ShivKishan Dubey
Narendra Kohli

CLUSTERING FOR CLARITY: IMPROVING WORD SENSE DISAMBIGUATION THROUGH MULTILEVEL ANALYSIS

Abstract *Existing Word Sense Disambiguation (WSD) techniques have limits in exploring word-context relationships since they only deal with the effective use of word embedding, lexical-based information via WordNet, or the precision of clustering algorithms. In order to overcome this limitation, the study suggests a unique hybrid methodology that makes use of context embedding based on center-embedding in order to capture semantic subtleties and utilizing with a two-level k-means clustering algorithm. Such generated context embedding, producing centroids that serve as representative points for semantic information inside clusters. Additionally, go with such captured cluster- centres in the nested levels of clustering process, locate groups of linked context words and categorize them according to their word meanings that effectively manage polysemy/homonymy as well as detect minute differences in meaning. Our proposed approach offers a substantial improvement over traditional WSD methods by harnessing the power of center-embedding in context representation, enhancing the precision of clustering and ultimately overcoming existing limitations in context-based sense disambiguation.*

Keywords embeddings, center embedding, multilevel clustering, word sense disambiguation (WSD), polysemy and homonymy

Citation Computer Science 25(2) 2024: 277–300

1. Introduction

Word Sense Disambiguation (WSD) is a fundamental task in Natural Language Processing (NLP) that attempts to determine the correct meaning of a given word given its context [19]. Related studies have led to the creation of algorithms/methodologies that use a range of resources like knowledge based that employ various relations in terms of is-a/part-of, and corpus-based that having with sense-tagged information. However, the materials needed for these methods must be created by hand by people, making them costly to acquire and maintain. Distributional techniques are best alternative to this, which distinguish words according to their meanings on the grounds that words that appear in similar instances will also have comparable meanings. Further, even while most VSMs are helpful, they all have the drawback of having just one vector for each word, which blatantly misses the mark when it comes to polysemy [8]. In accordance to this, multi-prototype VSM are with hybrid methodologies introduced with WSD tasks.

Polysemy is the phenomenon in which a singular term has multiple meanings that are interrelated. For instance, *"bank"* can refer to both a *financial institution* and a *riverbank*. Disambiguating between these various meanings is essential for accurate language comprehension and subsequent NLP applications. Homonymy, on the other hand, occurs when words with diverse meanings share the same form. For example, *"bat"* can refer to both a *winged mammal* and *an item of sporting equipment*. Correctly resolving homonyms is essential for preventing misinterpretation and ensuring precise semantic comprehension. Following Figure 1, clearly described about density of clusters for word bank which reflects about its close connections in itself.

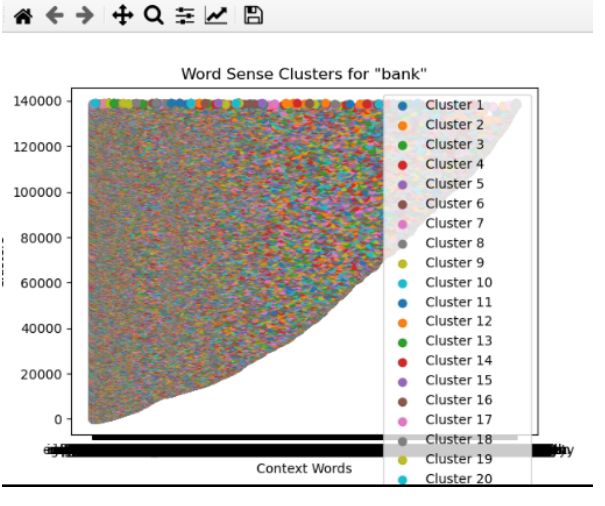


Figure 1. Cluster representation observed for ambiguous word *"bank"*. This represents word *"bank"* is how much overlapped and closed clusters in corpus due to its share contexts/senses

This figure clearly indicated that high degree of polysemy and the challenge of accurately disambiguating the word is reflected due to numerous senses or contexts associated with this ambiguous word *"bank"* and Close points indicated that the contexts associated with the word are similar or have overlapping characteristics due to the presence of related senses or the contextual similarities in the corpus.

Therefore, opted nested clustering [10] can provide a hierarchical structure to better organize and differentiate between different grouped contexts/senses, such contexts that represented as sentences even sometime complex sentences [7] which are hardly captured through traditional embeddings therefore to apply center embedding with context of left and right are very helpful to treat such grouped contexts. In this article, we propose with center-embedding based context embedding and utilize with nested level clustering (up to level 2) to address the challenges presented by polysemy and/or homonymy in WSD. Utilizing multilevel clustering permits us to generate clusters and sub-clusters that reflect the nuanced semantic relationships between words.

1.1. Provide the motivation

For Natural Language Processing (NLP) activities to advance, it is critical to address the shortcomings of conventional techniques to Word Sense Disambiguation (WSD). For precise language interpretation, information retrieval, machine translation, sentiment analysis, and other applications, NLP systems significantly rely on exact word sense disambiguation. However, these systems' efficiency and performance are hampered by the shortcomings of conventional techniques. Let's look at these restrictions in a Table 1 with some illustrations.

Table 1
Summerize view of limitations of traditional approaches

Limitations	Examples
Difficulty in capturing fine-grained distinctions among word senses	Consider the word *"run"*, which can have various senses such as *"to jog", "to manage", or "to operate"*. Traditional approaches often struggle to distinguish between such semantic differences, leading to incorrect interpretations [16]
Inability to handle high polysemy and homonymy	Traditional methods often fail to disambiguate words with multiple senses or different meanings. [20] e.g., the word *"crane"* can refer to *a bird* or *a large lifting machine*

To address these limitations, our proposed approach incorporates multilevel clustering which resolve such limitations (specially to capture polysemous, homonyms) in the proposed model of WSD:

Semantic Variations Captured: We find and group words that have similar semantic properties by using clustering methods to context word embeddings. This enables us to more properly capture the differences in word meanings within a certain context.

Handling Homonymy and Polysemy: Multilevel clustering makes it easier to recognize distinctive clusters and sub-clusters, allowing for a greater ability to distinguish between various meanings of polysemous and homonymous terms. By putting words in groups according to how they share a sense, it is easier to distinguish between terms like “bank” (financial organization vs. river bank) and ”bat” (flying animal vs. sports equipment).

Granularity: A more finely-grained representation of word senses is offered by the multilevel clustering technique. It enables the development of sub-clusters that can capture fine-grained semantic details inside a sense cluster. This granularity increases the accuracy of the disambiguation process and raises the overall effectiveness of NLP systems.

1.2. Our major contribution

The following are key objectives of our findings as:

- To develop a hybrid approach that makes use of multilevel clustering methods to increase the precision and level of word meaning disambiguation. This method will make it possible to interpret word meanings in various circumstances more precisely.
- We further address the difficulties presented by words having numerous meanings or senses (such contexts which represented in corpus with sentences even represent through complex sentences). Therefore, effectively capture such contexts through applying center-embedding based context embedding in multilevel clustering environment, and allowing to precisely comprehend and analyze their intended sense.
- To carry out in-depth analyses with detection of words having polysemy/homonyms and comparisons with current WSD techniques in order to gauge the performance and efficacy of our nested level clustering strategy.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work in the field of WSD. Section 3 presents the methodology and describes the multilevel center embedding approach in detail. Section 4 presents the experimental setup and the evaluation metrics used. Section 5 discusses the results and compares our approach with existing methods. Finally, Section 6 concludes the paper and discusses future directions for research in the field of enhanced word sense disambiguation.

2. Related previous works

Previous work in the area of word sense disambiguation (WSD) has looked into a number of strategies to increase the efficacy and accuracy of word sense disambiguation. Traditional approaches (in terms of graph-based, knowledge based and clustering based) have made a substantial contribution, but they frequently run into problems when dealing with polysemy and homonymy problems. Numerous research that are pertinent to our suggestion of adopting multilevel clustering for WSD have been done.

2.1. Knowledge based WSD approach

These approaches leverage the structured information within these resources to associate words with their appropriate sense and are very effective due to following reasons as rich semantic information,contextual disambiguation, transitivity of relations and no need for Sense-Tagged corpora. In terms of due to lack of contextual information and in cases of high polysemy or homonymy, where multiple senses of a word are plausible in a given context, knowledge-based methods may struggle to make precise disambiguation decisions.

In 2019, authors utilized genetic algorithms over wordnet and treated WSD task in Indo-aryan knowledge source specially for Gujrati language [21]. Later on, group of authors worked over SCSMM based approach [2] which combines sementic similarity, heuristics and document context to disambiguate terms.

2.2. Graph based WSD approach

Due to their ability to represent word meanings and context relationships in a structured manner. These approaches leverage the semantic relationships between words encoded in lexical knowledge bases like WordNet or other large-scale graphs built from text corpora. Some key reasons as word sense induction where senses are automatically induced from the graph's structure without relying on predefined sense inventories, easily represented relations of words and it's context in the form of nodes and edges that enables the model to consider the relatedness of different word senses and utilize this information to disambiguate the target word. Integration of external knowledge sources like WordNet, BabelNet, or domain-specific lexical resources that enhances the model's ability to access a rich pool of semantic knowledge and make informed disambiguation decisions.

As per above manners, group of authors utilizes random walks over large lexical knowledge base phenomenon using this graph based traditions [1] and got effective results during disambiguation task. To process over documents where authors [8] designed distributed graph based algorithms that cluster documents into same context and using heuristics vertex-centric algorithm inspired by metaphor of water cycle over WSD. In latest, graph based approach that proposed by authors [17] in which phenomenon utilized BFS to treated similarity and perform then WSD task very effectively.

Despite of effectiveness of graph based approach, words with multiple senses or ambiguous meanings can lead to densely connected graphs in traditional graph-based approaches, making sense disambiguation challenging. Another facts that can also make these approach in trouble when combines local context within clusters and global context between clusters which helps in disambiguating senses effectively therefore clustering approaches have good lead over this methods.

2.3. Clustering based WSD methods

A numerous cluster-based approaches have shown promising performance in WSD tasks that leveraged the inherent similarity and relatedness between instances to group them into clusters based on shared characteristics or context.

Following reasons explored why cluster-based approaches are effective in WSD.

- **Capture sensation Variation**: By grouping instances that have comparable contextual properties, cluster-based techniques may efficiently capture sensation variation [6].

- **Utilise Contextual Information**: To establish the sense of a target word by considering its surroundings. These methods can find patterns and relationships that help with disambiguation by examining the contextual parallels and divergences between occurrences.

- **Handle Polysemy and Homonymy**: In order to successfully handle these difficulties, cluster-based techniques divide instances into many clusters, each of which stands for a particular word meaning. As a result, more precise sense assignments may be made for terms having many meanings or ambiguous definitions.

- **Find Word Sense Relationships**: This approach can reveal hidden semantic connections between word senses. Hierarchies, similarities, and differences amongst senses may be found by looking at the connections between clusters and their sub-clusters. The process of disambiguation may be improved with the use of this information, which can also shed light on the semantic makeup of the intended term.

2.3.1. (Partially) supervised clustering

According to above effective facts, author proposed a classification based method which can utilized natural partitioning over mixed data (labeled and unlabeled) by maximizing stability criteria [15].

Method outperformed in order identification manner with semi-supervised k-means as base classifier. They investigated the stability criteria, which evaluates the degree of agreement between the sampled mixed data and the entire mixed data classification results. Following the assessment of the number of clusters, the ELP method was used to divide the mixed data into groups based on the estimated number of clusters, with each cluster being made up of comparable samples from the mixed data.

Later on, by investigated use of unlabeled data in semi supervised way through bootstrapping algorithm [10]. Another variations which associated with linear dimensionality reduction [4], where more separated clusters specially non-linear contexts are explored by group of authors through PMI based network clustering approach.

Utilizes a Semi-Autoencoder (SeAE) in the representation layer along with pair-wise constraint matrix based on PMI for accurate learning of distinctive features.

2.3.2. Unsupervised clustering

In 2004, group of authors [19] has designed unsupervised based clustering method which utilized context words surrounding noun and find relative candidates based on co-occurrence frequency which captured polysemy in noun effectively. They performed result analysis over WordNet. As getting effective performance over lexical resources further, where authors [3] overcome the sparseness of WordNet relations. Additionally they collect results for coarse-grained in English all-words task and fine-grained sub-task.

With the help of SVM classifiers over simple domain adaptation techniques, authors have given another aspect in such traditions where chosen senses utilizes for constructed clusters instances automatically [20]. Effective results are gained upto 74.7% precision score. Another unsupervised cluster based approach was explored by authors [13] where they utilized features' vector built from wordnet to represent senses as applicable over star clustering algorithm. Authors captured best performance among all unsupervised systems in SemEval-2007 with 72% F1-score.

In the area of information retrieval, authors have proposed unsupervised method for WSD [6] that utilized spectral clustering and reorders initially retrieved documents list by boosting documents that are semantically similar.

2.3.3. Other clustering methods

With the help of multilevel annotations markers [12] over lexical(lemma), tags(lex), grammatical tags, semantic taxonomy and combinations of these tags author effectively performed WSD task over RNC (Russian net corpus).

A very effective kernel based method over WSD was explored by authors [11] where they perform in-depth analysis and discussion of different strategies for representing context of polysemous words. Additionally, they explored kernel based strategies of feature selection and domain adaptation.

In 2017, a group of authors proposed sense inventory based unsupervised method [18], that utilized existing word embedding via clustering of ego-networks of related words.They got claimed effective results over the sense-balanced TWSI dataset as with 72% recall for WSD task.

Later on at very latest, authors proposed clustering based method by utilizing MFS (most frequent sense) of word [16] and treated with learned distribution which are effectively scored for other languages also.

2.4. Gap findings

Based on the previous work reviewed, there are several gaps and opportunities for further research on the proposed idea of using multilevel clustering for Word Sense Disambiguation (WSD).

Here is a summarized list that can supported for how multilevel clustering can be effective in addressing them:

- The successful organization of senses into hierarchies by multilevel clustering enables more subtle and precise disambiguation. It can ensure fine-grained sense distinctions.
- A number of methods need labeled data, which may be expensive and time-consuming. Multilevel clustering can lessen the demand for labeled data and make WSD more accessible, especially when used with unsupervised learning techniques.
- When words have homonymy and polysemous nature in fact in a very dense manner as Figure 1, traditional approaches may have trouble separating them. Such complexity may be managed by multilevel clustering, which efficiently captures sense variations and similarities hopefully.
- Multilevel clustering can more accurately capture context-dependent sense changes, resulting in more accurate disambiguation findings, by contextually grouping word senses.

Further in Section 3, proposed methodology along with detailed mathematical formulations are explored with multilevel clustering techniques that can lead to significant advancements in the field of WSD.

3. Methodology

The detailed mathematical formulation presented here outlines with following subsections of block diagram of proposed framework, formal notations, definitions and mathematical formulations as involved in multilevel clustering based WSD.

3.1. Block diagram of proposed framework

According to Figure 2, where to implement hybrid methodology of word sense disambiguation using three major modules as first to develop **text_instances**, for representing such instances to opt SemCor corpus which consists of a data file(.xml) with sense repository and this repository built through WordNet 3.0. This corpus also contain sentences with population as 37,176 into 352 different documents as docs.

Second main module of **context-embedding** that is considered here due to different sentences like simple and complex/compound type sentences are presented here, therefore go to next step for constructing context-embedding which is based on center-embedding with left and right contextual information inspired by [7]. Such context-embedding pass into clustering process where embedding of ambiguous word along with its context are operated in first level K-mean clustering. Process of how to calculate such center-embedding is discussed in next subsection 3.4.1 where The context embedding E_c is generated by applying a pre-trained (Spacy)word embedding model to each context token c_i. It capture the semantic information of the tokens and represent them as d-dimensional vectors.

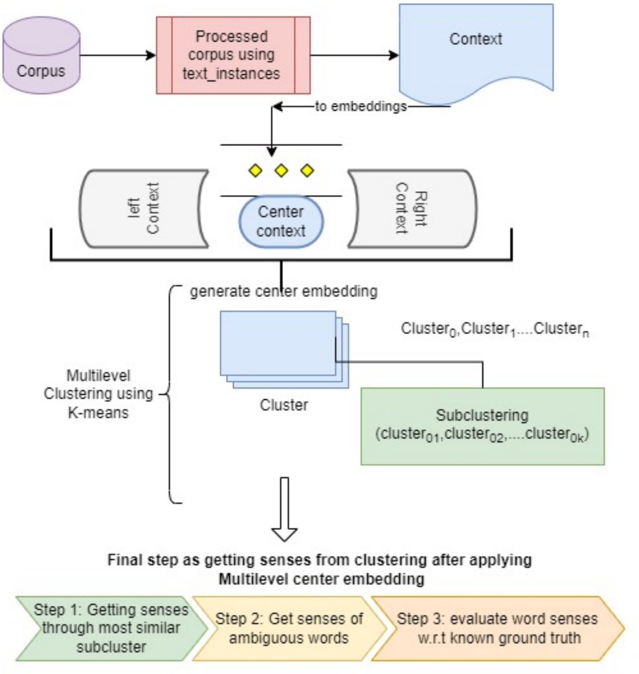


Figure 2. Block diagram for implementing Hybrid methodology of WSD task using Multilevel Clustering and center-embedding based context embedding. Figure consists with three major modules as instance collection from sense-directory, context embeddings, and nested level clustering

Now finally with significant module of **multilevel clustering** where the k-means algorithm minimizes the within-cluster sum of squares, which is equivalent to minimizing the squared Euclidean distances between data points and their respective cluster centers. Thus, by assigning definitions to the closest cluster centers, ensured that the definitions within each cluster are similar to each other and dissimilar from definitions in other clusters. At each level, the cluster centers are refined by computing the mean of the assigned definition vectors. This ensures that the new centers are representative of the definitions assigned to their respective clusters.

Applicability of proposed hybrid methodology, where nested level clustering with center-embedding is not only performed in levels but also in sub-levels, and finally go to capture most similar sub-clusters which are evaluated further with respect to known ground truth. This approach leverages the idea with center-embedding based context embedding into clustering procedure that the most common sense among similar definitions is likely to be the correct sense.

3.2. Optimal choice of number of level of clustering

As per proposed methodology, in which context-embedding is playing significant role to impose over clustering process. Proposed word sense disambiguation Algorithm-1 in further section that apply constraint between length of context embedding and number of clusters that needed as multilevel clustering. Therefore, we performed the task of getting frequent occurring of contexts up to clusters involvement as Figure 3, which clearly indicated here maximum number of contexts/senses (574361) are frequently lie at cluster 1, and then only with 2008 contexts are frequently observed when number of clusters is 2. Rest of contexts/senses which are very less frequent as 31, 6, 1 at number of clusters 3, 4 and 5 respectively. Therefore, with this observation, we decided to perform multilevel clustering upto level 2 which can performed over maximum possible contexts/senses through proposed hybrid methodology.

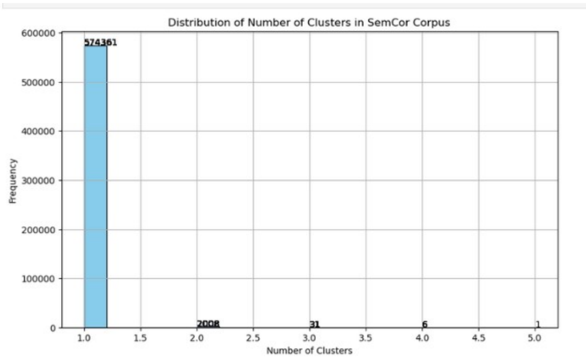


Figure 3. Reflecting Optimal choice of Nested-levels Clusters for the multilevel clustering sub-module of methodology

3.3. Formal notations

$$\begin{cases} C \Rightarrow c_1, c_2, ..., c_{N_c}, \text{ Set of context tokens \& } N_c \text{ is the number of context tokens.} \\ W \Rightarrow w_1, w_2, ..., w_{N_w}, \text{Set of word tokens, where } N_w \text{ is the number of word tokens.} \\ E_c \Rightarrow \text{Context embedding matrix}, \in \mathbb{R}^{N_c \times d}, \text{where } d \text{ is the dimensionality.} \\ E_w \Rightarrow \text{Word embedding matrix}, \in \mathbb{R}^{N_w \times d}, \\ k \Rightarrow \text{Number of clusters for context token embeddings.} \\ X_c \Rightarrow \text{Set of context embeddings as } \quad x_1, x_2, ..., x_{N_c}, \\ \text{where } x_i \text{ is the embedding for context token } c_i. \\ X_s \Rightarrow \text{Set of context embeddings in the most similar cluster as } \quad x_{1'}, x_{2'}, ..., x_M, \\ M \text{ is the number of embeddings in the most similar cluster.} \\ C_j \Rightarrow \text{Set of context embeddings assigned to cluster } j \text{ as } \quad x_1, x_2, ..., x_m, \\ m \text{ is the number of embeddings in cluster } j. \\ \mu_j \Rightarrow \text{Centroid of cluster } j \text{ as} \in \mathbb{R}^d, \\ \text{calculated as the mean of context embeddings in cluster } C_j. \end{cases} \tag{1}$$

3.4. Mathematical formulation

After assigning notations, now these notations are utilizing in formulation of proposed idea. Our formulation described in terms of following steps as pre-processing, word embeddings, and context embeddings, calculating Cluster-Centroids, level-2 clustering, to assign mapping Between Word Tokens and Synsets, and finally to retrieve words in the Most Similar Cluster. Preprocess the context and w_a using appropriate techniques, such as tokenization, stemming, and stop-word removal.

3.4.1. Word and context embeddings

By taking these notations as context C consisting of context tokens c_i, where $i = 1, 2, ..., N_c$. Word embeddings E_w for each word token w_i, where $i = 1, 2, ..., N_w$ and to choose pre-trained word embedding model. Now the aim is to generate context embeddings that capture the contextual information by considering the neighboring words within the context [5].

Definition 1 (Context Window, Word Embeddings and calculation of Center embedding)**.** ***To extract Context Window:*** *For each context token c_i in the context C, extract the context window of size k, which includes the $(k - 1)$ preceding and $(k - 1)$ succeeding tokens around c_i. Let $CW(c_i)$ denote the context window extracted for the token c_i.*

Word Embeddings*: By representing each token in the context window $CW(c_i)$ using its corresponding word embedding. Let $E_w(c_i)$ denote the word embedding for the token c_i.*

Finally, go with ***Center Embedding Calculation*** *as for the context window $CW(c_i)$, take the average of the word embeddings within the window. Let $CE(c_i)$ represent the center embedding [7] for the token c_i.*

Proof. Let $CW(c_i) = c_{i-k}, ..., c_{i-1}, c_i\widehat{c_i}, c_{i+1}, ..., c_{i+k}$ be the context window extracted around the token c_i, where k is the size of the context window. For each token c_j in the context window $CW(c_i)$, we can represent its word embedding using $E_w(c_j)$. The center embedding $CE(c_i)$ can be calculated as the average of the word embeddings within the context window as

$$CE(c_i) = (1/(2k + 1)) \cdot \Sigma_{c_j \in CW(c_i)}(E_w(c_j))$$

where Σ denotes summation and $(2k+1)$ represents the total number of tokens in the context window. By averaging the word embeddings within the context window, the center embedding captures the contextual information of the token c_i by considering its neighboring words. It allows us to represent the token in a vector space that encodes the semantic information within its local context. □

With this formulation, we can generate center embeddings for each context token in the given context. These center embeddings serve as the context embeddings, which can be used for further steps in the Word Sense Disambiguation task, such as clustering or similarity comparison.

3.4.2. Cluster centroid calculation

Definition 2. *Apply the k-means clustering algorithm to the context embeddings X_c with k clusters to obtain the cluster assignments and centroids at the first level.*

Proof. Given the context embeddings $X_c = x_1, x_2, ..., x_{N_c}$ and the number of clusters k, we can apply the k-means algorithm to X_c, which assigns each embedding to one of the k clusters and calculates the mean of the embeddings within each cluster. Let $C_1 = C_{1_1}, C_{1_2}, ..., C_{1_k}$ be the set of clusters at the first level, where each C_{1_j} represents a cluster with its assigned context embeddings. The centroid μ_{1_j} of cluster C_{1_j} is calculated as the mean of the context embeddings in C_{1_j} as following Equation (2).

$$\mu_{1_j} = (1/\mid C_{1_j} \mid) \cdot \Sigma_{x_{i \in C_{1_j}}}(x_i) \tag{2}$$

where $\mid C_{1_j} \mid$ represents the cardinality of cluster C_{1_j}, i.e., the number of context embeddings assigned to cluster C_{1_j}. □

3.4.3. Second level clustering

Definition 3. *For each cluster C_{1_j} at the first level, apply the k-means clustering algorithm with a different number of clusters (e.g., $K_2 < k$) to the centroid embeddings μ_{1_j} obtained from the previous step. This results in the second level clusters C_{2_j} corresponding to each first level cluster C_{1_j}.*

Proof. Given the centroid embeddings $\mu_{1_j} = \mu_{1_{j_1}}, \mu_{1_{j_2}}, ..., \mu_{1_{j_{K_1}}}$ for the first level cluster C_{1_j}, we can apply the k-means algorithm to μ_{1_j} with K_2 clusters, where K_2 is the desired number of clusters at the second level. This assigns each centroid embedding to one of the K_2 clusters and calculates the mean of the embeddings within each cluster. Let $C_{2_j} = C_{2_{j_1}}, C_{2_{j_2}}, ..., C_{2_{j_{K_2}}}$ be the set of clusters at the second level corresponding to the first level cluster C_{1_j}. The centroid $\mu_{2_{j_k}}$ of cluster $C_{2_{j_k}}$ is calculated as the mean of the centroid embeddings in $C_{2_{j_k}}$ in Equation (3) as.

$$\mu_{2_{j_k}} = (1/\mid C_{2_{j_k}} \mid) \cdot \Sigma_{\mu_{1_i} \in C_{2_{j_k}}}(\mu_{1_i}) \tag{3}$$

where $\mid C_{2_{j_k}} \mid$ represents the cardinality of cluster $C_{2_{j_k}}$, i.e., the number of centroid embeddings assigned to cluster $C_{2_{j_k}}$. □

3.4.4. Mapping between word tokens and synsets

As to perform lemmatization on each word token w_i to obtain its lemma form, denoted as $lemma(w_i)$. For each lemma form $lemma(w_i)$, find the corresponding synsets in WordNet, denoted as S_i. Establish then the mapping between word tokens and synsets, associating each word token w_i with its set of synsets S_i.

Definition 4. *To performs lemmatization on each word token w_i to obtain its lemma form, $lemma(w_i)$. It then finds the corresponding synsets in WordNet, denoted as S_i, which represents the possible senses of the word.*

Proof. Lemmatization is a linguistic process that transforms a word into its base or dictionary form. By applying lemmatization to each word token w_i, we obtain $lemma(w_i)$, which represents the base form of w_i. WordNet provides a lexical database that maps words to synsets, which are sets of synonymous words with a shared meaning. Thus, we can find the corresponding synsets S_i in WordNet for each $lemma(w_i)$, representing the possible senses of the word. □

3.4.5. Retrieve words in the most similar cluster

Definition 5. *For the ambiguous word w_a, calculate its embedding vector E_a using the pre-trained word embedding model. Calculate the dissimilarity between E_a and each centroid $\mu_{2_{j_k}}$ at the second level using cosine similarity. Select the most similar cluster C_{2_s}, which has the smallest dissimilarity, and retrieve the words in C_{2_s} as the disambiguated senses of the ambiguous word.*

Proof. Let E_a be the embedding vector for the ambiguous word w_a. Calculate the dis-similarity between E_a and each centroid $\mu_{2_{j_k}}$ using cosine similarity measure. Select the cluster C_{2_s} that minimizes the dis-similarity, i.e., $C_{2_s} = argmin(dis\text{-}similarity(E_a, \mu_{2_{j_k}}))$. Retrieve the words in C_{2_s} as the disambiguated senses of the ambiguous word. Let E_a be the embedding vector for the ambiguous word w_a, and let $\mu_{2_{j_k}}$ be the centroid embedding vector for cluster $C_{2_{j_k}}$ at the second level. The cosine similarity between E_a and $\mu_{2_{j_k}}$ can be calculated as following Equation (4).

$$similarity(E_a, \mu_{2_{j_k}}) = (E_a \cdot \mu_{2_{j_k}})/(\|E_a\| \cdot \left\|\mu_{2_{j_k}}\right\|) \tag{4}$$

where $\cdot$ denotes the dot product and $\|\|$ represents the Euclidean norm. The cosine similarity ranges between -1 and 1, with higher values indicating more similarity and lower values indicating more dis-similarity. However, we want to calculate dis-similarity, so we can define the dis-similarity as $1 - similarity(E_a, \mu_{2_{j_k}})$. This way, higher values indicate more dis-similarity. To find the most similar cluster, we iterate over all centroid embeddings $\mu_{2_{j_k}}$ and calculate the dissimilarity now and then to select the cluster C_{2_s} that minimizes the dissimilarity as following Equation (5).

$$C_{2_s} = argmin(1 - similarity(E_a, \mu_{2_{j_k}})) \tag{5}$$

Finally, we retrieve the words in cluster C_{2_s} as the disambiguated senses of the ambiguous word w_a.

□

3.5. Word sense disambiguation

As per above mathematical formulation, to implement with this following Algorithm 1 of the WSD, that is to determine the correct sense of the ambiguous word w_a within the given context C. It takes arguments as context, an ambiguous word, word embedding, cluster centroids, and a dissimilarity measure as input.

Algorithm 1 Word Sense Disambiguation (WSD) with nested level(2) clustering and center embedding calculation

function WSD(C, w_a, E_w, μ_{1_j}, $\mu_{2_{j_k}}$, dissimilarity measure)
    ▷ To perform Pre-processing
    preprocess_context_tokens(C)
    preprocess_ambiguous_word(w_a)
    ▷ Word and Context Embeddings
    $E_a \leftarrow$ **calculate_embedding**(w_a, E_w)
    ▷ First-level Clustering
    $\textit{min\_dissimilarity} \leftarrow \infty$
    $\textit{most\_similar\_cluster} \leftarrow$ None
    for each centroid μ_{1_j} in first-level cluster centroids **do**
        $\textit{dissimilarity} \leftarrow$ **calculate_dissimilarity**(E_a, μ_{1_j}, dissimilarity measure)
        if $\textit{dissimilarity} < \textit{min\_dissimilarity}$ **then**
            $\textit{min\_dissimilarity} \leftarrow \textit{dissimilarity}$
            $\textit{most\_similar\_cluster} \leftarrow$ cluster associated with μ_{1_j}
    ▷ Retrieve Words in the Most Similar First-level Cluster
    $\textit{disambiguated\_senses\_level\_1} \leftarrow$ **retrieve_words**($\textit{most\_similar\_cluster}$)
    ▷ Second-level Clustering
    $\textit{min\_dissimilarity\_level\_2} \leftarrow \infty$
    $\textit{most\_similar\_subcluster} \leftarrow$ None
    for each centroid $\mu_{2_{j_k}}$ in second-level cluster centroids **do**
        $\textit{dissimilarity\_level\_2} \leftarrow$ **calculate_dissimilarity**(E_a, $\mu_{2_{j_k}}$, dissimilarity measure)
        if $\textit{dissimilarity\_level\_2} < \textit{min\_dissimilarity\_level\_2}$ **then**
            $\textit{min\_dissimilarity\_level\_2} \leftarrow \textit{dissimilarity\_level\_2}$
            $\textit{most\_similar\_subcluster} \leftarrow$ subcluster associated with $\mu_{2_{j_k}}$
    ▷ Return Words in the Most Similar Second-level/First level cluster(s)
    return $\textit{disambiguated\_senses\_level\_1}$, $\textit{disambiguated\_senses\_level\_2}$
function CALCULATE_EMBEDDING(w_a, E_w)
    $\textit{target\_index} \leftarrow$ index of w_a in $\textit{tokens}$
    $\textit{start\_index} \leftarrow \max(0, \textit{target\_index} - \textit{context\_window\_size})$
    $\textit{end\_index} \leftarrow \min(\text{len}(\textit{tokens}), \textit{target\_index} + \textit{context\_window\_size} + 1)$
    $\textit{context\_window} \leftarrow \textit{tokens}[\textit{start\_index} : \textit{end\_index}]$
    $\textit{word\_embeddings} \leftarrow [E_w(c_j) \text{ for } c_j \in \textit{context\_window}]$
    $\textit{center\_embedding} \leftarrow \frac{1}{\text{len}(\textit{word\_embeddings})} \times \sum_{c_j \in \textit{context\_window}} E_w(c_j)$
    return $\textit{center\_embedding}$

It proceeds with initial tasks as $Tokenizes$ and preprocesses the context C and ambiguous word. It generates the word embedding for ambiguous word, calculating the word and context embedding using the pre-trained (Spacy) word embeddings. Now furhter go with First-level Clustering where For each centroid μ_{1_j} in the first-level cluster centroids, calculate dissimilarity between word embedding and each centroid, and then to identify the cluster with the minimum dissimilarity as the most similar first-level cluster. Now to get the words in the most similar first-level cluster.

After go through with first level clustering go then second level clustering where for each centroids $\mu_{2_{j_k}}$ again to calculate dissimilarity between word embedding and each centroid $\mu_{2_{j_k}}$. Here we now identify the sub-cluster with the minimum dissimilarity as the most similar second-level sub-cluster. Finally return by selecting words in the most similar first level cluster and second-level sub-cluster, and the procedure returns the set of disambiguated senses of the ambiguous word w_a.

4. Experimental setup

Corpus: For Word Sense Disambiguation (WSD), the **Semcor** corpus is a frequently used benchmark dataset in computer linguistics and natural language processing. It is a piece of the Brown Corpus, a bigger corpus that includes literature from numerous genres. The Semcor corpus, which was created especially for WSD tasks, consists of English phrases that have WordNet lexical database word sense tags added to them [14]. It offers an important resource for analyzing and creating WSD models and algorithms. The Semcor corpus' main traits and qualities are as follows:

- Each word in the Semcor corpus has its associated WordNet sense explicitly tagged next to it.
- When several senses of a word are present in WordNet, the Semcor corpus concentrates on adding sense annotations to potentially ambiguous terms. The annotations try to clarify each uncertain word's appropriate meaning in the context.
- The Semcor corpus's annotations are in line with the WordNet sense inventory, which makes it easier to create and test WSD algorithms that rely on WordNet for sense definitions and distinctions.

Pre-trained embedding: A library of **spaCy** provide a valuable utility in natural language processing (NLP) tasks, including Word Sense Disambiguation (WSD). Here's a brief overview of the utility of pre-trained embeddings from spaCy:

- It can leverage this semantic information to better understand the meaning of words and their potential senses within a given context.
- The reduced dimensionality makes it computationally efficient to process and compare word embeddings, which is crucial in tasks like WSD where many word senses need to be considered.

As getting the source information from authors [19], where they give summarize three benchmark datasets as Semcor, Senseval, and Korena WN where maximum number of polysemous words are presented in Semcor with large number of instances population which can be helpful to identify senses. As with such data availability, we have opted to processed our proposed work in Semcor Corpus.

5. Results and discussion

In this article, we have first calculated performance score in terms of precision, recall and F1-score on SemCor corpus as a success indication. As a result, in our effort using the this corpus, we picked the target terms in unsupervised manner using multilevel clustering. For this reason, we have chosen a word as a target term only if its semantic differences are distinct in corpus. This dataset, in our opinion, provides more realistic accuracy results as compared to existing state-of-arts. The following findings in terms of various results are shown here with regard under sub-clusters up-to level-2 as the Algorithm 1.

5.1. Performance of proposed method

According to implement proposed WSD task using multilevel clustering over chosen corpus where (around 170000 sentences are correctly processed) to get overall F1-score as **88.36%** as Figure 4. Due to that, most of text (in green colored bars shown correctly captured while white color-strips reflected as not-captured senses) in corpus treated successfully under level-1/2 sub-clustering as proposed in Algorithm 1. Some of samples that are also getting predicted senses with individual performance score is represented below as Figure 5 where with respect to use a context, ambiguous word and along with known senses getting after executing Algorithm 1.

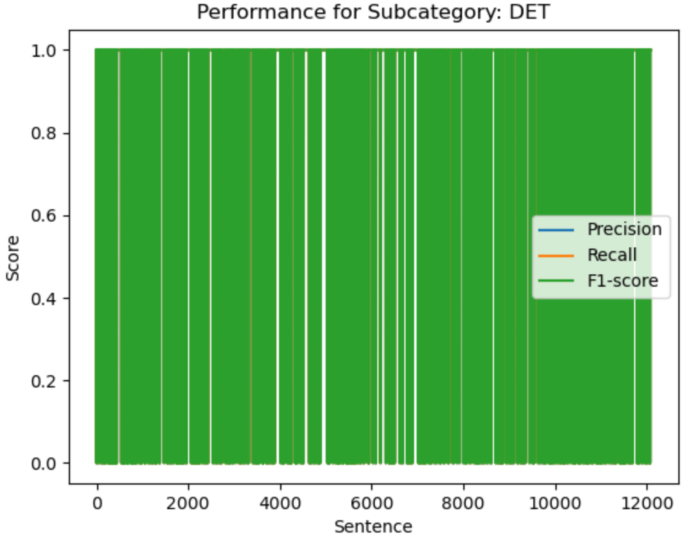
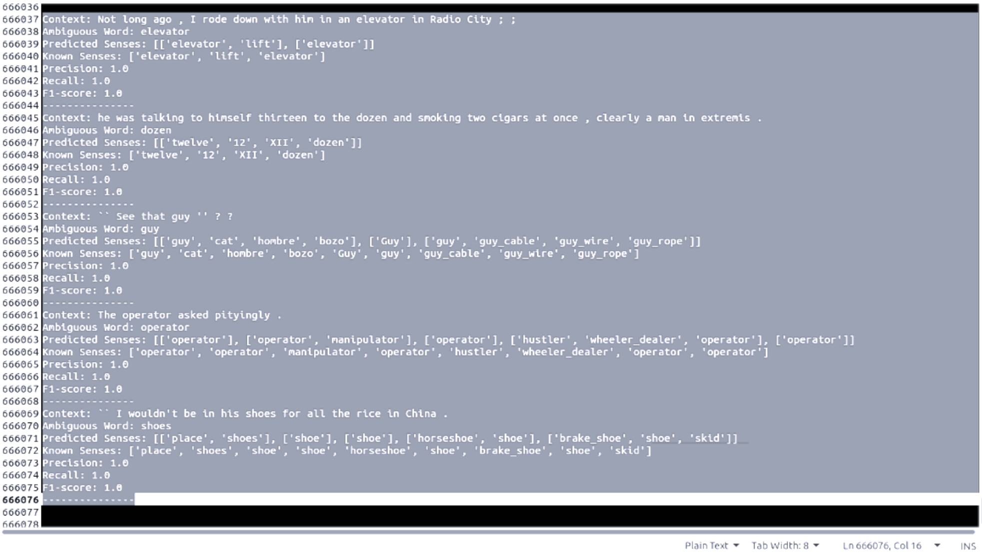


Figure 4. Visualization of Performance score of Proposed Hybrid methodology for WSD using multilevel clustering and center embedding based context embedding over sub-category sentences (with population of 12000) in SemCor Corpus



```
Context: Not long ago , I rode down with him in an elevator in Radio City ; ;
Ambiguous Word: elevator
Predicted Senses: [['elevator', 'lift'], ['elevator']]
Known Senses: ['elevator', 'lift', 'elevator']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: he was talking to himself thirteen to the dozen and smoking two cigars at once , clearly a man in extremis .
Ambiguous Word: dozen
Predicted Senses: [['twelve', '12', 'XII', 'dozen']]
Known Senses: ['twelve', '12', 'XII', 'dozen']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: `` See that guy '' ? ?
Ambiguous Word: guy
Predicted Senses: [['guy', 'cat', 'hombre', 'bozo'], ['Guy'], ['guy', 'guy_cable', 'guy_wire', 'guy_rope']]
Known Senses: ['guy', 'cat', 'hombre', 'bozo', 'Guy', 'guy', 'guy_cable', 'guy_wire', 'guy_rope']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: The operator asked pityingly .
Ambiguous Word: operator
Predicted Senses: [['operator'], ['operator', 'manipulator'], ['operator'], ['hustler', 'wheeler_dealer', 'operator'], ['operator']]
Known Senses: ['operator', 'operator', 'manipulator', 'operator', 'hustler', 'wheeler_dealer', 'operator', 'operator']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
Context: `` I wouldn't be in his shoes for all the rice in China .
Ambiguous Word: shoes
Predicted Senses: [['place', 'shoes'], ['shoe'], ['shoe'], ['horseshoe', 'shoe'], ['brake_shoe', 'shoe', 'skid']]
Known Senses: ['place', 'shoes', 'shoe', 'shoe', 'horseshoe', 'shoe', 'brake_shoe', 'shoe', 'skid']
Precision: 1.0
Recall: 1.0
F1-score: 1.0
---------------
```

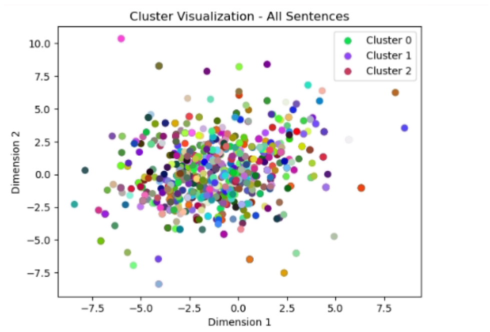

Figure 5. Examined 5 random Cases from Semcor corpus: Individual sense predictions of successfully captured senses which represented as [Contexts: captured sentence from corpus, Ambiguous word, Predicted Senses, Known Senses: ground truth values from sense-directory, with effective precision, recall, F1-score]

After getting such effective score over SemCor corpus, we have also performed some other demonstrations which are shown and discussed in following sub-sections to get in-depth analysis of proposed idea over WSD using multilevel clustering upto level-1 & 2 processing.

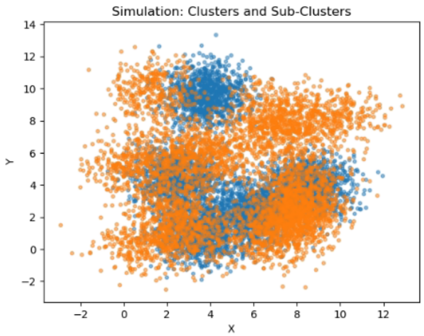
5.2. Demonstration of clustering results

As according to proposed idea where the clustering of word embedding based on the k-means clustering algorithm. Each point in the plot represents a word context, and the color of the point corresponds to the predicted sense of that word context. Figure 6 give some remarks over here as it show clear boundaries between clusters, indicating that the clustering process effectively groups word contexts with similar meanings together. Different clusters should contain points in Figure 6a and 6b, representing various word senses, showing that the WSD method captures distinct word senses and assigns them to separate clusters. Overall If clusters significantly overlap, it might suggest ambiguity in word senses or challenges in disambiguation, which could indicate why here is necessary to incorporate multilevel clustering.

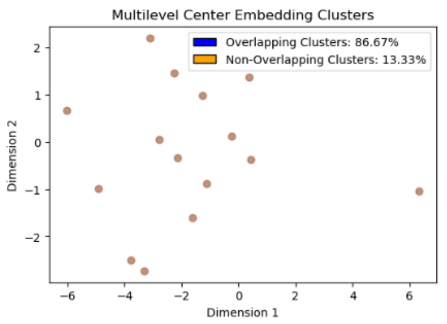
With respect to the overlapping nature of the clusters as Figure 6c and 6d, it categorized the points into two groups: overlapping clusters and non-overlapping clusters based on the k-means clustering results. Overlapping clusters suggest that multiple word senses are assigned to the same cluster, indicating potential polysemy or ambiguous contexts. A higher overlapping percentage might indicate challenges in disambiguating word senses, while a lower percentage suggests successful separation of senses into distinct clusters.



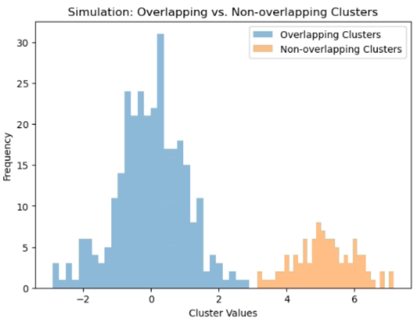
a) Clusters representation of sentences which shown about how these are grouped as in three different grouped-contexts cluster 0,1,2



b) Distinguish representation of clusters with sub-cluster representation that shown about mostly contexts are classified into cluster and its sub-clusters



c) When center embedding applied over these clusters: 86% clusters are overlapping which help in performance of proposed WSD task, non-overlapping 13% score is depicting about limitation of proposed methodology

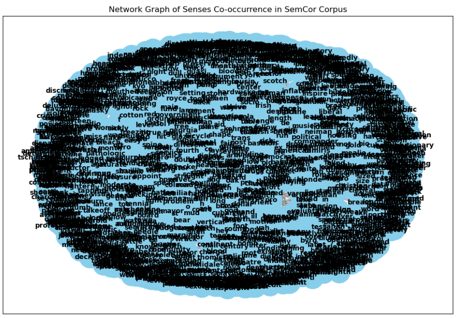


d) Another view of Figure 6c through bar chart which more clear representation of overlapping score among clusters

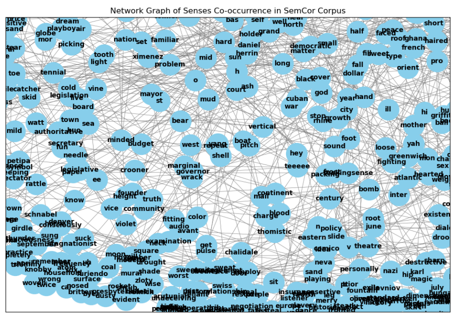
Figure 6. Clustering demonstration and Some key findings are observed using a, b, c, d sub-Figures: It also give support and insist the possibility of why chosen multilevel clustering upto the cluster with level 1 and 2

5.3. Distribution of senses in corpus as captured through proposed formulation

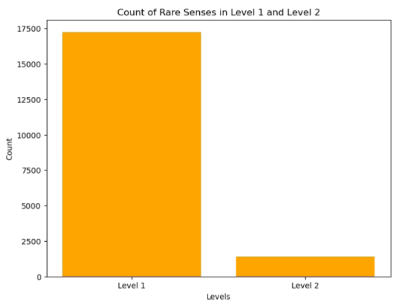
For the demonstration of senses distribution, in which shows relationship in co-occurrence, how much senses distributed in levels and what these senses connected to each others in the SemCor corpus. It helps to understand which senses are more prevalent and how they appear in the context. This information is crucial for understanding the overall sense usage in the corpus and identifying dominant senses or those that occur more frequently. The co-occurrence network graph as Figure 7a ang Figure 7b show how different senses are related to each other based on their co-occurrences in the SemCor corpus. It helps to identify senses that tend to co-occur frequently, indicating that they are semantically related or used together in similar contexts. As proposed initially to cover the aspect of polysemy, this visualization where a sense co-occurs with multiple other senses, it may indicate polysemy, where a word has multiple related senses. With the help of Figure 7d, where findings as with level-1 having approx 72% senses presence while in level-2 value is approx 19% as with proposed idea. In order to execute WSD utilizing multilevel clustering, it was shown that the majority of senses could be captured in level 1 in order to satisfy the greatest number of senses through connected contexts, while level 2 could considerably improve the performance score of the proposed technique for the situations of existing the majority of rare senses as in Figure 7c.



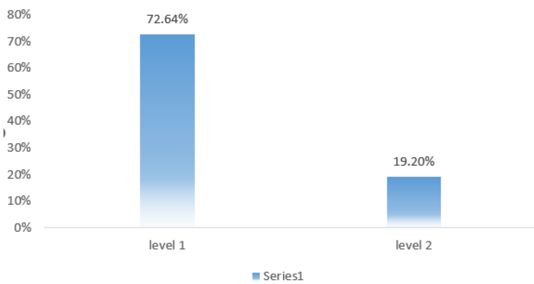
a) Network graph representation: Sense with co-occurrence contexts are captured for whole SemCor corpus, this clearly represent about how contexts/senses are closely bound in network



b) Amplified version of Figure 7a (bottom left part of network) about sense with co-occurrence graph so that to its dense nature are captured clearly



c) Predicted rare-senses score in leveled manner



d) Score of predicted senses in level-1 and level-2

Figure 7. To represent various relationship among senses with it's co-occurred senses to each other & capture senses in level 1 and 2

Additionally we have demonstrated our proposed method, by taking following example with input sentence for capturing predicted senses on the basis of similar context to input sentence and ambiguous word as Figure 8.

```
Input Sentences:
She sat by the bank of the river and watched the water flow.

Ambiguous Word: bank
Known Senses:
bank.n.01 (Cluster 6)
depository_financial_institution.n.01 (Cluster 8)
bank.n.03 (Cluster 7)
bank.n.04 (Cluster 6)
bank.n.05 (Cluster 13)
bank.n.06 (Cluster 7)
bank.n.07 (Cluster 6)
savings_bank.n.02 (Cluster 8)
bank.n.09 (Cluster 8)
bank.n.10 (Cluster 11)
bank.v.01 (Cluster 5)
bank.v.02 (Cluster 4)
bank.v.03 (Cluster 4)
bank.v.04 (Cluster 4)
bank.v.05 (Cluster 2)
deposit.v.02 (Cluster 3)
bank.v.07 (Cluster 2)
trust.v.01 (Cluster 5)
--------------
Level-1 Predicted Senses: ['savings_bank.n.02']
Level-2 Predicted Senses: ['watch.n.04', 'sit_down.v.01', 'water_system.n.02', 'stream.n.04', 'river.n.01']

Percentage of Level-1 Predicted Senses Similar to Known Senses: 100.00%
Percentage of Level-1 Predicted Senses Different from Known Senses: 0.00%
```

Figure 8. An application of proposed method which shown the working of how to predicted senses at leveled manner

In the sentence where in respect to take input with ambiguous word and find the range of known senses from WordNet hierarchy, now as proposed algorithm, find predicted senses in leveled manner where level-1 reflected the results as similar to known senses while on the other hand, when algorithm worked over level-2 it also predicted some other senses based upon the similar context to input sentence. The results are getting very attractive and meaningful at both level-1 and 2.

6. Conclusion and future directions

As proposed the hybrid methodology of word sense disambiguation (WSD) that utilizing multilevel clustering along with center-embedding based context embedding, it has demonstrated promising results in capturing various senses of ambiguous words in related contexts. The strategy grouped context terms into clusters using word/center embedding and multilevel (level-2) k-means clustering, which enables the system to speculate on possible meanings for the ambiguous word. Comparatively, the model is shown to be capable of accurately identifying pertinent senses in context, as seen by the performance evaluation's in Table 2. The system's ability to handle words with numerous meanings is shown by the polysemy capture rate of 29.6%, with low homonymy-capture rate of 2.3% implies that the model can distinguish between words with similar forms but different meanings as Figure 9.

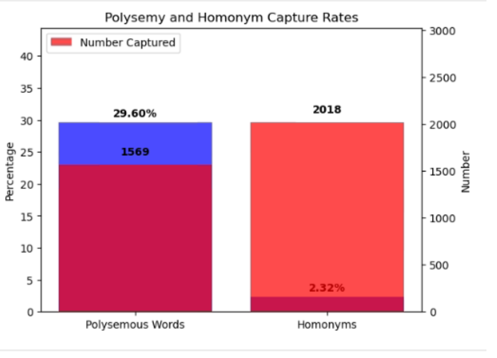


Figure 9. Polysemy and Homonym captured by proposed method

Table 2
Performance comparison of existing vs. proposed methods for Word Sense Disambiguation

Existing methods	Coverage [%]	Precision [%]	Recall [%]	F1-score [%]
[19]	100	45.4	45.4	–
[3]	100	–	–	70
[13]	–	–	–	49.8
[5]	–	66.5	64.8	65.7
[9]	100	–	–	79
As Proposed method	100	88.6	88.13	**88.36**

6.1. Limitations and future scope

Despite having achieved encouraging results, there are following rooms available of possibilities of improvements.

- In the view of Fine-Grained Sense Disambiguation where by grouping words at various levels, future research can concentrate on streamlining the clustering procedure to get a more precise word meaning disambiguation.
- In the case of contextual embedding, furhter go through with deep-learning strategies, where to utilize Bi-LSTM to handle and catpured more effective context-embedding (through treatment over complex sentences) which effectively worked over word senses hopefully.
- As the results of polysemy, the model demonstrated effectiveness in capturing a word's various meanings (polysemy), but in case of homonymy where score not observed effectively. Further research may look at methods to better handle words with similar forms but dissimilar meanings (homonymy).

When using the suggested multilevel clustering to further investigate other types of relations, such as homographs, it is hoped that the definition of a homograph would be developed to allow for the exploration of further levels while leveraging the definitions of WordNet-synsets.

Acknowledgements

With great appreciation, We would like to thank the reviewers whose insightful comments have greatly improved the quality of this article. Additionally, to express our gratitude to the Editor, along with the team-members by providing suggestions in helping to shape this work into its finished form.

Lastly, I also want to express my sincere gratitude to my son (Krishang), whose new outlook and astute insights gave me important clues for locating and resolving ambiguity issues in the texts. His advice has been very helpful in maintaining connection and clarity throughout to develop proposed algorithm.

References

[1] Agirre E., López de Lacalle O., Soroa A.: Random Walks for Knowledge-Based Word Sense Disambiguation, *Computational Linguistics*, vol. 40(1), pp. 57–84, 2014. doi: 10.1162/COLI_a_00164.

[2] AlMousa M., Benlamri R., Khoury R.: A novel word sense disambiguation approach using WordNet knowledge graph, *Computer Speech & Language*, vol. 74, 101337, 2022. doi: 10.1016/j.csl.2021.101337.

[3] Anaya-Sánchez H., Pons-Porrata A., Berlanga-Llavori R.: TKB-UO: Using Sense Clustering for WSD. In: *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 322–325, 2007. doi: 10.3115/1621474.1621544.

[4] Berahmand K., Li Y., Xu Y.: A Deep Semi-Supervised Community Detection Based on Point-Wise Mutual Information, *IEEE Transactions on Computational Social Systems*, vol. 11(3), pp. 3444–3456, 2023. doi: 10.1109/TCSS.2023.3327810.

[5] Bhingardive S., Singh D., Rudramurthy V., Redkar H., Bhattacharyya P.: Unsupervised most frequent sense detection using word embeddings. In: *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pp. 1238–1243, 2015. doi: 10.3115/v1/n15-1132.

[6] Chifu A.G., Hristea F., Mothe J., Popescu M.: Word sense discrimination in information retrieval: A spectral clustering-based approach, *Information Processing & Management*, vol. 51(2), pp. 16–31, 2015. doi: 10.1016/j.ipm.2014.10.007.

[7] Dubey S., Kohli N.: A Multilevel Center Embedding approach for Sentence Similarity having Complex structures. In: *2023 World Conference on Communication & Computing (WCONF)*, pp. 1–8, 2023. doi: 10.1109/wconf58270.2023.10235102.

[8] Guerrieri A., Rahimian F., Girdzijauskas S., Montresor A.: Tovel: Distributed graph clustering for word sense disambiguation. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 623–630, IEEE, 2016. doi: 10.1109/icdmw.2016.0094.

[9] Huang L., Sun C., Qiu X., Huang X.: GlossBERT: BERT for word sense disambiguation with gloss knowledge. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3509–3514, 2019. doi: 10.18653/v1/d19-1355.

[10] Le A.C., Shimazu A., Huynh V.N., Nguyen L.M.: Semi-supervised learning integrated with classifier combination for word sense disambiguation, *Computer Speech & Language*, vol. 22(4), pp. 330–345, 2008. doi: 10.1016/j.csl.2007.11.001.

[11] Li X., Qing S., Zhang H., Wang T., Yang H.: Kernel methods for word sense disambiguation, *Artificial Intelligence Review*, vol. 46, pp. 41–58, 2016. doi: 10.1007/s10462-015-9455-5.

[12] Lyashevskaya O., Mitrofanova O., Grachkova M., Romanov S., Shimorina A., Shurygina A.: Automatic Word Sense Disambiguation and Construction Identification Based on Corpus Multilevel Annotation. In: *Text, Speech and Dialogue: 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1–5, 2011. Proceedings 14*, pp. 80–90, Springer, 2011. doi: 10.1007/978-3-642-23538-2_11.

[13] Martín T., Berlanga-Llavori R.: A clustering-based approach for unsupervised word sense disambiguation, *Procesamiento del Lenguaje Natural*, vol. 49, pp. 49–56, 2012.

[14] Miller G.A., Chodorow M., Landes S., Leacock C., Thomas R.G.: Using a semantic concordance for sense identification. In: *HLT '94: Proceedings of the workshop on Human Language Technology*, 1994. doi: 10.3115/1075812.1075866.

[15] Niu Z.Y., Ji D.H., Tan C.L.: Learning model order from labeled and unlabeled data for partially supervised classification, with application to word sense disambiguation, *Computer Speech & Language*, vol. 21(4), pp. 609–619, 2007. doi: 10.1016/j.csl.2007.02.001.

[16] Pasini T., Scozzafava F., Scarlini B.: CluBERT: A cluster-based approach for learning sense distributions in multiple languages. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4008–4018, 2020. doi: 10.18653/v1/2020.acl-main.369.

[17] Patil A.P., Ramteke R., Bhavsar R., Darbari H.: Graph-Based Algorithm for Word Sense Disambiguation: A Performance and Comparison, *Sambodhi*, vol. 44(03), pp. 77–79, 2021.

[18] Pelevina M., Arefyev N., Biemann C., Panchenko A.: Making sense of word embeddings, *arXiv preprint arXiv:170803390*, 2017. doi: 10.48550/arXiv.1708.03390.

[19] Seo H.C., Chung H., Rim H.C., Myaeng S.H., Kim S.H.: Unsupervised word sense disambiguation using WordNet relatives, *Computer Speech & Language*, vol. 18(3), pp. 253–273, 2004. doi: 10.1016/j.csl.2004.05.004.

[20] Shirai K., Nakamura M.: JAIST: Clustering and classification based approaches for Japanese WSD. In: K. Erk, C. Strapparava (eds.), *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 379–382, Association for Computational Linguistics, Uppsala, Sweden, 2010. https://aclanthology.org/S10-1085.

[21] Vaishnav Z.B., Sajja P.S.: Knowledge-Based Approach for Word Sense Disambiguation Using Genetic Algorithm for Gujarati. In: S.C. Satapathy, A. Joshi (eds.), *Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies*, vol. 1, pp. 485–494, Springer, Singapore, 2019. doi: 10.1007/978-981-13-1742-2_48.

Affiliations

ShivKishan Dubey
Harcourt Butler Technical University, Kanpur, skd@aith.ac.in

Narendra Kohli
Department of Computer Science, HBTU Kanpur, Uttar Pradesh, nkohli@hbtu.ac.in

Received: 19.10.2023
Revised: 06.02.2024
Accepted: 20.02.2024