduced five aspects that have impact on selecting appropriate internal validation measures of clustering: (a) the monotonicity of different internal validation indices, (b) the impact of noise, (c) the density

[1] https://radimrehurek.com/gensim/models/word2vec.html
[2] Term Frequency-Inverse Document Frequency

of clusters, (d) the clusters that are closed to each other (sub-clusters), and (e) the skewed distribution of data in clusters. In the WSI task, the most important aspect that should be taken into the consideration for capturing the number of clusters is the 'density'. The noise and the skewed distribution are also relevant aspects for this task, but we do not study them in this paper and leave them for further studies.

The silhouette coefficient score [40], also called silhouette index, is one of the well-known metrics that scales the validity of the clustering result and makes a distinction between the clearly defined clusters and the vague ones. Equation (1) computes the silhouette coefficient score of the instance *i* (*s*(*i*)):

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (1)$$

where *a*(*i*) is the average distance between instance *i* and all instances that are in the same cluster as *i*, and *b*(*i*) is the minimum average distance between instance *i* and all instances in clusters of which *i* is not a member. The value of the silhouette score is between 1 and -1. If the score is closer to 1, it determines that there is high density within the objects of a cluster, and the objects are well clustered. If the score is closer to -1, this indicates that the objects in the cluster are not classified well.

Instead of computing the silhouette score for each individual object of a cluster, the average silhouette score is computed; therefore, the higher this average score, the higher density of the objects in a cluster and well-clustering of the data.

The silhouette coefficient score can provide a way to assess parameters to choose the optimum number of clusters. This property of the score is very relevant and useful for WSI. Using the average silhouette score makes it possible to have a better approximation on the number of clusters; i.e., for each number of clusters, we calculate silhouette and select the cluster number which results the highest value of silhouette. In our research, we use the silhouette coefficient score to measure the density of clusters for the WSI task.

Algorithm (1) shows the pseudo-code of our proposed model. As can be seen in the code, four data sources as the input are required: (a) list of target words to induce their senses; (b) a raw corpus; (c) a collection of unlabeled data containing the target words; and (d) a set of annotated data where the target word is labeled in terms of its sense in the local context. In the first step of our algorithm, Word2Vec is used for representing words in the vector space model. In the next step, the local context of the target words (either at the sentence level or limited to the surrounding words) are extracted. Then, the context vectors of the target words are built from the unlabeled and labeled data based on weighted average word vector.

Algorithm (1): Density-based our proposed WSI model

Input: Target word TW
Raw text corpus *C* for embedding
Set of unannotated data *UD* containing *TW*
Set of annotated data *AD* containing *TW* for evaluation
Step1: Create words' vectors V of concatenated data (*C* + *UD* + *AD*)
Step2: Extract context (either sentence or window) of *TW* from *UD* and *AD*
Step3: Compute TF-IDF of context words to be used as weights
Step4: Create context vector for each instances of *UD* and *AD* based on weighted average word vector
for *i* = 2 **to** *i* = *t* **do**
  Step5: Run K-means clustering algorithm on the created context vectors of *TW*
  Step6: Calculate average silhouette score *Si*
  if *i* > 2 and $\mathbf{S_i < S_{i-1}}$ **then**
    Step7: Stop and select $i-1$ as the optimum cluster number
  end if
end for
if no optimum cluster number is selected **then**
Step8: select *t* as the optimum cluster number
end if

In the next step, the K-means algorithm is run on the created context vectors of a target word. K-means requires a predefined number of clusters. This number varies 2 to 20 and in each experiment the average silhouette score is calculated. As far as the silhouette score is increasing, the number of clusters increases as well. The algorithm is stopped as soon as the silhouette score decreases. In the case of finding no optimum number of clusters, it halts by reaching the upper bound of the loop.

This process results in the time complexity of O(*nkit*), where *n* is the number of instances containing the target word, *k* is the number of clusters, *i* is the number of iterations, and *t* is the number of trials to find the optimum number of clusters which starts from 2 and continues to 20 in the worst case. It should be mentioned that the state-of-the-art models, namely CRP and Kmeans-3, have the time complexity of O(n^2) and O(*nki*), respectively.

5- Experimental Result

5-1- Data Set

To run our experiments, we require three data sets: the labeled data to be used for evaluating the clustering

performance of each target word, the unlabeled data set to be used for clustering each target word, and the data pool for creating the vector representation of the words. To evaluate the clustering results and to create the clusters containing the target words in their contexts, we use the SemEval2010 data set for the WSI task [37] that is mostly from the news domain. In total, 100 words (50 verbs and 50 nouns) are the target words in this data set. This data set contains 8,915 instances as test data with sense annotation and 888,722 unannotated sentences as training data. In the evaluation, two evaluation metrics, namely V-measure and F-measure, are used. These two metrics are explained in Section 5.3.

The data that we use for creating word vectors is The Westbury Lab Wikipedia Corpus developed by Shaoul and Westbury [41]. This corpus that is freely available is collected from the dump of English Wikipedia articles in April 2010. The corpus contains almost 990 million word tokens of the general domain and it has been used for similar tasks as reported in the literature [4, 5]. It should be mentioned that the documents with less than 2000 characters long are excluded from the corpus.

5-2- Baselines

In SemEval2010 [37] three baselines are introduced: (a) the Most Frequent Sense (MFS): in this baseline, all instances are assigned to a single cluster which contains the most frequent sense; (b) one instance per cluster, thereafter named 1S1C: in this baseline, each instance is assigned to a separate cluster; therefore the number of clusters is equal to the number of instances; and (c) random baseline where an instance is randomly assigned to a cluster. The randomization can be done more than once, so that the average result is considered as the final result. In experiments, randomization has been done five times and the average result of the experiments is reported.

Moreover, we use three state-of-the-art models, namely the CRP model proposed by Li and Jurafsky [6], the K-means-3 cluster which assumes 3 senses for each word proposed by Neelakantan et al. [5], and the SemEval2010 Average Participants that is the average system performance of the 26 groups participated in the SemEval2010 WSI task. Furthermore, the state-of-the-art models are considered as additional baselines to compare with the clustering performance of our proposed model.

5-3- Evaluation Metrics

To evaluate the accuracy of the clustering performance, various metrics are proposed. VanRijsbergen [42] proposed F-measure as a metric for evaluating external clustering. Dom [43] and Meila [44] proposed using an entropy-based approach to evaluate how good the clustering result is. Additionally, Rosenberg and Hirschberg [45] proposed V-measure as another entropy-based approach. This metric is a harmonic mean of evaluating both internal and external clustering. Among the metrics in the literature, F-measure and V-measure are frequently used which are explained in more detail.

F-measure proposed by VanRijsbergen [42] is the metric for computing the accuracy of information retrieval as in Equation (2):

$$F - measure = \frac{(1+\beta) \times P \times R}{(\beta \times P) + R} \tag{2}$$

where P is precision, R is recall, and β is a weighting parameter. If $\beta > 1$, more weight is assigned to recall, and in case $\beta < 1$, more weight is assigned to precision. If $\beta = 1$, precision and recall are considered equally. Equations (3) and (4) compute precision and recall, respectively. In all equations, K is the CLUSTER set, which is the hypothesized clusters from the clustering output and C is the CLASS set, which is the correct partitioning of the data; i.e., for a target dataset with N elements, we have two partitions: the guess partition K, and the gold partition C.

$$P = \frac{n_{ij}}{|k_i|} \tag{3}$$

$$R = \frac{n_{ij}}{|c_i|} \tag{4}$$

where n_{ij} is the number of members of class $ci \in C$ that is the element of cluster $kj \in K$.

V-measure computes the harmonic mean of homogeneity, h, and completeness, c, of clustering to capture the clustering success as computed in Equation (5):

$$V - measure = \frac{(1+\beta) \times h \times c}{(\beta \times h) + c} \tag{5}$$

Homogeneity means that in each CLUSTER, there are a few numbers of CLASSes. The best mode of homogeneity is when a cluster consists of only samples of one class. Completeness, which is the reverse of homogeneity, means that each CLASS is appeared in a few numbers of CLUSTERs. The best mode of completeness is when all samples of the same class are within a single cluster.

As Rosenberg and Hirschberg [45] explain, homogeneity and completeness are formally defined as:

(6)

$$h = \begin{cases} 1 & if \quad H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & else \end{cases}$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N} log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|} a_{ck}}{N} log \frac{\sum_{k=1}^{|K|} a_{ck}}{N}$$

(7)

$$c = \begin{cases} 1 & \text{if } H(K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases}$$

where

$$H(K|C) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N} log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_{ck}}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|} a_{ck}}{N} log \frac{\sum_{c=1}^{|C|} a_{ck}}{N}$$

$C=\{c_i \mid i = 1, \ldots, n\}$ is the set of CLASS, $K = \{k_i \mid 1, \ldots, m\}$ is the set of CLUSTER, and N is the number of data points in the data set, and a_{ck} is the number of elements of class c in cluster k.

The advantage of V-measure over F-measure is that in the evaluation, both homogeneity and completeness are taken into the consideration, while in F-measure, only the distribution of classes in clusters, i.e., homogeneity, is considered and it does not care about whether in each cluster the number of classes are minimized. This difference indicates that V-measure is more reliable than F-measure and it accurately evaluates the performance of the clustering result.

5-4- Setup of Experiments

As mentioned, we use the K-means clustering algorithm for our experiments. This clustering algorithm requires vector representation of the data; consequently, we use the Gensim Python library to create the vectors of the words. The setup of creating the words' vectors and to make the vectors as distinct as possible are as follows: (a) the Skip-gram model is employed for building vectors to better capture the context; (b) to use the Skip-gram model, similar to Huang et al. [4], the information of the local context containing 8 words, 4 words before and 4 words after the target word, is extracted; (c) similar to Neelakantan et al. [5], the dimension of each vector is set to 300; and (d) words with frequency 5 and above are kept to build the vectors. The vector of the rest words is considered zero. In the next step, the context vectors are created as weighted average of words' vectors. Based on the idea proposed by Huang et al. [4], we calculate TF-IDF of each word and use it as a weighting value for each vector to compute the context vector.

In our proposed models, two modes of input data are provided for the K-means algorithm. In the first mode, the 'SentContext' mode, the weighted vectors of the words contained in the sentence are summed up to build the sentence vector of the target word, and then the score is normalized based on the sentence length. In the second mode, the 'WinContext' mode, we use the limited context of the target word, 4 words before and 4 words after the target word, to build the sentence vector. The reason to limit the context in this mode to the 8 words is to be similar to the number of context words used for building the words' vector.

The sketch of our proposed model was described in the pseudo-code of Algorithm (1). In our experiments, the data is clustered with different fixed cluster numbers in the K-means algorithm such that the clusters' number vary from 2 to 20 for both noun and verb categories. We perform clustering for each cluster number, starting from 2. Then for each cluster number, we compute the silhouette coefficient score to measure the density of clusters. As long as the silhouette coefficient score increases, the algorithm adds up to the number of clusters, and the clustering task is reperformed. This task is repeated, and as soon as the silhouette coefficient score is decreased, the clustering process stops, and the optimum cluster number with the highest silhouette score is selected as the best number of clusters.

5-5- Results and Discussion

The performance of our model is evaluated based on the external and internal evaluation methods. In Table (1), the summary of the obtained results of the external evaluation method including the baseline models, the state-of-the-art models, and the two modes of our proposed model are reported.

According to the results, the 1S1C baseline outperformed all of the models according to V-measure and neither modes of our model nor the state-of-the-art models were able to beat this baseline. In contrast, this baseline obtained the lowest score according to F-measure. Moreover, the MFS baseline performed the worst according to V-measure and the best according to F-

measure. These two baselines can be considered as a spectrum such that the MFS baseline (homogeneity) is the worst point and the 1S1C baseline (completeness) is the best point. This indicates that there is a trade-off between the two metrics, and the ultimate goal is to achieve a result that moves towards the reasonable value in both metrics.

Table (1): Results of the baselines, the state-of-the-art models, and our proposed model

	Model	***V-Measure(%)***			***F-measure(%)***		
		all	***noun***	***verb***	***all***	***noun***	***verb***
BASELINE	Random Sampling	4.4	4.2	4.6	31.9	30.4	34.1
	1S1C	**31.70**	35.8	25.6	0.09	0.11	0.08
	MFS	0	0	0	**63.40**	57	72.7
STATE-OF-THE-ART	CRP	5.7	7.4	3.2	**55.3**	49.4	63.8
	Kmeans-3	**9.8**	13.5	4.3	55.1	50.7	61.6
	SemEval2010 Average Participants	6.63	7.08	5.95	8.85	9.44	10.83
PROPOSED MODEL	SentContext	**14.28**	16.6	10.9	**44.59**	42.45	47.7
	WinContext	**15.19**	16.7	13	39.57	36.76	43.66

The random baseline that is closer to the real application has obtained better results than the MFS baseline based on V-measure, and better than the 1S1C baseline based on F-measure. As can be seen in the table, our proposed model and three of the state-of-the-art models have beaten this baseline according to the V- and F-measure for both noun and verb categories.

Based on the reported results, the Kmeans-3 model performed the best among the state-of-the-art models according to V-measure for both categories.

Comparing our proposed model with the baselines for both categories, the two modes of our model outperformed the random sampling model based on both V- and F-measure metrics. We further compared the two modes of our model with themselves. In general, for both categories, the WinContext mode obtained a better performance than the SentContext mode according to V-measure. The performance of the WinContext mode for the verb category is 2.1% better than the SentContext mode; while for the noun category, the performance of the two modes are relatively similar. This achievement determines that considering the local context of verbs can identify the meaning of the word; while for the nouns a wider context might be required which varies from one word to another.

Comparing the two modes of our proposed model with the state-of-the-art models, we observed that our proposed model outperformed the state-of-the-art models according to V-measure for both noun and verb categories, while the F-measure is kept in a reasonable range. Our model has beaten the SemEval2010 Average Participants baseline according to F-measure as well as V-measure for both categories. According to the results, we can conclude that the density within the objects of a cluster has a direct impact on homogeneity and completeness in the V-measure metric; consequently it causes to increase the accuracy of the clustering result. The clustering density property has no impact on the F-measure evaluation metric as seen in the results of the state-of-the-art models that achieved a higher F-measure score than our proposed model.

Additionally, the internal evaluation of the clustering algorithm is performed and the average silhouette score of the selected clusters of the target words is computed. The detailed results of each target word in the gold data for both categories are reported in Table (2).

We further compared the output of our proposed model with the SemEval2010 gold standard data in terms of number of identified senses, reported in Table (3). The average number of clusters and the maximum number of clusters in our proposed model have relatively obtained the expected results in the gold data for both categories and our model captures the number of senses relatively accurate, while in the proposed model by Neelakantan et al. [5] the number of senses is set to 3 which is not accurate compared to the gold data. As seen in this table, the density of clusters in both noun and verb categories of the SentContext mode is higher than the WinContext mode. In addition, the density of the clusters for the verb category in both modes is higher than the density in the noun category. One reason for this is that verbs have smaller number of senses; therefore the clusters are denser than nouns that have larger number of senses.

6- Conclusion

In this paper, we used the K-means clustering algorithm, as a parametric clustering algorithm, for the WSI task. Due to the nature of the parametric clustering algorithm, the number of clusters should be predefined that is not possible for the WSI task. To tackle the problem, we proposed a model that uses the density of the clustering algorithm to identify words' senses. To build the model, word embedding with Skip-gram is utilized for this task. In our experiments, we used the silhouette coefficient score to measure density of clusters and estimate the best number of clusters. The experimental results of the external evaluation metric showed that our proposed model has beaten the state-of-the-art models. The obtained results determined that the high density within the objects of a cluster has a direct impact on well-clustering of objects. Moreover, the average number of clusters and the maximum number of clusters in the output of our proposed model are relatively close to the gold data.

Table 2: Number of clusters of the `NOUN' and `VERB' categories in the Gold Data (*GD*), the output of the SentContext (*SC*) and WinContext (*WC*) modes along with their corresponding Silhouette score (*S*)

NOUN						***VERB***					
Word	***GD***	***SC***	S_{SC}	***WC***	S_{WC}	***Word***	***GD***	***SC***	S_{SC}	***WC***	S_{WC}
access	8	5	0.044	5	0.032	accommodate	3	6	0.067	7	0.033
accounting	5	9	0.062	5	0.037	analyze	2	3	0.074	4	0.047
address	5	6	0.071	7	0.054	appeal	4	7	0.062	7	0.027
air	11	5	0.073	4	0.052	apply	4	6	0.077	6	0.044
body	14	4	0.084	6	0.029	assemble	2	5	0.066	6	0.03
camp	7	6	0.056	8	0.024	assert	3	5	0.071	6	0.04
campaign	4	4	0.064	5	0.039	bow	5	5	0.063	5	0.067
cell	6	3	0.157	5	0.06	cheat	2	4	0.077	3	0.078
challenge	10	3	0.076	3	0.053	commit	3	4	0.091	3	0.089
chip	5	4	0.108	4	0.082	conclude	4	4	0.083	7	0.038
class	6	5	0.041	6	0.024	cultivate	4	6	0.069	4	0.064
commission	8	5	0.063	4	0.039	defend	2	3	0.139	3	0.074
community	7	10	0.041	7	0.025	deny	3	4	0.081	9	0.04
dealer	7	4	0.093	5	0.051	deploy	2	8	0.07	8	0.049
display	5	4	0.075	11	0.022	divide	5	4	0.064	5	0.033
edge	10	3	0.082	6	0.028	expose	2	4	0.072	6	0.049
entry	8	4	0.073	5	0.03	figure	5	6	0.041	7	0.033
failure	7	13	0.077	5	0.056	frame	4	3	0.132	7	0.059
field	6	6	0.071	6	0.048	happen	4	9	0.026	4	0.024
flight	7	4	0.068	9	0.019	haunt	2	9	0.034	5	0.049
foundation	3	8	0.064	7	0.037	insist	2	5	0.062	6	0.02
function	6	3	0.136	6	0.037	introduce	3	5	0.082	6	0.033
gap	7	8	0.057	4	0.05	lay	6	7	0.087	9	0.042
gas	6	6	0.064	5	0.042	level	4	5	0.059	3	0.104
guarantee	10	9	0.052	10	0.032	lie	4	3	0.083	4	0.063
house	13	7	0.051	5	0.038	mount	5	5	0.097	4	0.066
idea	6	7	0.033	10	-0.002	observe	4	4	0.091	7	0.055
innovation	5	12	0.019	4	0.044	operate	2	5	0.05	6	0.032
legislation	4	4	0.04	5	0.024	owe	3	5	0.086	5	0.054
margin	7	7	0.054	7	0.048	pour	4	4	0.156	6	0.066
mark	5	7	0.094	4	0.031	presume	2	7	0.071	4	0.102
market	4	7	0.028	9	0.009	pursue	2	6	0.067	5	0.043
mind	8	5	0.062	6	0.035	question	2	6	0.056	8	0.036
moment	9	6	0.076	7	0.047	reap	2	6	0.095	10	0.129
movement	7	3	0.122	3	0.045	regain	2	4	0.093	4	0.049
note	6	4	0.083	3	0.1	relax	3	3	0.115	4	0.064
office	6	5	0.072	4	0.046	reveal	2	3	0.083	4	0.04
officer	8	3	0.092	10	0.013	root	4	6	0.072	4	0.077
origin	5	4	0.1	6	0.023	separate	2	7	0.086	3	0.072
park	9	6	0.059	7	0.024	shave	2	7	0.055	6	0.059
promotion	5	4	0.084	5	0.057	signal	2	5	0.076	3	0.086
rally	7	6	0.061	5	0.024	slow	2	4	0.106	6	0.019
reputation	11	5	0.089	5	0.047	sniff	3	7	0.063	4	0.089
road	5	4	0.092	4	0.042	stick	4	7	0.042	5	0.056
screen	9	5	0.065	5	0.035	straighten	3	7	0.096	5	0.076
shape	7	4	0.089	6	0.044	swear	5	6	0.054	4	0.067
speed	4	6	0.072	4	0.056	swim	2	5	0.065	3	0.09
television	4	8	0.05	10	0.038	violate	2	4	0.095	4	0.077
threat	8	6	0.027	7	0.026	wait	2	3	0.097	4	0.061
tour	8	6	0.035	8	0.022	weigh	6	5	0.089	5	0.077

Table 3: Comparing the Average Number of Clusters (*ANC*) and Maximum Number of Clusters (*MNC*) along with the Average Silhouette Score (*ASS*) in gold data and our proposed model for both noun and verb categories

Model	***ANC***		***MNC***		***ASS***	
	Noun	***Verb***	***Noun***	***Verb***	***Noun***	***Verb***
SemEval2010 gold-data	6.96	3.12	14	6	-	-
SentContext	5.64	5.22	13	6	0.07	0.078
WinContext	6.03	5.26	11	6	0.038	0.057

References

[1] F. de Saussure, *Cours de linguistique générale*, C. Bally, A. Sechehaye, and A. Riedlinger, Eds. Lausanne, Paris: Payot, 1916.

[2] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, Vol. 38, No. 11, 1995, pp. 39-41.

[3] E. Hovey, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: The 90% solutions," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 57-60.

[4] E. Huang, R. Socher, C. D. Manning, and A. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju Island, Korea: Association for Computational Linguistics, 2012, Vol. 1, pp. 837-882.

[5] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," in *Proceedings of the Conference on Empirical Methods in Natural Language.* Doha, Qatar: Association for Computational Linguistics, 2014.

[6] J. Li, and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?" in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1722-1732.

[7] D. M. Blei, M. I. Jordan, T. L. Griffiths, and J. B. Tenenbaum, "Hierarchical topic models and the nested Chines restaurant process," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*. MIT Press, 2003, pp. 17-24.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial database with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. Han, and U. M. Fayyad, Eds. AAAI Press, 1996, pp. 226-231.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z . Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111-3119.

[10] O. Levy, and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Vol. 2. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 302-308.

[11] I. Beltagy, K. Erk, and R. Mooney, "Semantic Parsing using distributional semantics and probabilistic logic," in *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, Association for Computational Linguistics, 2014, pp. 7-11.

[12] A. Das, D. Ganguly, and U. Garain, "Named entity recognition with word embeddings and Wikipedia categories for a low-resource language," *ACM Transaction on Asian Low-Resource Language Information Processing*, Vol. 16, No. 3, 2017, pp. 1-19.

[13] L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Transaction on Audio, Speech and Language Processing*, Vol. 26, No. 3, 2018, pp. 671-681.

[14] L. Song, Z. Wang, H. Mi, and D. Gildea, "Sense embedding learning for word sense induction," in *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics*. The *SEM 2016 Organizing Committee, 2016, pp. 85-90.

[15] Z. S. Harris, "Distributional structure," *Word*, Vol. 23, No. 10, 1954, pp. 146-162.

[16] L. Wittgenstein, *Philosophical Investigations*, Oxford, UK: Blackwell Publishing Ltd, 1953.

[17] J. R. Firth, "A synopsis of linguistic theory 1930-1955," *Studies in Linguistic Analysis (Special Volume of the Philosophical Society)*, 1957, pp. 1-32.

[18] G. A. Miller, and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processing*, Vol. 6, No. 1, 1991, pp. 1-28.

[19] D. M. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 993-1022.

[20] Y. W. The, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet process," *Journal of the American Statistical Association*, Vol. 101, No. 476, 2006, pp. 1566-1581.

[21] G. M. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613-620.

[22] D. Jurafsky, and J. H. Martin, *Speech and Language Proeccing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2018, https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[23] Y. Peirsman, and D. Geeraerts, "Predicting strong associations on the basis of corpus data," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 648-656.

[24] T. K. Landauer, and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological Review*, Vol. 104, No. 2, 1997, pp. 211-240.

[25] M. Sahlgren, *The Word-space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces*, Ph.D. dissertation, Stockholm University, Stockholm, Sweden, 2006.

[26] Z. S. Harris, *A Theory of Language and Information: A Mathematical Approach*, Oxford, England: Oxford University Press, 1991.

[27] D. Lin, "Automatic retrieval and clustering of similar words," in *Proceedings of the 17th International Conference on Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 1998, Vol. 2, pp. 768-774.

[28] S. Padó, and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, Vol. 33, No. 2, 2007, pp. 161-199.

[29] K. M. Hermann, and P. Blunsom, "The role of syntax in vector space models of compositional semantics," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, Vol. 1, pp. 894-904.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, Vol. 14, pp. 1532-1543.

[31] J. Wang, M. Bansal, K. Gimpel, B. D. Ziebart, and C. T. Yu, "A sense-topic model for word sense induction with unsupervised data enrichment," *Transaction of the Association for Computational Linguistics*, Vol. 3, 2015, pp. 59-71.

[32] A. Amrami, and Y. Goldberg, "Word sense induction with neural biLM and symmetric pattern," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4860-4867.

[33] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227-2237.

[34] D. Alagić, J. Šnajder, and S. Padó, "Leveraging lexical substitutes for unsupervised word sense induction," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018.

[35] E. A. Corrêa, and D.R. Amancio, "Word sense induction using word embeddings and community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, Vol. 523, 2019, pp. 180-190.

[36] B. Perozzi, R. Al-Rfou', V. Kulkarni, and S. Skiena, "Inducing language networks from continuous space word representations," in *Complex Networks*, P. Contucci, R. Menezes, A. Omicini, and J. Poncela-Casasnovas, Eds. Cham: Springer International Publishing, 2014, pp. 261-273.

[37] S. Manandhar, I P. Klapaftis, D. Dligach, and S. S. Pradhan, "SemEval-2010 task 14: Word sense induction & disambiguation," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp 63-68.

[38] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkley, California: University of California Press, 1967, Vol. 1, pp. 281-297.

[39] Y. Lie, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proceedings of the 2010 IEEE International Conference on Data Mining*. Washington, D.C., USA: IEEE Computer Society, 2010, pp 911-916.

[40] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, Vol. 20, No. 1, 1987, pp 53-65.

[41] C. Shaoul, and C. Westbury, "The Westbury Lab Wikipedia Corpus," 2010, http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[42] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[43] B. E. Dom, "An information-theoretic external cluster-validity measure," IBM, Technical Report, 2001.

[44] M. Melia, "Comparing clusterings – an information based distance," *Journal of Multivariate Analysis*, Vol. 98, No. 5, 2007, pp. 873-8995.

[45]A. Rosenberg, and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp 410-420.

Masood Ghayoomi received his PhD degree in Computational Linguistics from Berlin Freie University, Berlin, Germany in 2014, and M.S. degree in Computational Linguistics from Nancy2 University, Nancy, France and Saarland University, Saarbrücken, Germany, in 2009. Currently he is a faculty member at the Institute for Humanities and Cultural Studies. His research interests include Computational Linguistics, Natural Language Processing, Machine Learning, Corpus Linguistics, Syntax and Lexical Semantics.