Supervised, Unsupervised and Semi-Supervised Word Sense Disambiguation Approaches

Anandakumar Haldorai
Sri Eshwar College of Engineering, Coimbatore, India.
anandakumar.psgtech@gmail.com

ArticleInfo
R. Arulmurugan et al. (eds.), *First International Conference on Machines, Computing and Management Technologies*, Advances in Intelligent Systems and Technologies
Doi: https://doi.org/10.53759/aist/978-9914-9946-0-5_8

Abstract – Word Sense Disambiguation (WSD) aims to help humans figure out what a word means when used in a certain setting. According to the Neuro Linguistic Programming (NLP) community, WSD is an AI-complete issue with no human solution in sight. WSD has found widespread usage in a wide variety of applications, including but not limited to: Machine translation (MT), Information Retrieval (IR), Data Mining (DM), Information Extraction (IE), and Lexicology (Lex). It is discovered that WSD may be learned effectively using a variety of different methodologies, including supervised, semi-supervised, and unsupervised methods. These methodologies are sorted into groups according to the kind and quantity of annotated (identified) corpora (data) they need as the primary source of information utilized to distinguish between senses. The unsupervised method employs unannotated (unidentifiable) corpora for training, whereas the semi-supervised method requires a less number of annotated corpora than supervised methods. All these three strategies will critically be discussed in this study.

Keywords – Word Sense Disambiguation (WSD), Dialogue for Reverse Engineering Assessment and Method (DREAM), Machine Translation (MT).

I. INTRODUCTION

Word Sense Disambiguation (WSD) [1] involves choosing the suitable meaning for an uncertain word in relation to its environment. Increasing WSD's precision may have significant positive effects on many text processing systems, where it is among the most difficult jobs. This article proposes a novel unsupervised approach that makes no assumptions about the features or structure of the target language and instead relies only on the co-occurrence graph of a monolingual corpus. An ambiguous word co-occurrence network is constructed using a corpus as the basis for the proposed technique, and a sub-graph is then used to represent the context of an ambiguous term. This network has strong connections between most of the terms. Several similarity functions are used in conjunction with the senses and context graph to determine the projected meanings of any complex or ambiguous words.

Lexical ambiguity is a basic feature of almost every language. A word may have many senses or interpretations, in which case it is referred to be an ambiguous term. WSD is the process of clearing up this kind of semantic confusion. This means that WSD is the process of identifying a suitable synonym for a term in a specific situation and using that synonym instead of the incorrect one. Take the term "grain," which has two distinct connotations in English. A tiny, tough cereal plant seeds such wheat, and the striations created by wood fibers or fabric textures, both as nouns. By analyzing the context in which a term is used, word sense disambiguation may then choose the most appropriate replacement for it. Knowledge-based training data and feature selection facilitate word-sense disambiguation. Depending on the task at hand, knowledge-oriented training data may be unlabeled or labeled with word senses, e.g., information from the fields of medicine and administration. In the field of computational linguistics, WSD was originally designed to aid in the process of Machine Translation (MT) [2]. Solutions to the problem have been proposed using a variety of unsupervised, semi-supervised and supervised learning approaches.

In order to predict signal transduction pathways from gene expression data, certain computer techniques have been created, with the great preponderance dependent on unsupervised learning. Although several studies have examined various approaches to network inference, there has yet to be a comprehensive assessment of unsupervised, supervised and semi-supervised approaches, and various questions remain unanswered. We answer essential issues like how many samples are needed for various procedures and what sorts of experimental data each approach is best suited to analyze. In the following paragraphs, we first cover evaluations that concentrate on supervised and semi-supervised approaches, and then analyze large-scale comparisons (more than five methods). Finally, we turn our attention to the remaining, more nuanced comparisons from a practical standpoint.

Wang, Ma, Wang, Tao, Ren, & Zhu [3] conducted the most latest and comprehensive comparison to date. There were a total of 38 simulated data sets used to evaluate the accuracy of prediction of one supervised and eight unsupervised methodologies. There were substantial disparities in prediction accuracy amongst the approaches, with the supervised method emerging on top even after the parameters of the unsupervised learning were improved. We compare these

unsupervised techniques against supervised and semi-supervised ones across a broad variety of networks and experimental types of data, and we expand the original research to include 17 unsupervised techniques (knockout, knockdown and multifactorial).

The DREAM (Dialogue for Reverse Engineering Assessment and Method) [4] (see **Fig. 1**) is a yearly open competition in network inference that conducts a thorough analysis of the sophistication in network inference. This evaluation is confined to unsupervised methods. The results of DREAM show how difficult it is to infer relationships between nodes in a network. A large proportion of the teams' forecasts were "computationally identical to randomized predictions," as noted by Hill, Schumacher, and Jirak [5]. Nevertheless, a significant finding from the DREAM competitions is that, in particular circumstances, straightforward approaches may succeed:"... the z-score predictions would have ranked first (tie), first and second in the 100-node, 10-node and 50-node sub-challenges, correspondingly.

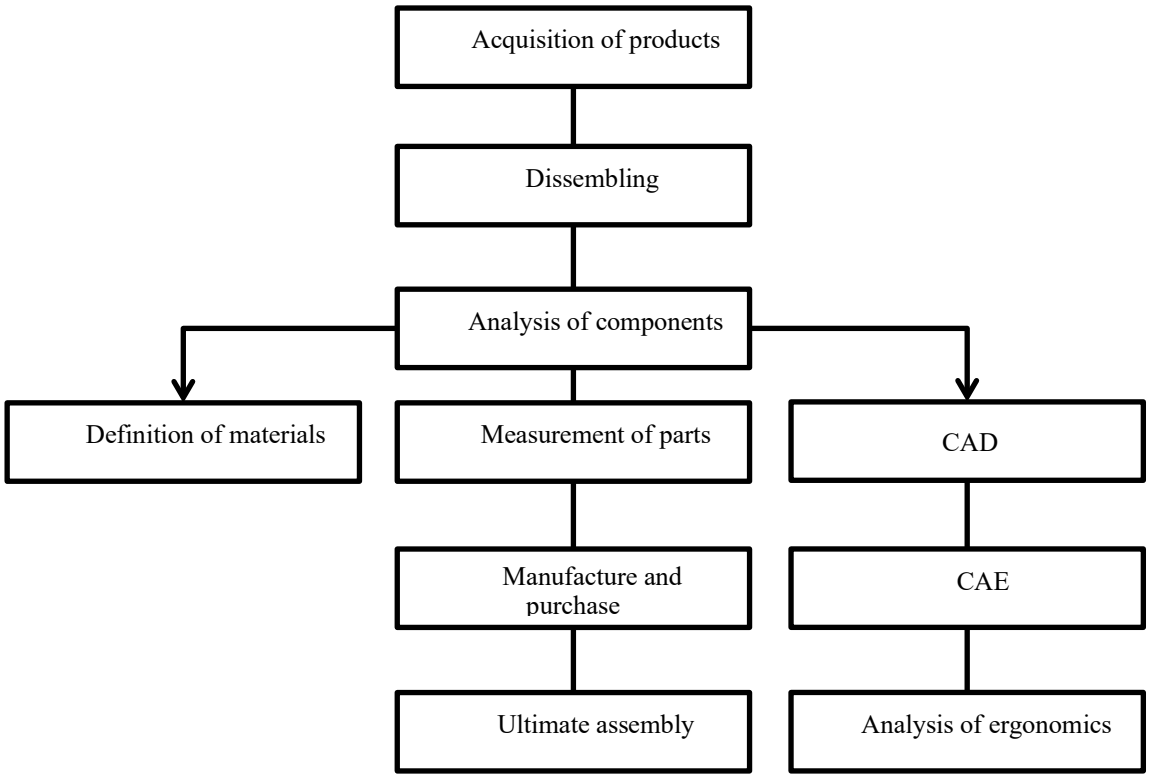

Fig 1. An organization of reverse engineering

Although expression data alone may be used by unsupervised systems, their prediction accuracy is often lower than that of supervised approaches. Supervised approaches, on the other hand, need training data regarding known interactions, which is often scarce. Although not as reliable predictors as supervised systems, semi-supervised approaches represent a middle ground and may be trained with minimal interaction datasets. Wani and Raza [6] conducted one of the few comparisons to supervised approaches; they used the benchmark data set for Escherichia coli to compare Supervised Inference of Regulatory Networks (SIRENE) to unsupervised approaches e.g., Relevance Networks (RN), Reconstruction of Accurate Cellular Networks (ARACNE), Bayesian network (BN), and Context Likelihood of Relatedness (CLR), Support Vector Machines (SVMs) were shown to be superior to two unsupervised approaches in a study by Bellotti, Matousek, and Stewart [7]. Similar semi-supervised and supervised approaches are used in our assessment, but we also utilize a large number of unsupervised methods, differentiate across experimental typpes, and conduct repeated experiments to provide a fuller perspective. Relatedly, Alejandrino, Bolacoy Jr, and Murcia [8] examined six unsupervised approaches using synthetic expression data on bigger networks of 100, 200, and 500 nodes.

Although there have been a number of smaller-scale assessments, most of them have only compared four unsupervised approaches (ARACNE, MRNET, RN, and CLR) against an unique methodology on tiny data sets. Sashalmi [9] presented the ARACNE approach, which outperformed BN and RN methodology on simulated models in terms of accuracy and recall. Park et al. [10] reviewed the bias in the forecasts of those approaches, while Wilmot [11] reviewed all four major unsupervised inference approaches on larger sub-networks of yeast (100 – 1000 nodes) by use of simulated expression dataset. To better understand the interactions between genes in E. coli, Miller, Feng, Li, and Rabitz [12] contrasted CLR to Linear Regression, RN, and ARACNE model using data from RegulonDB. Using simulated networks, the authors contrasted the prediction precision of SFFS + MCE, MRNET and ARACNE, a feature selection technique, and showed that the latter was better for networks with a modest node degree. As part of their research, Van den Bulcke et al. [13]

formed a synthetic network of regulation generator and compared the performance of four distinct network prediction algorithms (CLR, ARACNE, Symmetric-N and DBmcmc) over a range of network sizes and experimental designs.

In a study comparing RN, Graphical Gaussian Models (GGMs), and BNs, Altenbuchinger, Weihs, Quackenbush, Grabe, and Zacharias [14] examined the Raf pathway, a basic biological signalling pathway with 11 proteins, and simulation model. This study used observational data to conclude that BNs and GGMs are superior than RN. Klinger and Blüthgen [15] conducted an experiment on a simulated system with 10 genes and different levels of noise, reverse engineering by multiple regression, contrasting regulatory strengths analysis, dynamic BNs and Partial Correlations (PC). In the absence of noise, the PC strategy excelled. After constructing a simple synthetic in vivo network consisting of just five genes, zbek [16] analyzed time-series data and steady-state expressions. BANJO and ARACNE were shown to be less accurate than two models dependent on partial differential equation. To evaluate Ordinary Differential Equations (ODE), ARACNE, and BANJO, Hairer and Lubich [17] employed simulated expression data and random networks.

In the supervised method, the algorithms use wordnets or knowledge bases that have previously been trained or categorised (sense-tagged) to sort through the new data. Supervised learning approaches need a large amount of labeled data for optimal training. Data used for training in a semi-supervised setting may be either unlabelled labeled, or a mix of the two. As opposed to labeled data, the unsupervised learning method employs clustering techniques on raw data. Research on unsupervised and semi-supervised approaches for attain a state-of-the-art performance is ongoing. Here is how the remainder of the paper is organized: Section II presents a critical analysis of supervised approach discussing different approaches such as Naïve Bayes Approach, Decision List Approach, Decision Tree Approach, Support Vector Machine (SVM) Approach, Exemplar-Based Learning Approach, and Neural Network Approach. Section III focuses on semi-supervised approaches such as Yarowsky Bootstrapping Approach, Bilingual Bootstrapping Approach, and Label Propagation Approach. Section IV reviews unsupervised approaches such as Context Group Discrimination Approach, Co-occurrence Graphs Approach, and WSD using Parallel Corpora. Lastly, Section V draws final remarks to the article.

II. SUPERVISED APPROACH

A supervised system learns to discriminate the meanings of words by analyzing annotated corpora. The training and testing stages make up the supervised method. In order to construct classifiers using machine learning methods, the training step necessitates the use of a sense-annotated training corpus. In the testing step, classifiers attempt to identify the necessary senses by analyzing the context of the statement. One may utilize any of many available classifiers, often known as "word experts," to choose the most suitable category in which to place a given word. In all cases, a supervised algorithm outperforms its competitors. Similarity-based, probabilistic, discrimination-based, and linear classification-based techniques are only some of the supervised approaches that may be used. In similarity based techniques, disambiguation is achieved by comparing features of untrained dataset with features of learned dataset, and then attributing meaning to the structure with the greatest similarities.

Parameter sets, such as the conditional or joint probability distribution, may be estimated using probabilistic techniques. Using a method known as discriminating rule approaches, a new sample is classified by selecting one or more rules that match to the properties of the sample and assigning a word a meaning dependent on the rules' predictions. The capability shared by all supervised algorithms is the usage of feature pairs associated with a sense for training. Here we will go through some of the most useful supervised methods for determining the intended meaning of a word. **Table 1** lists some of the commercial applications that may make use of supervised learning models for development and improvement.

Table 1. Commercial applications for supervised learning models for development and improvement

Commercial applications for supervised learning models	
Image- and object-recognition:	The use of supervised learning methods in machine vision and image analysis allows for the detection, separation, and classification of objects in video and still pictures.
Predictive analytics:	The development of predictive analytics systems that give in-depth insights into a variety of business data points is a prominent use of supervised learning models. To better defend actions or make changes that will benefit their company, business executives may now expect specific consequences based on a particular output variable.
Customer sentiment analysis:	Organizations can automatically extract and categorize crucial information such as context, emotion, and purpose from massive datasets using supervised algorithms for machine learning. To enhance brand engagement, this information may be utilized to better comprehend client interactions.
Spam detection:	In the context of detecting spam, supervised learning has found useful use. To better organize spam and non-spam emails, businesses may use supervised segmentation methods to train datasets to spot patterns or outliers in new data.

Naïve Bayes Approach

The Naive Bayes algorithm is a type of supervised learning technique. In order to estimate probabilistic variables, this technique uses the probabilistic model, a statistical technique. Joint probability distributions and conditional probabilities are common ways that this probabilistic method communicates ideas in a particular setting and set of categories. The

algorithm makes use of classifiers that are majorly based on the theorem of Bayes to evaluate the conditional probabilities of every sense (e.g., k (or words for which elements are defined $(x_1,x_2,x_3,\ldots..x_n,)$. $P\,(xi\,/k)$ and $P\,(k)$ are the probabilistic metrics of modeling and they are calculated from the dataset, using the relative frequency count.

Decision List Approach

The decision list approach is another kind of supervised methodology used to categorize test instances based on an ordered collection of criteria. The decision list approach makes use of weighted "if then else" rules. Each word's syntactic and semantic properties, as well as its part of speech, are considered by this approach. To properly train classifiers, which must first learn to recognize the most crucial characteristics, the training labeled dataset is utilized. When establishing the predicate rules, the (feature-value, sense, score). These rules are then arranged into a decision list by sorting them into a non-increasing order. When a word is tested for ambiguity, the input feature vector is compared to the entries in the judgment list; the entries with the highest scores are chosen as the precise sense.

Decision Tree Approach

Supervised decision trees [18] are an application of the prediction-based technique. The training is performed using the sense labeled corpus as the knowledge source. In this approach, rules in the form of yes/no statements are employed to make categorization decisions. The training data is then recursively processed using these principles. Most notably, this approach uses every internal node to represent a feature, every edge to represent a feature's value, and every leaf node to represent an essential sense. When comparing the decision tree technique with the decision list methods, it is important to note that the feature vectors are same in both situations. In order to get to the leaf node of the screening tree, the ambiguous word and its associated feature vector are thoroughly investigated. After then, the meaning of the term that was obtained at a leaf node is the one that should be used. **Fig. 2** shows the decision tree that depicts solving the car-selection problem.

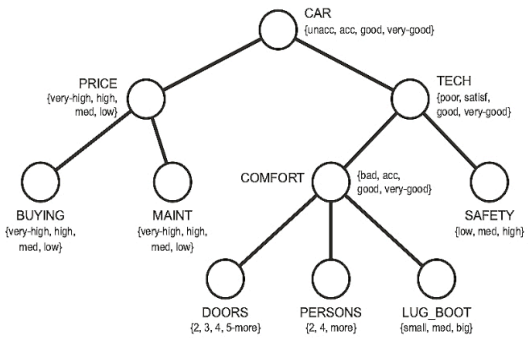

Fig 2. Solving the Car-Selection Problem with a Decision Tree

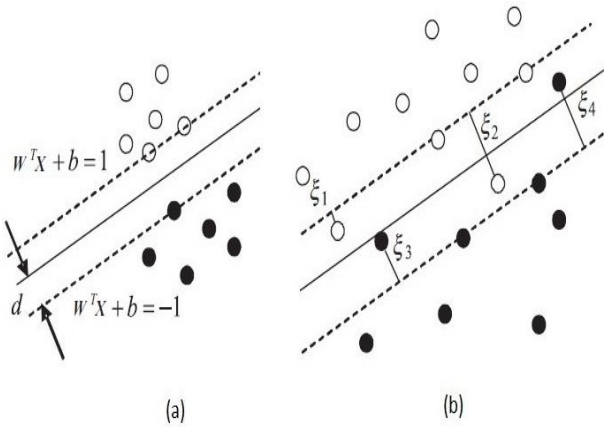

Fig 3. Representation of Support Vector Machine

Support Vector Machine (SVM) Approach

The Support Vector Machine (SVM) method is a supervised strategy for classifying data as negative or positive. It relies on the basic idea of a linear subspace constructed from a set of labeled data. This technique employs a support vector machine (SVM) binary classifier to categorize data as true or false. Multicast classification is used by SVM to disambiguate words with multiple meanings. This is then transformed into a binary classification issue, using sense Si vs. the other perceptions as the categories. **Fig. 3 (a)** shows Support Vector Machine (SVM) analysis that uses a hyperplane with the greatest possible margin d (distance between the closest samples of the classes) to distinguish between two groups of data. **Fig. 3 (b)** shows Support vector machines (SVM) for the non-separable situation, with samples near the edge punished [19]

Exemplar-Based Learning Approach

The training data is stored in memory and retrieved later using this supervised learning technique. Machine Learning is used here in the form of the k-Nearest Neighbor (kNN) technique to classify test data in accordance with the k most similar pre-recorded samples. Each characteristic of the test dataset $x = (x_1,\ldots,x_m)$ is compared with its corresponding feature in the dataset $xi = (x_i\,j,\ldots,x_i\,m)$ to determine the set of closest neighbors.

Neural Network Approach

The neural network is one in which the connections between artificial neurons are studied. In neural network technique, the ambiguous word is disambiguated using either Back-Propagation Oriented Feed Forward Network or Hidden Markow

Model. Learning methods use as their inputs a feature pair and the outcomes that are predicted from it. Using the necessary replies as a key, these input properties are utilized to divide the training settings into distinct groups. Neurons' weights are changed so that they produce the greater values needed for the desired outputs. Groups of neurons that are either chemically linked or functionally coupled make form a biological neural network. It is possible for a large number of connections to exist between a single neuron and many other neurons in a network. Synapses often occur between axons and dendrites, while other types of connections, such as dendrodendritic synapses, are conceivable. In addition to electrical signaling, signaling may also occur as a result of the diffusion of neurotransmitters.

Data processing paradigms such as Artificial Intelligence (AI), cognitive modeling, and neural networks take cues from the organization of neurons in the brain. Cognitive modeling and AI both make an effort to mimic the behavior of organic brain networks. Artificial neural networks have been effectively used to voice recognition, image analysis, and adaptive control in the area of AI, stimulating the establishment of software agents (in a variety of media, including computer and video games) and autonomous robots. In the past, the von Neumann paradigm was the basis for how digital computers worked; today, digital computers still execute explicit instructions by reading and writing data in memory. However, neural networks were developed as a result of attempts to simulate the way information is processed in living organisms. Neural network computing, in contrast to the von Neumann approach, does not partition the computer's memory from its processing power. With the help of neural network theory, scientists have gained a deeper understanding of brain neurons' roles, laying the groundwork for future AI research. **Fig. 4** shows the aspect of incorporating neural networks into an evolutionary strategy for system identification.

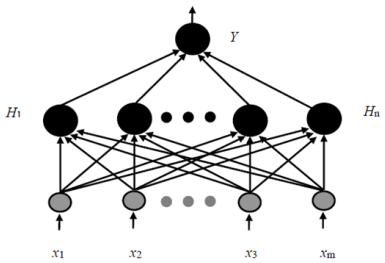

Fig 4. The Use of Neural Networks for System Identification Using an Evolutionary Approach

III. SEMI–SUPERVISED APPROACH

The semi-supervised learning method combines training on labeled and unlabeled data. A "semi supervised technique" is used because it incorporates information needed for both supervised and unsupervised methods. Most experiments suggest that the performance of machine learning systems improves when they are fed both unlabeled data and a modest amount of labeled data. Sometimes called "minimally supervised learning," the semi-supervised method has gained popularity in recent years. For ambiguity resolution, both supervised and semi-supervised approaches assume something about the language and its discourse. Data is increasing at an exponential rate, making it impossible to categorize it quickly. Consider a typical TikTok user who shares 20 videos each day. One billion people are using it right now. In this setting, semi-supervised learning has several potential applications, including but not limited to Web content classification, Text document classification, and Speech recognition, as illustrated in **Table 2**.

Table 2. Applications of Semi-Supervised Learning

Illustration of Semi-Supervised Learning applications	
Speech recognition	In order to overcome the difficulties and improve performance, semi-supervised learning may be employed while labeling audio. Semi-supervised learning, namely the self-training approach, has been effectively deployed by Facebook (now Meta) to its voice recognition models, resulting in better performance. The next step was to use the pre-trained model, which was developed using one hundred hours of annotated audio data. As a further step, 500 hours of unstructured speech samples were added, and the models' accuracy was improved by self-training. The findings show a drastic improvement, with the Word Error Rate (WER) dropping by 33.9%.
Text document classification	Constructing a classifier for documents written in a natural language is another scenario where semi-supervised learning has proven useful. Because it is so time-consuming for human annotators to go through many, wordy texts in order to classify them into broad categories like "type" or "genre," this technique is particularly well-suited to this task.
Web content classification	There are billions of websites out there displaying different kinds of material; therefore it would require a massive effort to classify the web by manually adding labels to each page. Semi-supervised learning is used in its many forms to annotate and properly categorize online material in order to enhance the user experience. Multiple search engines, including Google's, include SSL as a component of their ranking system because it helps them better understand human language and the relevance of prospective search results to requests. Use of Secure Sockets Layer (SSL) by Google Search helps the search engine provide results that are more relevant to a user's search.

Classifiers may be constructed on top of Deep Neural Networks (DNN) e.g., Long Short-Term Memory (LSTM) networks, which are adept at discovering long-term relationships in data and retraining old knowledge. In most cases, a larger amount of unlabeled and labeled data is needed to successfully train a neural network. A semi-supervised learning approach is enough since it only requires training a basic LSTM model on a small number of text examples with manually annotated most relevant terms, and then applying that model to a much larger number of unlabeled data. In this approach, we are provided with m randomly distributed instances of class X $(x_1, x_2, x_3, \ldots\ldots x_n)$ and their associated labels $(y_1, y_2, y_3, \ldots\ldots y_n)$. In addition, there are n instances of X without labels $(x_{m+1}, x_{m+2}, \ldots\ldots x_{n+m})$. There is less work involved in semi-supervised learning, but the results are more accurate. In this article, we'll go down the most essential semi-supervised approaches.

Yarowsky Bootstrapping Approach

The bootstrapping method was first implemented in 1995 by Yarowsky, who also made one of the most influential contributions to the field of Natural Language Processing. The Yarowsky approach is gradual and an example of a simple iterative algorithm that can function with just a limited number of examples from each sense [20]. Due to the fact that semi-supervised techniques rely on labeled examples, these labeled examples may be fed into other supervised methods for initial classifier training. After some first classifiers have been trained, they are utilized to mine the remaining untagged corpus for a bigger training set. Any trained sets that exceed a specified threshold are saved for use in subsequently training an entirely new collection of data. Re-labeling, retraining, and training are cycled through until no more changes can be seen. The main advantage of this approach is that it just needs a little amount of the initial training data to train a huge number of subsequent data sets. This approach allows humans to attain an unprecedented level of precision. Every time around the loop, new instances are incorporated to the training collection. As a result of these revisions, the recalls are enhanced.

Bilingual Bootstrapping Approach

Bilingual bootstrapping is a revolutionary method for classifying words in dictionaries as either synonyms or antonyms. To learn how to translate sentences, bilingual bootstrapping employs both unclassified dataset and a small amount of classified content in both the source and target languages. Data in both languages should come from the same domain, but they need not be presented in a synchronous fashion. Using the following two processes, it builds classifiers for both languages in parallel. To do this, we will first (1) use the classified data from both languages to build a classifier for each language, and then (2) use the classifiers we built to classify new data in each language, which will then be added to the language's existing classified data. For (1), we may make use of categorised data from both languages since terms from one language have equivalents in the other and vice versa. Increased classifier efficiency is achieved by mutual classification of labeled data and through sharing of labeled data across the two languages. There have been a number of experimental evaluations of the efficacy of bilingual bootstrapping for disambiguating translated words, and all of them have shown that it consistently and considerably beats monolingual bootstrapping. To achieve its better performance, bilingual bootstrapping makes strategic advantage of the imbalance between the ambiguous phrases of the two languages.

Label Propagation Approach

Vertices in a connected network stand for both labeled and unlabeled examples; the method works by continually disseminating labeled data from each vertex to surrounding vertices through weighted connections, and then contextualizing the labelling of unidentifiable samples after each iteration. The LP method makes use of weighted edges to disseminate the label data of a particular vertex in a network to its neighbors until a universal stable phase is achieved. If the edge weights are increased, labels may move more freely. This means that there is a stronger correlation between the labels and proximity (the global consistency assumption). With each iteration of the label propagation stage, the soft labels of the first labeled samples are typically clamped, allowing the labeled data to be used to restock label sources. Therefore, the labeled data function as sources, pushing labels into the surrounding unlabeled data. The weighted edges of the classes will be pushed through by the named instances, whereas the edges of the classes with lower weights will form gaps. With the right data structure in place, LP algorithms may utilize unlabeled data to aid in the development of a classification plane.

IV. UNSUPERVISED APPROACH

One form of machine learning algorithm, unsupervised learning is able to discern trends in data without the use of labels. The aim is to compel the computer to learn about its environment in a fashion similar to how humans do—through imitation—and then utilize that knowledge to fuel its own original thought processes. In comparison to supervised learning, which relies on human labels such as "ball" or "fish," unsupervised approaches demonstrate self-organization by encoding feature preferences in the machine's parameters and authorizations to capture trends as probability density functions. Semi-supervised learning, in which only a fraction of the information is categorized, and learning algorithm, in which the computers is given just a numerical performance indicator as input, are on the other end of the continuum from fully-supervised learning.

Differential (recognition) and generative tasks are two common ways of classifying work for neural networks (imagination). The line between supervised (used for classification) and unsupervised (used for generation) learning is blurry; see Venn diagram. Supervised learning is more effective for tasks like object identification, although unsupervised learning may be used to sort things into categories. As time goes on, certain tasks use both approaches, while others go back and forth between them. For instance, traditional supervised image detection has given way to hybrid approaches using unsupervised pre-training and then reverting to supervision through failure, relu, and tunable learning rates.

During training, an unsupervised network attempts to produce an output that is similar to the input data, and then it utilizes the error in that output to refine its own predictions (i.e. correct its biases and weights). A mistake may be described as a low probability of the false output or as an unpredictable high-energy state in the system, dependent on the setting. While backpropagation is used extensively in supervised learning, a wide variety of other techniques are used in unsupervised learning, such as the Boltzmann learning rule, the Hopfield learning rule, Wake Sleep, Variational Inference, Maximum A Posteriori, Contrastive Divergence, Maximum Likelihood, and Gibbs Sampling. More information is provided in **Table 3** below.

The unsupervised method to word meaning disambiguation makes use of un-annotated raw data and learns from the clusters of words that are produced on the basis of assumed similarities. These methods assume that adjacent words with the same meaning will also have some similarities. Some kind of context similarity metric is used to determine the intended meaning of a word. Word Sense Discrimination (WSD) is carried out by unsupervised WSD, and the technique splits all word occurrences into different categories by deciding whether or not two instances of the same word belong to the same meaning. However, there is more work involved in determining how effective these techniques are. Unsupervised techniques primarily focus on locating patterns of meaning.

When it comes to solving the knowledge acquisition bottleneck, unsupervised approaches are the way to go. When compared to other disambiguation methods, the performance of the unsupervised technique is always worse. One example of an unsupervised technique is the word-clustering method, in which words are grouped together in terms of how closely they resemble the target words oriented on the semantic similarity of a particular attribute (such as subject-verb, adjective-noun, etc.). Another method of context clustering involves finding terms that occur alongside the lexical word and then finding the center of that vector. An alternative approach uses a graph based on some linguistic link, and then assigns weights to each edge of the network depending on how closely they are connected. The target word is first iterated over to find the node with the greatest degree, and then the minimal spanning tree is used to separate out the possible meanings. There is a risk that cases in the training data will be misclassified if an unsupervised approach is used. Different kinds of things may be found in the same cluster. The number of clusters may not always correspond with the total number of possible meanings for the target term.

Table 3 illustrate the three tasks—clustering, dimensionality reduction, and association—, which are the most typical unsupervised learning application models

Table 3. Tasks in unsupervised learning application models

Illustration of clustering, dimensionality reduction, and association tasks	
Clustering	Clustering is a data mining technique for identifying similar sets of data that may be grouped together. For instance, K-means clustering algorithms classify data into groups based on their similarity, with K representing the group size and level of detail for image reduction, and market segmentation.
Dimensionality reduction	When a dataset has an excessive number of characteristics (or dimensions), a dimensionality reduction technique may be used to simplify the learning process. Tolerable reductions in data inputs are achieved without compromising data quality. This method is often used in data preparation, for instance when generative algorithms filter out irrelevant visual input to improve picture quality.
Association	Association is a type of unsupervised learning that uses different rules to investigate potential relationships between dataset parameters. Similar to "Customers Who Bought This Item Also Bought" suggestions, these approaches see widespread application in market basket analytics and recommendation systems.

Context Group Discrimination Approach

This approach, originally developed by McGarrity, Huebner, and McKinnon [21], takes it one step further by distinguishing between the meanings of words after context vectors have been generated. The purpose of this method is to categorize the ambiguous words in the corpus by their meanings. The method uses a multi-dimensional real-valued vector space to represent sensations, words, and context. Similarities in their contexts are used to group the events together. While the cosine function is still used to identify contextually comparable instances, the clustering is carried out using the iterative, probabilistic modeling for maximum likelihood prediction known as the Expectation Maximization method. After the ambiguous words have been identified, the circumstances of each occurrence of them are recorded as context vectors, and a technique known as mean agglomerative clustering is utilized in the sense acquisition phase. Words are compared using a formula that takes into account how many neighbors they share. The more semantically similar the two

sets of terms are, the closer the two sets of meanings are. After that is done, the occurrences are clustered according to their shared contexts. Structural semantic interconnections take a very similar technique (hybrid algorithm).

Using a method known as context-group discrimination, the typically term of ambiguous term may be deduced from a cluster of usages that share comparable contexts. For the specific implementation discussed here, high-dimensional, actual-valued vector spaces are used. All representations in context-group discrimination come from a massive text corpus, making it a corpus-based approach. **Fig. 6** depicts the architecture necessary for context-group discrimination. The training set uses the ambiguous term to assign a coordinate in Word Space to each occurrence of the word (revealed by way of a single illustration: To illustrate, the dashed line connecting the instructional material and Word Space). In order to create the map, we first consult Word Space for word vectors (mentioned below). All of the training-text scenarios are transferred to Word Space, and then the resultant point cloud is segmented such that units inside a subdivision are near together and divisions are as far apart as feasible.

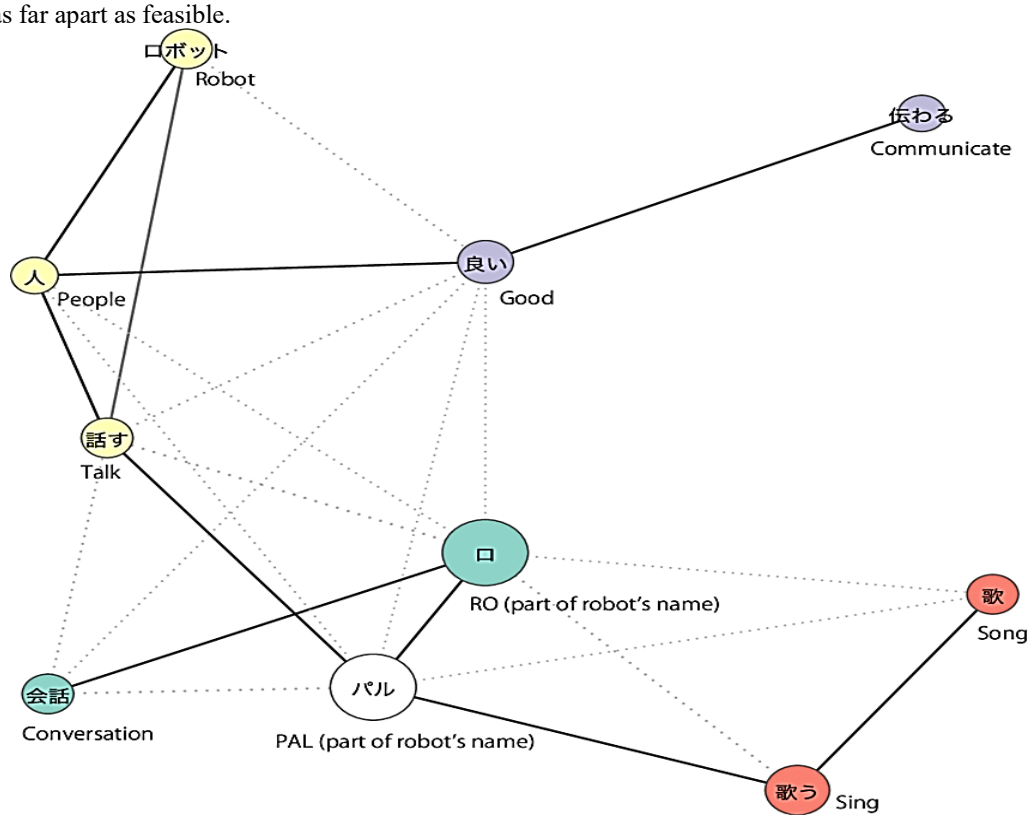

Fig 5. Acquired metadata co-occurrence network graph.

Dotted lines indicate the boundaries of the generated clusters. It is expected that the meanings of the ambiguous term may be grouped together and that each cluster corresponds to a distinct meaning (an assumption that will be put to the test in the future). The square centroid of each cluster stands in as its representation. When the confusing word comes in a new context (in the picture labeled "test setting") after training, the problem is addressed by mapping the new context onto Word Space (circle representing the place below the dashed line). Then, a group is picked based on its centroid's proximity to the context's centroid (solid arrow). Once this sense has been determined, the situation may be classified as an instance of its usage. We need to model three different kinds of entities: Senses, Contexts, Words. Senses, Contexts, Words are all represented by vectors. Consideration of the surrounding words in the corpus is used to generate word vectors, context vectors are produces from sense vectors and word vectors are generated by grouping the dispersion of contextual vectors.

Since vector spaces are widely used for representation in IR, we decided to employ them. Perhaps the most ubiquitous framework in IR is the vector space model. Many comparative studies of IR performance have placed systems based on it in the top ranks. In light of this, we propose using the vector-space paradigm for the encoding of words. To mimic the prevalent practice of representing documents and queries in IR as vectors with a single dimension each, we map words to this vector space. There is still another method of calculating Word similarity is a dimensional representation of words, where each dimension represents a document. Since there are fewer occurrence-in-document occurrences than word co-occurrence instances, these word models are often sparser and, therefore, less useful than word-based modelling. Also, dictionaries and hand-encoded characteristics have been used to create word vectors. The proposed corpus-based strategy avoids the drawbacks of both a general dictionary and a subject-specific book by relying only on machine learning (for example, on chemistry).

Finally, structural information such as head-modifier relationships may be used to calculate word similarity. In contrast to structure-based representations, document-based representations, and co-occurrence-based representations are sparser.

Whether structural aspects or associational traits are more illuminating is up for debate. Vinh [22] have suggested methods of word representation that are conceptually similar to our own. Vinh's method replaces traditional co-occurrence counts with mutual information scores between the word being represented and the dimension words as the vector entries. In this section, we'll go through the techniques used to derive a vector and to distinguish between senses.

Co-occurrence Graphs Approach
Instead of using vectors, as was done in the past, algorithms in this area employ graphs to depict the words. The text is transformed into a graph in which each word is a vertex and the syntactic relationships between them are edges. The graphs are built from the context units (such as metadata in **Fig. 5**) that include the target words.

Hyperlex
If the target word appears with other words in the same paragraph, then those words become vertices and the edge connecting them to the target word equals the sequence the two words appear together in the same paragraph. The edge weights decrease as the recurrence of these words together increases.

WSD using Parallel Corpora
Words with various meanings in one language have been shown to have separate translations in other languages via experimental means. Shen and Wang [23] use this assumption in a disambiguation method. The system was developed to efficiently and automatically annotate a huge corpus of sense marks. The algorithm requires multilingual input corpora to do its task (hence known as parallel corpora).

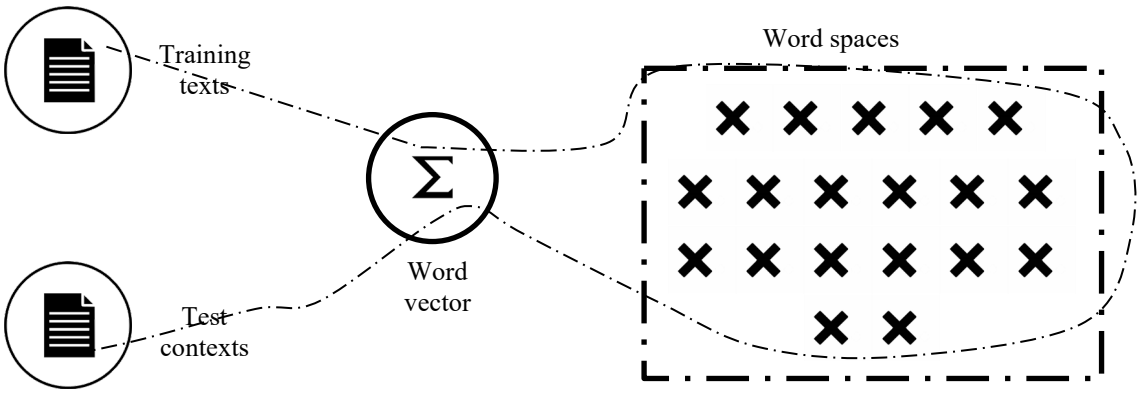

Fig 6 The foundational architecture of context-based grouping.

The training set contexts of the ambiguous word are converted into Word Space context vectors (upper dotted arrow) by incorporating the word vectors within the contexts. Sense vectors are the representation of the concentricity of a cluster of context vectors (shown by the dotted lines) (squares). An ambiguous word's ("test context") surrounding text is mapped to a context vector in Word Space, where it may be more easily distinguished (lower dashed arrow ending in circle). To determine which sense is most appropriate for a given situation, we look for the vector that is closest to it (solid arrow).

Word vectors and sense vectors are used in context-group discrimination to separate applications of the ambiguous word. The ambiguous word v occurs at time t in the following ways: (i) In Word Space, map t to the vector representation using the values of the lexical words (the lower dashed line in **Fig. 6**). To do (ii), amass all *j* sense vector of v. (the squares in the illustration that represent). (iii) Assign the specified value to whichever sense whose sense vectors is nearest to t. (a definitive arrow indicating the designated task). This method selects the context grouping whose sense vectors are still most comparable to the sense vector of the frequency of the word in order to remove any ambiguity. Semantic features of a given context or sensations are captured by the matching context vector or sense vector. Therefore, the sense vectors with the highest similarity degrees to the context vector are the most accurate interpretation of the situation. Therefore, the event is classified as belonging to that sense by context-group discrimination.

V. CONCLUSION

Machine Learning (ML) is an Artificial Intelligence (AI) subfield that enables machines to "improve" and "learn" themselves with little to no human guidance. Learning (model fitting) requires humans to have some facts or data (also labeled samples or cases) at our disposal so that we may investigate the possibility of underlying patterns, embedded in our data. These behaviours are only a set of functions or a set of rules for making choices. Models that are trained on a labeled dataset may then be utilized to make predictions. The training technique takes as input a labeled dataset, and the output of the learning algorithm is an inferred function that can be used to generate predictions about fresh observations that have not been observed before. After enough training, the model can generate targets for any novel input. In order to compensate for mistakes, the learning algorithm may additionally compare its results to the right result (ground truth label) (e.g. via back-propagation). In this research, we analyzed the most prominent techniques for Word Sense Disambiguation

(WSD), including semi-supervised, unsupervised and supervised learning methodologies. For those interested in Neuro Linguistic Programming (NLP), WSD is a crucial area to explore. The aforementioned methods are effective in resolving WSD. When compared to semi-supervised and unsupervised methods, the supervised method is determined to be more efficient due to the nature of the data it uses.

References

[1]. P. Durgaprasad, K. V. N. Sunitha, and B. Padmajarani, "Resolving lexical level ambiguity: Word sense disambiguation for Telugu language by exploiting IndicBERT embeddings," in Communication, Software and Networks, Singapore: Springer Nature Singapore, 2023, pp. 357–368.

[2]. W. Jooste, R. Haque, and A. Way, "Philipp Koehn: Neural Machine Translation: Cambridge university press, 30 jun 2020, www.cambridge.org/9781108497329, DOI: 10.1017/9781108608480," Mach. Transl., vol. 35, no. 2, pp. 289–299, 2021.

[3]. G. Wang, J. Ma, Y. Wang, T. Tao, G. Ren, and H. Zhu, "SUDF-RS: A new foreign exchange rate prediction method considering the complementarity of supervised and unsupervised deep representation features," Expert Syst. Appl., vol. 214, no. 119152, p. 119152, 2023.

[4]. "Reverse engineering biological networks. Opportunities and challenges in computational methods for pathway inference. Proceedings of the workshop entitled Dialogue on Reverse Engineering Assessment and Methods (DREAM). September 7-8, 2006. Bronx, New York, USA," Ann. N. Y. Acad. Sci., vol. 1115, pp. xi–xiv, 1–285, 2007.

[5]. A. J. Hill, R. S. Schumacher, and I. Jirak, "A new paradigm for medium-range severe weather forecasts: probabilistic random forest-based predictions," arXiv [physics.ao-ph], 2022.

[6]. N. Wani and K. Raza, "MKL-GRNI: A parallel multiple kernel learning approach for supervised inference of large-scale gene regulatory networks," PeerJ Comput. Sci., vol. 7, no. e363, p. e363, 2021.

[7]. T. Bellotti, R. Matousek, and C. Stewart, "A note comparing support vector machines and ordered choice models' predictions of international banks' ratings," Decis. Support Syst., vol. 51, no. 3, pp. 682–687, 2011.

[8]. J. C. Alejandrino, J. P. Bolacoy Jr, and J. V. B. Murcia, "Supervised and unsupervised data mining approaches in loan default prediction," Int. J. Electr. Comput. Eng. (IJECE), vol. 13, no. 2, p. 1837, 2023.

[9]. Á. Sashalmi, "A geopolitikai elemzés a nemzetközi hatalmi viszonyok vizsgálatának szolgálatában : Csurgai Gyula: Geopolitical Analysis. A Multidimensional Approach to Analyze Power Rivalries in International Relations. Róma: Aracne, 2019. 208 o," Külü. szle., vol. 20, no. 3, pp. 282–290, 2021.

[10]. I. Park, P. D. Windschitl, J. E. Miller, A. R. Smith, J. O. Stuart, and M. Biangmano, "People express more bias in their predictions than in their likelihood judgments," J. Exp. Psychol. Gen., 2022.

[11]. D. Wilmot, "Great Expectations: Unsupervised inference of suspense, surprise and salience in storytelling," arXiv [cs.CL], 2022.

[12]. M. A. Miller, X.-J. Feng, G. Li, and H. A. Rabitz, "Identifying biological network structure, predicting network behavior, and classifying network state with High Dimensional Model Representation (HDMR)," PLoS One, vol. 7, no. 6, p. e37664, 2012.

[13]. T. Van den Bulcke et al., "SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms," BMC Bioinformatics, vol. 7, no. 1, p. 43, 2006.

[14]. M. Altenbuchinger, A. Weihs, J. Quackenbush, H. J. Grabe, and H. U. Zacharias, "Gaussian and Mixed Graphical Models as (multi-)omics data analysis tools," Biochim. Biophys. Acta Gene Regul. Mech., vol. 1863, no. 6, p. 194418, 2020.

[15]. B. Klinger and N. Blüthgen, "Reverse engineering gene regulatory networks by modular response analysis – a benchmark," Essays Biochem., vol. 62, no. 4, pp. 535–547, 2018.

[16]. L. Özbek, "An adaptive extended Kalman filtering approach to nonlinear dynamic gene regulatory networks via short gene expression time series," Commun. Fac. Sci. Univ. Ank. Ser. A1 Math. Stat., vol. 69, no. 2, pp. 211–220, 2020.

[17]. E. Hairer and C. Lubich, "Numerical analysis of ordinary differential equations," in Encyclopedia of Applied and Computational Mathematics, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1053–1059.

[18]. I. Umoren, E. Polycarp, and G. Ansa, "Spectrum scheduling classification using conditional probability and a decision tree supervised learning approach," 2022.

[19]. J. Han, T. Zhang, Y. Li, and Z. Liu, "RD-NMSVM: neural mapping support vector machine based on parameter regularization and knowledge distillation," Int. j. mach. learn. cybern., vol. 13, no. 9, pp. 2785–2798, 2022.

[20]. D. Rao, N. Garera, and D. Yarowsky, "JHU1: An unsupervised approach to person name disambiguation using web snippets," in Proceedings of the 4th International Workshop on Semantic Evaluations - SemEval '07, 2007.

[21]. L. A. McGarrity, D. M. Huebner, and R. K. McKinnon, "Putting stigma in context: Do perceptions of group stigma interact with personally experienced discrimination to predict mental health?," Group Process. Intergroup Relat., vol. 16, no. 6, pp. 684–698, 2013.

[22]. L. A. Vinh, "The number of occurrences of a fixed spread among n directions in vector spaces over finite fields," Graphs Comb., vol. 29, no. 6, pp. 1943–1949, 2013.

[23]. Q. Shen and Q. Wang, "Phase difference based Doppler disambiguation method for TDM-MIMOFMCW radars," arXiv [eess.SP], 2022.