Improvement of Translation Accuracy for the Word Sense Disambiguation System using Novel Classifier Approach

Ajith Abraham
School of Artificial Intelligence,
Bennett University, Greater Noida,
Uttar Pradesh, India
abraham.ajith@gmail.com

Bineet Kumar Gupta
Machine Intelligence Research Labs,
Scientific Network for Innovation and
Research Excellence, Auburn, USA
bkguptacs@gmail.com

Satya Bhushan Verma
Shri Ramswaroop Memorial University,
Barabanki, India
satyabverma1@gmail.com

Archana Sachindeo Maurya
Shri Ramswaroop Memorial
University, Barabanki, India
archanamaurya2308@gmail.com

Mohammad Husain
Islamic University of Madinah Al Madinah
Al Munawarah, Saudi Arabia
dr.husain@iu.edu.sa

Arshad Ali
Islamic University of Madinah Al Madinah
Al Munawarah, Saudi Arabia
a.ali@iu.edu.sa

Sami Alshmrany
Islamic University of Madinah Al Madinah Al Munawarah
Saudi Arabia
s.alshmrany@iu.edu.sa

Sanjay Gupta
HCL Technologies, India
sanjay_gupta@hcl.com

Abstract: *Machine Translation (MT) is a crucial application of Natural Language Processing (NLP). This MT technique automatic and based on computers. One of the most modern techniques adopted in MT is Machine Learning (ML). Over the past few years, ML has grown in popularity during MT process among researchers. Ambiguity is a major challenge in MT. Word Sense Disambiguation (WSD) is a common technique for solving the ambiguity problem. ML approaches are commonly used for the WSD techniques and are used for training and testing purposes. The outcome prediction of the test data gives encouraging results. Text classification is one of the most significant techniques for resolving the WSD. In this paper, we have analyzed some common supervised ML text classification algorithms and also proposed a "hybrid model" called "AmbiF." We have compared the results of all analyzed algorithms with the proposed model "AmbiF. The analyzed supervised algorithms are Decision Tree (DT), Bayesian network, Support Vector Machines (SVMs), K-Nearest Neighbor (KNN), Random Forest (RF), and Logistic Regression (LR). The range of accuracy for all the algorithms that were examined is between sixty-eight and eighty-four percent. To improve the accuracy of the AmbiF model, we have merged the DT, SVM, and Naïve Bayes (NB)-classifier approach. For testing the model, we have used the ten-fold cross-validation test method. The AmbiF model's accuracy has been reported eighty-five percent. Comparing the AmbiF model to all other analyzed supervised ML classification algorithms, it has also demonstrated great precision, recall, and F-score.* Waikato Environment for Knowledge Analysis *(WEKA)'s ML-tool is used to analyze the algorithms and the AmbiF model.*

Keywords: *Bayes theorem, machine learning, machine translation, naïve bayes classification, supervised approach, unsupervised approach, and word sense disambiguation.*

Received November 4, 2023; accepted September 18, 2024
https://doi.org/10.34028/iajit/21/6/14

I. Introduction

Since the beginning of the time, humans have used a variety of machines to complete various operations more quickly. These machines made life easier and faster for humans by completing their demands in all areas. The idea of human and computer interfaces first appeared in the 1980s, after which, it has developed quickly [19]. Machine Translation (MT) is one of them. Machine Learning (ML) plays a vital role in MT. ML is the rapidly growing sub field of Artificial Intelligence (AI) and is designed to develop the human intelligence by learning from the environment. ML algorithms are helpful to make the computers to learn. ML is currently being actively used in all aspects of daily life. The term ML was first used by Samuel [31]. According to him term ML is defined as "ML is a sub field of AI that trains computers to become independent and also enhancing the learning process from the experiences without any human interaction." Web-based search engines, photo labelling applications, junk mail detectors, MT etc., are only a few examples of real-world applications where ML is helpful.

1.1. Applications of Machine Learning

Some of the most popular ML in Natural Language Processing (NLP) include Speech and Language

Recognition (SLR), Information Extraction (IE), Information Retrieval (IR), and MT. Among these applications the most significant is MT, which uses ML algorithms to translate speech and text from source language to target language. Figure 1 displays the various ML applications.

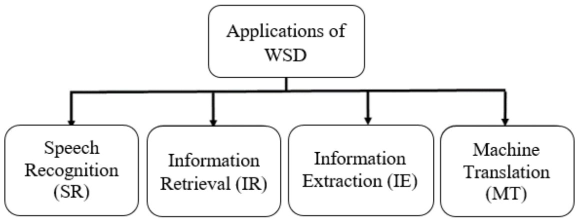

Figure 1. Applications of ML.

1. Speech Recognition (SR)

Speech is the primary form of human interaction. With the use of SR, oral words can be converted into the written text [29]. Figure 2 shows the SR process.

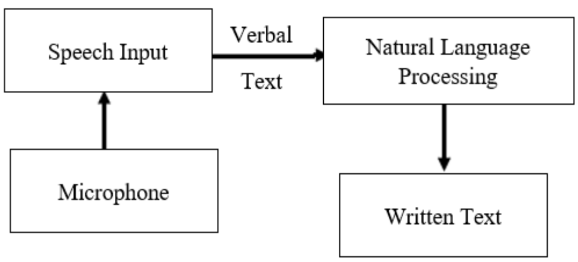

Figure 2. Process of SR.

The primary benefit of SR technology is the speed at which papers can be written. This technique is faster than any human being could type.

2. Information Retrieval (IR).

The information that is most important to the user's query is organized, stored, recovered, and estimated by an IR system. A user sends an enquiry in the form of a request that is built in natural languages whenever he needs any information. As Lancaster (1979), indicated that: "An IR system does not provide the user with information on the topic of their query. It just provides information about the availability and location of documents related to their query" [36]. The word documents also contain non-textual elements like graphics and audio. Figure 3 depicts the fundamentals process of IR procedure.

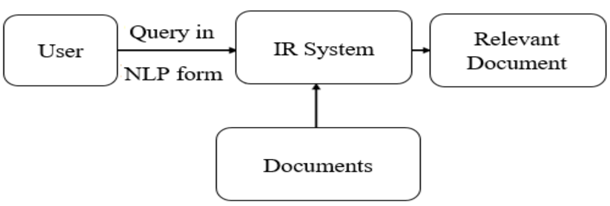

Figure 3. Process of IR system.

After processing on raw data, the IR process always provides the information of the user's request.

3. Information Extraction (IE).

The amount of raw data on the World Wide Web is vast. There is enough information in the data to answer every given inquiry. Accessing pertinent information is made possible by a number of information processing tools, including IE, text summarization, and question-answering software. The IE system provides an output as an overview after receiving raw data as input. It provides the slots that the retrieval system offers.

Example: "This morning I have seen a fast running white colored bullet."

Extracted Information are:

- Colour: white.
- Object name: bike.
- Model: bullet.
- Speed: fast.
- Day: today.
- Time: morning.

An IE system can be made through the use of an engineering process based on the knowledge and rules. It can also be produced through the training process. The strategy that performs best when all of the resources related to linguistics and human specialists are accessible is knowledge-based engineering. Different training algorithms can apply to the available training data set. For solving the ambiguity problem, we can extract the information from the semantic analysis [30].

4. Machine Translation (MT).

NLP applications heavily rely on MT. MT is the process of translating text automatically from one language into another without human involvement. In Figure 4, the translation process is shown.

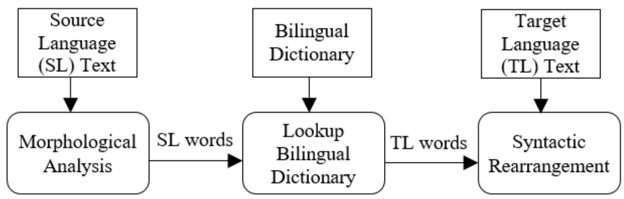

Figure 4. MT process.

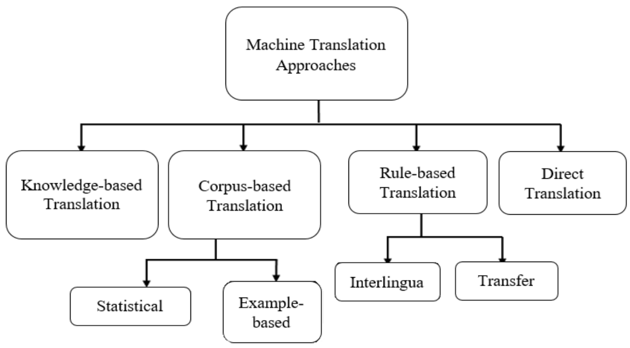

Figure 5. Classification of MT approaches.

In the multilingual nation of India, just three percent of the population can understand English [34].

Therefore, a MT system is required in order to convert any information from one language to other languages. The MT system can be categorized according to the internal process of translation. Based on the different methodologies, the MT system can be broadly divided into four categories, which is shown in Figure 5.

Each MT strategy has its own advantages and disadvantages. Today, the most common method involves the testing of various combinations. Obtaining a translation free of errors is the main objective of the MT process. Finding a translation that is 100 percent accurate is more difficult due to some structural and stylistic differences in different languages. These differences includes word sense, idioms, word order, pronoun resolution, and ambiguity. Figure 6 illustrate the structural and stylistic variations and the detailed explanation of structural and stylistic differences is shown in Table 1.

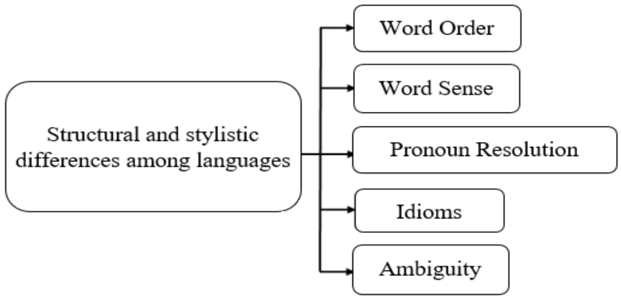

Figure 6. Differences in structure and style that occurs during the MT process.

Table 1. Dissimilarities in structure and style elements that appear during the MT processes.

S. No.	Problem	Explanation	Example
1.	**Order of words**	As a result, translation is more challenging because order of words differ from language to language. As an illustration, we can observe that sentences in Sanskrit are structured with subjects, objects, and verbs order while sentences in English are structured with subjects, verbs, and objects order.	Sentences in the English language follows Subject-Verb-Object order while sentences in Sanskrit language go along with Subject-Object-Verb order. 1. Sita (S) reads (V) book (O). (English sentence) 2. सीता (S) पुस्तकं(O) पठति(V). (Sanskrit sentence)
2.	**Sense of words**	It is possible to translate a word's meaning in one language into a different meaning in the target language. This difficulty may significantly complicate the choice of significant meaning in the target language.	In sentence "This is a big crane." While translating from English to Sanskrit the word "crane" it have two meanings: 1. "यन्त्रः " 2. "खगः"
3.	**Resolution of pronoun**	For a translation to be accurate, the prepositional phrases term must be resolved correctly. A prepositional phrases reference connects a previously described object to another text portion of the sentence	Consider the sentence "Hari is a businessman. He has a white car." In this sentence subject "Hari" is connected to the pronoun "he."
4.	**Idioms**	The idioms have indirect meanings and used in sentences, Idiomatic phrases can be changed it their native language equivalents.	Consider the following idiom: "Better late than never" Its meaning is "Better to arrive late than not to come at all" that cannot be directly translated.
5.	**Ambiguity**	Those terms in the target language, that have multiple meanings are referred to as ambiguous words. This issue is termed as ambiguity problem. This ambiguity issue has to be resolved before translation.	In sentence "This well is dry and deep" Well is an ambiguous word in this sentence, and when translating it into the Sanskrit language, we must find a way to make it clearer because Sanskrit gives i.e., "कूपः" and "कुशलम्" meanings.

1.2. Ambiguity: A Major Challenge during Process of Machine Translation

Almost all main languages contain numerous terms that can imply multiple meanings depending on the situation. These words are classified as "ambiguous words" while their presence is termed to as "ambiguity" [13]. Approximately, all natural languages suffers from the ambiguity challenge. Every languages have various ambiguous words, which can have different meanings depending on the Parts-Of-Speech (POS). Ambiguity is the term used to describe the situation where a word or phrase in a sentence has more than one meaning or interpretation. There are many words that have different meanings and can be interpreted in numerous ways to mean in different things. The English phrases "I am well" and "This well is dry" for instance, could be translated into Sanskrit as "in a good or satisfactory way" (कुशलम्) or "a shaft sunk into the ground to obtain water, oil, or gas" (कूपः) depending on the context. The ambiguity can be grouped into two categories i.e. word-level ambiguity and sentence-level ambiguity. This classification is shown in Figure 7.

Following a review of the relevant literature, we may categorize ambiguity into the following five groups, which are shown in Figure 8 and each sorts of ambiguity is described and thoroughly discussed with an example in Table 2.

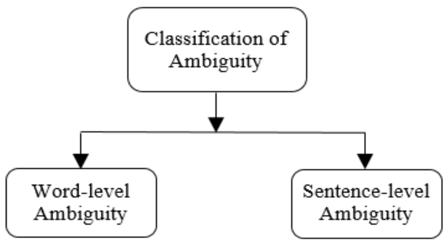

Figure 7. Different levels of ambiguity.

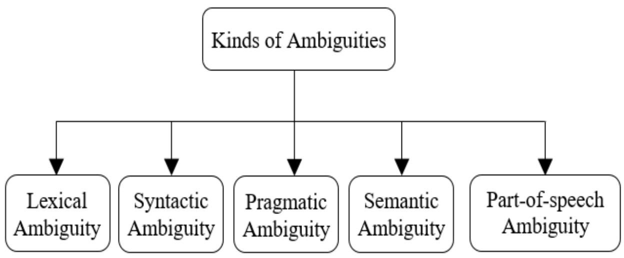

Figure 8. Various kinds of ambiguities.

Table 2. Various kinds of ambiguities, with explanations and examples.

S. No.	Kinds of ambiguity	Description	Example	Explanation
1.	Lexical ambiguity	Lexical Ambiguity suggests that certain words may have more than one meaning.	"The fisherman went to the bank"	In this sentence The word "bank refers to a "financial institution" or a "region of land near the river" in this statement.
2.	Semantic ambiguity	A sentence is said to be semantically ambiguous when there are several possible interpretation.	"Ram kissed his son, and so did John"	Interpretation 1: "Ram kissed his son and also John kissed to Ram's son." Interpretation 2: "Ram kissed to his son and also John kissed to his son."
3.	Pragmatic ambiguity	Pragmatic ambiguity suggests that a sentence has more than one meaning.	"Do you know what time is it?"	This sentence has multiple meanings. a) Inquiring about the current time. b) Expressing anger to someone who failed to do the work on due time.
4.	Syntactic ambiguity	When more than two interpretations of a statement may be made, there is syntactic ambiguity.	"Both the old man and woman are watching the TV"	In the given sentence there are two interpretation made by the word old i.e. Adjective old is attached with the man only or with the both man and woman.
5.	POS ambiguity	When a single word can be used as a noun, verb, adjective, adverb, etc. as well as another part of speech, this is known as part-of-speech ambiguity	"This present is very beautiful	The "present" has two meanings in the given example, each corresponding to a distinct part of speech: a) First is the present is a noun which means "a gift" or "the period of time now occurring." b) Second is the adjective that implies "in a particular place."

1.3. Ambiguity: AMajor Challenge in the English Language

All Indian languages have ambiguity of various kinds. Because English is the mother language in many nations, it is widely believed to be the most important language in the world. The term "ambiguous words" refers to the many words in the English language that have multiple connotations in various contexts. When used with adjectives, verbs, or nouns, present or past participles in English invariably lead to part-of-speech ambiguity [27, 33]. These ambiguous terms must be appropriately disambiguated for the translation to be accurate in the target language when we translate from one natural language to another. Some ambiguous words and their meaning with respect to their corresponding POS is shown in Figure 9.

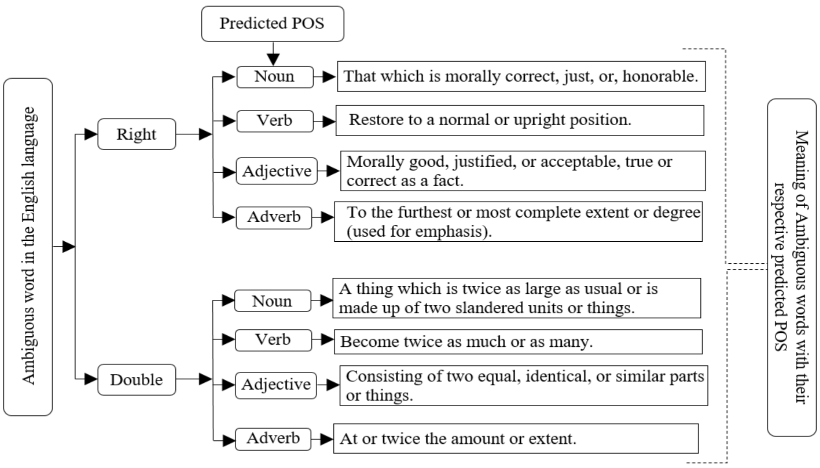

Figure 9. Few English ambiguous words and senses with relation to their respective POS.

1.4. Ambiguity: Resolution and Word Sense Disambiguation (WSD)

Document can be translated from the source dialect to the target dialect using MT. Different types of ambiguity have been observed in almost all natural languages. These ambiguous words must be properly decoded in order to be translated into the correct target language when we translate from one language to another. The disambiguation procedure can be used to resolve the ambiguity issue. The most significant application in which we employ WSD methods for the elimination of various forms of ambiguity is MT [5]. As a result, WSD is the disambiguation technique for determining a word's precise meaning. Word sense determination, often known as WSD. Finding the precise meaning and appropriate sense of an ambiguous word in a sentence is a process known as WSD. This method identifies the precise meaning of a sentence when as stem refers to a word having several meanings. Because of this, ambiguous words can have multiple meanings, and the WSD approach can assist in determining the precise meaning of a word [25]. For WSD processes, a variety of strategies can be used. Knowledge-based, supervised learning, and unsupervised learning methodologies are among them. The supervised learning methodologies have been covered in this work. On the WEKA tool, all algorithms are assessed and analyzed. Also each algorithm's classification accuracy is reported. Figure 10 displays these techniques.

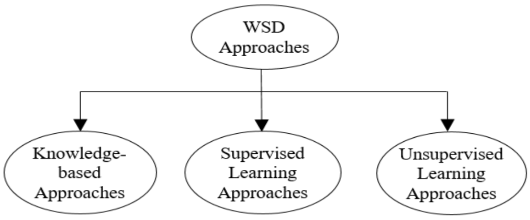

Figure 10. Various kinds of WSD approaches.

1.5. WEKA: An Open-Source Machine Learning Tool

All the ML approaches are analyzed and run on the WEKA tool. The WEKA is abbreviated as "Waikato Environment for Knowledge Analysis." This analysis tool was created by the University of Waikato in New Zealand. It is free software that offers tools for preparing data, implementing ML algorithms, and visualizing results. It enables us to create ML models and apply them to real-life disambiguation problems.

It can be found on the World Wide Web for free. Because this software is written in Java, it can be ported and used on a variety of platforms. WEKA has both a command line and a graphical user interface. The explorer and the experimenter are the two main categories of its operation [7].

- We can pre-process a dataset using the explorer, run it through a learning algorithm, and evaluate the develop classifier. A specific kind of learning approach is a classifier.
- The experimenter allows us to combine multiple learners and compare their performance so we can select one approach for prediction.

Figure 11. Screenshot of WEKA ML-tool.

WEKA's most important and valuable feature is the implementation of learning models. Filters, which are tools for preparing data, are another useful element of WEKA. WEKA's primary focus is on classifier and filter methods. Implementations of methods for learning association rules and clustering data without a class value are also included. The dataset to which the algorithms will be applied must be in the Attribute Related File Format (ARFF) format. All of the standard text classification tasks are covered in this tool, including classification, clustering, and association, regression, and attribute selection. These tasks require a huge amount of data visualization and preparation. Various data visualization and preprocessing tasks are available. Each algorithm and technique uses a single relational table as input, which can be read from a file or generated through a database request [18]. The screenshot of the WEKA tool is shown in Figure 11.

2. Organization of the Paper

This paper is organized as follows: In section 2, Related Work. Section 3, we have explained and covered the various supervised ML approaches. In section 4, we have suggested our proposed model AmbiF along with its datasets, functionalities, and testing methodologies. In section 5, we have talked about the detailed accuracy and outcome and discussion of the research. In section 6, finally, we have covered the study's conclusion and its direction for future.

3. Related Work

The majority of the research on categorizing texts into distinct categories, including text classification, categorization, and routing, is founded on the same kind of bag-of-words representations that are employed in IR. There have been no successful attempts to increase performance by adding word sense data into text representations. However, the study use of the Reuters news collection has demonstrated the importance of topical domains and the potency of a WSD method to document categorization. A further study has shown that improving performance by including features in document representations for text document classification yields competitive outcomes on the Reuters-21578 and OHSUMED datasets. These developments have been facilitated by the incorporation of synonyms as well as the recognition of multiword phrases. More progress can be made by encompassing concepts. In work that is closely related to this, [15] showed improvements in document clustering. Naïve Bayes (NB) classifier was used to classify heart diseases. Experiments have been conducted on the coronary heart disease dataset to analyze its effects and outcomes. It claims to have 100% accuracy and outperforms NB [16]. Neural approach is used to develop a text classifier that is based on the learning Vector Quantization algorithm. It is a classification method that uses a competitive supervised learning algorithm [22].

4. Supervised Machine Learning Approaches

There are three commonly used ML approaches that we can use for the correct classification of the document. These approaches are the knowledge-based approach, the supervised learning approach, and the unsupervised learning approach. Functioning of ML approaches are shown in Figure 12.

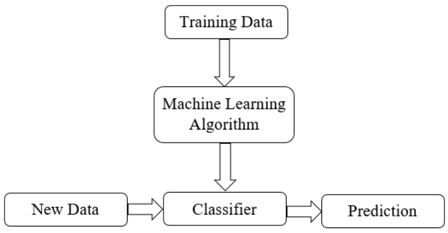

Figure 12. Working of ML approaches.

Among these three approaches, supervised approach is the most commonly used and is a probability based approach. These supervised ML algorithms can help the computers correctly classify the sentences on the basis of training provided using training data and extracted features. These algorithms are useful for determining the right meaning of an ambiguous word during the WSD process [23]. In Figure 13, commonly used supervised ML techniques are displayed.

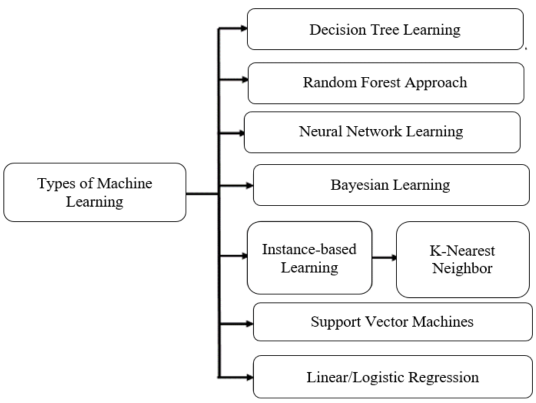

Figure 13. Classification of supervised ML approaches.

4.1. Decision Tree

One of the most significant and popular supervised learning algorithms is Decision Tree (DT) learning. It is a tree-based ML technique. These trees can be utilized to make judgments and put those decisions into action by implementing systems. A common type of data structure for storing information for searching and sorting operations is the tree. These trees can also be used to make judgments and put systems into place depending on those decisions. The following features can define a binary DTs. To make speedy decisions, these binary trees are used. A DTis a predictor, h: X Y that moves from the root node of a tree to a leaf in order to forecast the label associated with an instance x. For the sake of simplicity, we concentrate on the binary classification context, that is, Y=0, 1, however DTs can also be used for other prediction issues. The successor child is selected based on a partitioning of the input space at each node along the root-to-leaf path. Most of the time, the splitting is determined by one of x's characteristics or by predetermined set of splitting rules. A distinctive label is present on each leaf.

Example of DT: building a DT for the fruit classification is shown in Figure 14.

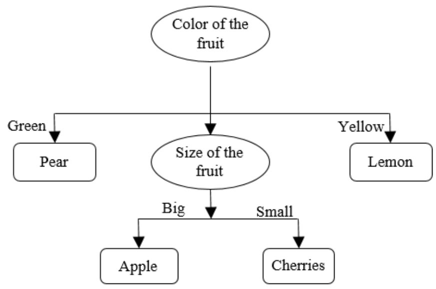

Figure 14. Example of a DT.

In Figure 13, a DT for the fruit classification is shown. To check the category of a given fruit, the DT first examines the color of the fruit. It color is green or yellow, then the tree immediately predicts that the fruit is "pear" and "lemon" respectively. If the color s red then it further examines the size of the fruit. If the size is small then it is cherry otherwise fruit is an apple. We have to know the color and size of that fruit. If the color is red and size is big then it is an apple otherwise it is a cherry.

4.1.1. Sample Complexity

Thresholding the value of a particular characteristic is the foundation of a common splitting rule at the interior nodes of the tree. In other words, we shift to the node's right or left child depending on 1[xi<θ], where i[d] is the index of the pertinent feature and R is the threshold. In these scenarios, a DTmay be conceptualized as the division of the instance space, X=Rd, into cells, with each leaf of the tree denoting a single cell.

In blocks of size log2 (d+3) bits apiece, n+1 blocks will be used to depict a tree with n nodes. The last block signifies the completion of the code, with the preceding n blocks encoding the tree's nodes in a depth-first order (preorder). Each block shows if the current node is one of the following:

- A node with the internal type 1[xi=1] for some i[d]
- A leaf with a value of 1 and a leaf with a value of 0
- The code has ended.

Since there are d+3 alternatives overall, each block needs to be described using log2 (d+3) bits.

Considering that each internal node has two offspring, It is simple to demonstrate that the tree is being encoded without any prefixes and that the description length of a tree with n nodes is (n + 1) log2 (d+3).

4.2. Random Forest

One of the best and most widely used text classification technique is Random Forest (RF). This classifier uses an ensemble approach that groups different DTs together. In contrast to a single DT, a RF algorithm will have higher accuracy value. The same dataset's DTs can be used to construct a RF, but they cannot be correlated. The output of this procedure will be a tree that is built using the findings of various DTs [37]. Based on the characteristics or attributes of a new item, each DT generates a vote for its categorization, and the classification of the item is made by the tree with the highest number of votes. Figure 15 displays a graphic representation of the RF algorithm.

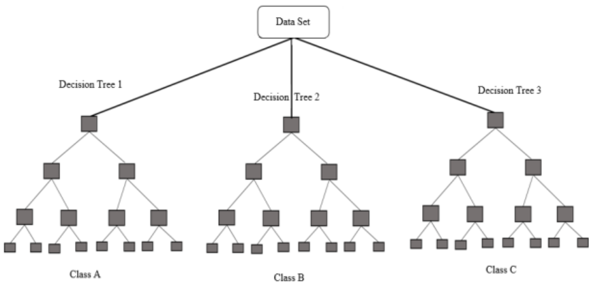

Figure 15. The RF algorithm is shown diagrammatically.

Compared to a single DT, this algorithm's accuracy is substantially higher while reducing the probability of overflow. This approach does not experience a time barrier because the DT operate in parallel. There is no need for normalization or scaling because this model offers excellent precision on a variety of topics. The problem of overfitting of the single DT can be reduce by constructing the ensembles of trees.

Breiman (2001) describe the RFs in the following ways: A RF is a type of classifier that consists of a group of DTs, each of which is built by using the algorithm A on the training set S plus an additional random vector θ, which is drawn from a distribution. A majority vote over the forecasts of the various trees results in the prediction of the RF. The algorithm A and the distribution over must be defined in order to identify a specific RF.

4.3. Naïve Bayes' Classifier

The supervised probability-based ML technique known as the NB classifier relies on the Bayes theorem. The majority of text categorization issues are solved with this classifier method. The training data-set contains a large number of variables that are all independent of one another. The term "features" refers to these independent factors. This method uses the Bayes' theorem to calculate the likelihood of specific qualities in a given class [4, 11, 35]. The highest value in this situation indicates the class that is the most "significant." The NB model is simple to use and effective for training high-dimensional datasets. It is straightforward, simple to use, and performs better than every other classification approach. To determine the probability value for each class, apply the formula below:

$$POS =(argmax)┬c\ P(c)∏\_(x=1)^m\ 〖P(fx/c) \tag{1}$$

$$POS=(argmax)┬c\ 〖P(c/f1,f2,………fx)〗 \tag{2}$$

$$POS=(argmax)┬c(f1,f2,……fx/c)P(c)/(P(f1,f2,…..fx)) \tag{3}$$

Here, POS represents the part of speech of the ambiguous word w, $f1$, $f2$, $f3$, ……. fx are the selected attributes, and x represents the total number of features extracted.

One of the widely used methods for classifying texts is the NB. This method is probability-based, offers a quantifiable way to assess how well other algorithms perform. The conditional probability of an event occurring is estimated in Bayesian learning, and that event is also related to certain other events [1, 24].

A well-known example of how generative assumptions and parameter estimations may make learning easier is the Naive Bayes classifier. Consider the issue of predicting a label y 0, 1 using a vector of attributes $x= (x1,..., xd)$, where we presumptively place each xi in the range of 0, 1. Recall that is the Bayes optimum classifier.

$$h\_Bayes\ (x) = argmax\ y0,1\ P[Y = y \mid X = x] \tag{4}$$

We require two-dimensional parameters, each of which corresponds to $P[Y=1|X=x]$ for a certain value of $x0$, $1d$, to characterise the probability function

$$P[Y = y|X = x] \tag{5}$$

In the Naive Bayes approach we make the (rather naive) generative assumption that given the label, the features are independent of each other. That is,

$$P[X=x \mid Y=y]=Y\ d\_i=1P[X\_i=X\_i|Y=y] \tag{6}$$

With this assumption and using Bayes' rule, the Bayes optimal classifier can be further simplified:

$$h\_Bayes\ (x)=argmax\ y\epsilon\{0,1\}P/Y=y/X=x \tag{7}$$

$$= argmaxy\in\{0,1\}P[Y = y]P[X = x|Y =y] \tag{8}$$

$$= argmax\ y\in\{0,1\}\ P[Y = y]\ Ydi=1P[Xi = xi|Y = y] \tag{9}$$

That is, now the number of parameters we need to estimate is only $2d+1$. Here, the generative assumption we made reduced significantly the number of parameters we need to learn. When we also estimate the parameters using the maximum likelihood principle, the resulting classifier is called the Naive Bayes classifier.

4.3.1. Bayes' Theorem

Bayes' Rule or Bayes' Law are various names for Bayes' Theorem. This theorem helps in calculating the conditional probability of any occurring event using prior probability. ML makes extensive use of the Bayes theorem [2, 38]. If there are two occurrences P and Q, then the following equation can be used to determine the Bayes' theorem formula:

$$P(P/Q)=(P(Q/P)*P(P))/(P(Q)) \tag{10}$$

Here, P and Q are the two occurring events.

$P(P|Q)$ is the posterior probability (P is an occurring event that occurs after the occurrence of event Q). $P(P)$ is the prior probability (this is the probability that is calculated before the occurrence of an event). $P(Q|P)$ is the Likelihood Probability of event Q that is calculated after the occurrence of evidence of an event *P* and the Marginal Probability is *P*(*Q*).

Bayes' rule is used to determine the conditional probability characteristics of a given class [3, 32]. The conditional probabilities of value for each term and attribute in a given sentence are calculated using this approach. The best outcome is produced by the highest value.

4.4. K-Nearest Neighbor (KNN)

The KNN approach is a simple but effective non-parametric grouping method [14]. In accordance with the votes of its k neighbors, this technique preserves all examples that are now available and groups new ones [39]. After classifying the unlabeled observations, KNN, a common statistical method, is used to classify the observations. The training dataset and testing dataset feature features of observations are gathered. Regression and classification problems can both be addressed by the technique. With the help of this technique, two crucial ideas can be implemented:

- One method relies on measuring the separation between two attributes in the training and test samples that are similar. Choose the category to which the neighbor belongs first, then locate the k closest neighbors, and then choose the class for the upcoming data [20].
- Choosing the value of k, the parameter that determines how many neighbors the KNN algorithm can use, is an alternative strategy. The effectiveness of the KNN algorithm is significantly influenced by the selection of k [42].

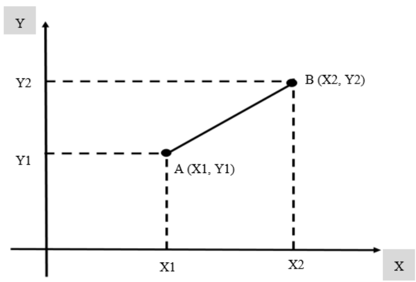

Figure 16. Displaying the distance between the extracted attributes and the ambiguous words.

Choosing the best value for k in this strategy requires careful consideration. Both too small and too huge are undesirable. When k is too little, the model becomes excessively specific, struggles to generalize, and is more sensitive to noise. Although the model predicts well on the training dataset, but it performs poorly on the test data. This condition is called the "over-fitting" of the model. For both the learning and assessment data sets, the extended model won't be a good indicator if k is set too high. This situation is termed as under-fitting [14]. WEKA's inbuilt KNN function is known as IBK. Figure 16 illustrates the computation of the Euclidean distances D1 and D2 for the traits that matched the test data. This diagram illustrates how the distances were calculated with K=1.

The following equation can be used to get the Euclidean distance between any two points:

$$=\sqrt{((X_2-X_1)^2+(Y_2-Y_1)^2)} \qquad (11)$$

Here, the coordinates of the test word are in this case *X*1 and *Y*1and the coordinates of the matched feature are *X*2 and *Y*2.

Suppose *X* is an instance domain and metric function is *ρ*: then the distance between any two points between *X* is return by the equation:

For example, if *X*=*Rd* then *ρ* can be the Euclidean distance calculated by the following equation:

$$\rho(X,X^{'})=|(|x-x^{'}|)|=\sqrt{〖\Sigma_{(i=1)}^{d}(xi-x^{'}i)〗^2} \qquad (12)$$

A set of training examples is denoted by *S*=(*x*1, *y*1), ..., (*xm*, *ym*). Let 1(*x*), ..., *m*(*x*) be a reordering of 1, ..., *m* in accordance with their separation from *x*, *xi* for each *x X*, i.e., for all *i*<*m*.

$$\rho(X,X_{\pi i(x)})\leq\rho(X,X_{\pi i(x+1)}) \qquad (13)$$

When k=1, then 1-NN rule can be defined as follows:

$$h_{(s(x)=y(\pi 1(x)))} \qquad (14)$$

One may define the prediction for regression issues, namely *Y*=*R*, as the average goal of the KNNs.

$$h_{(s(x))}=1/2\ \Sigma(i=1)^{k}\ y_{(\pi i(x))} \qquad (15)$$

More specifically, the k-NN rule with regard to some function: 〖(*X*×*Y*)^(*k*)→*Y*, the k-NN rule with respect to will be:

$$h_{(s(x))}=\emptyset((X_{(\pi 1(x),)}\ y_{\pi 1(x)}\ ,\ldots\ldots\ 〖(X_{(\pi k(x),)}\ y_{\pi k(x)})) \qquad (16)$$

We may get a weighted average of the targets based on their separation from *x* if *Y*=*R*:

$$h_{(s(x)}=\sum_{(j=1)}^{k}(\rho(x,x_{(\pi i(x))})/(\rho(x,x_{(\pi i(x))})y_{(\pi i(x))}) \qquad (17)$$

4.5. Support Vector Machine Algorithm

Support Vector Machine (SVM) are supervised learning techniques, therefore they require labelled, previously studied data to categorize newly discovered data. The fundamental strategy for categorizing the data begins with an effort to develop a function that divides the data points into the appropriate labels with, a) the least amount of mistakes or b) the widest margin. Due to the labels' improved ability to identify one another, greater vacant spaces near to the splitting function produce less mistakes.

SVM operates on both simple and complicated datasets and is like a sharp knife. It is a more effective and robust algorithm for creating ML models. It primarily helps with classification problems. This method allows us to represent each data item as a [9] point with a value for each feature in n-dimensional space. The hyperplane can then be found in order to undertake classification. This hyper-plane makes a clear distinction between the two different classes [17]. The

Sequential Minimal Optimizations (SMO) is the in-built function for SVM in WEKA. Figure 17 is showing the support vectors and hyperplane of the datasets.

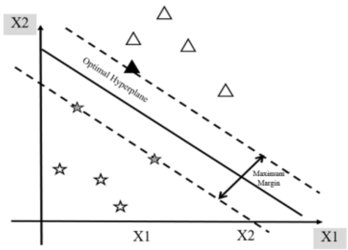

Figure 17. Features indicating the support vectors in SVMs algorithms.

4.6. Logistic Regression

This supervised text classification system uses a probability-based methodology. When the output will be a binary number and the data sets are unconditional, a predictive method is applied. Binary classification issues are those that base their solutions on binary output [40]. The LR curve is "S" shaped curve called sigmoid and is displayed in Figure 18.

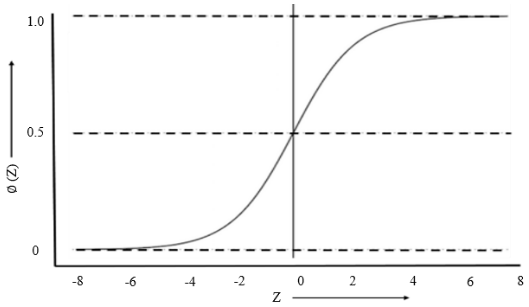

Figure 18. Curve of LR.

We use the sigmoid function in ML to convert predictions to probabilities. The function maps any real value between 0 and 1. The sigmoid function of a LR is denoted by the following equation:

$$"\varnothing(Z)"=1/(1+"e\text{-}z") \tag{18}$$

Here, "$\varnothing(Z)$" is the calculated probability.

The hypothesis class for homogenous linear function is given by the following equation:

$$H\_sig=\varphi sig{}^{\circ}L\_d=\{X\_7\rightarrow\varphi sig(w,x\_()):w\in R\text{^}d \tag{19}$$

If (w, x) is very large then $\varphi\_(sig\ )\ ((w,x))$ is close to 1, while if its value is too small then $\varphi\_(sig\ )\ ((w, x))$ is close to 0.

5. Proposed Methodology

In this analysis, we suggest a hybrid model called "AmbiF" to get a more precise text categorization result. The hybrid model combines multiple categorization techniques into a single model to enhance the system performance [12]. This model was created for the predictions of test sentences having an ambiguous words. We employed the stacking method of the assembling methodology in our model [26, 41]. Figure 19 depicts the suggested model for the new class prediction.

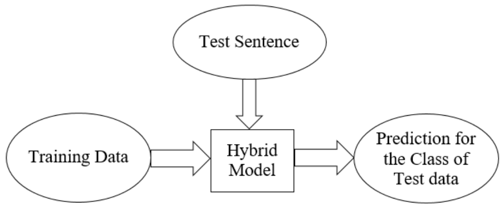

Figure 19. Suggested model for the prediction of test sentences.

5.1. AmbiF Model

The "AmbiF" model was developed to more accurately remove POS ambiguities. We have merged the three supervised text classification methods in one model. The study is carried out in the WEKA data analysis tool using the methods SVM, DT, and NB [10]. A comprehensive hybrid model is shown in Figure 20.

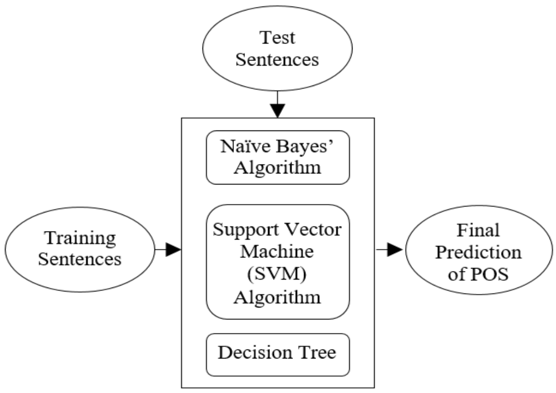

Figure 20. The hybrid model "AmbiF."

We merged SVM, DT, and NB classifier techniques in the suggested model AmbiF to obtain high-quality disambiguation. The in-built function in WEKA for SVM is SMO, and J48 for DT.

5.2. Functioning of AmbiF Model

The following actions are taken while using the AmbiF model:

1. The input is given in English sentences with ambiguous words having different part-of-speech.
2. Tokens are produced from the sentence's individual words.
3. If an ambiguous word is discovered during the morphological analysis process using a lexical/ambiguous dataset, it becomes our target word, and we must disambiguate it.
4. Using the AmbiF model in relation to the training dataset, we must disambiguate the target word. SVM, DTs, and Naive Bayes are the three supervised ML categorization methods included in this model.
5. After using this model, we have obtained the text that had been cleared of all ambiguity and sent it to EtranS

to determine its precise meaning.

The overall functioning of the AmbiF model is presented in Figure 21.

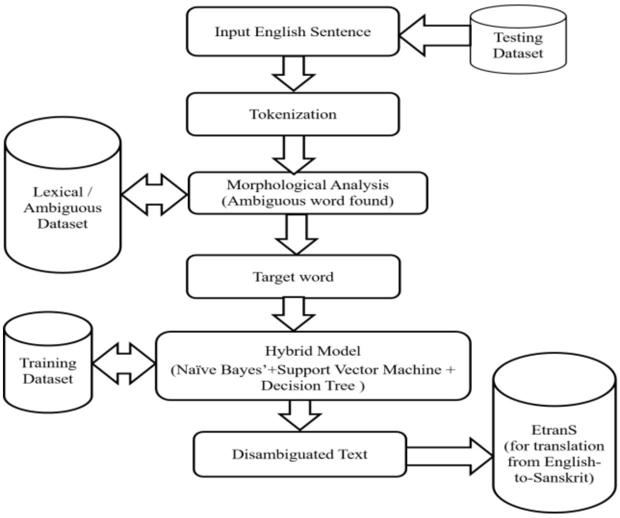

Figure 21. Functioning of the AmbiF model.

5.3. Datset for Experiment

The use of datasets is crucial in ML. There are two categories of dataset: training dataset and test dataset. The sentences in the training dataset contain a variety of ambiguous keywords. To predict the POS, the system has been trained using the AmbiF model to extract information from the surrounding around the ambiguous target word. A group of sentences are used as test data to evaluate a particular model. The training that was given to the model has a direct correlation with the prediction's accuracy. The outcomes are better the better trained a model is. We have used a data set of 2000 sentences to evaluate the effectiveness of supervised ML algorithms along with the model AmbiF.

The goal is to train the algorithm and get a reliable POS prediction for the provided test data. The training of the system and correct prediction for the given test data set will be made possible by the information provided by words close to the ambiguous word. The training dataset is based on the collocation method of window size three. With window size 3, this dataset includes information about the nearby words in relation to the target term. It denotes the three words immediately left and right of the ambiguous target term [28]. Also includes the appropriate part of speech for these adjacent words, and class of the ambiguous target word. The format of the dataset is shown in Table 3.

Depending on the part of speech of their unclear target word in the sentence, the sentences are classified into five class: nouns, verbs, adverbs, prepositions, and adjectives. Nominal attribute types make up the attributes that were chosen. The Naive Bayes classification approach's collocation model serves as the foundation for this feature extraction technique.

Table 3. Sample of dataset.

S.No.	Sentences	W-3	POS(W-3)	W-2	POS(W-2)	W-1	POS(W-1)	W	POS(w)	W-+1	POS(W+1)	W+2	POS(W-+2)	W+3	POS(W+3)	Classes
1	He is sitting on the back seat.	nil	nil	He	Pronoun	sitting	Verb	back	Adjective	seat	Noun	nil	nil	nil	nil	Adjective
2	Her back is not staright.	nil	nil	nil	nil	her	Pronoun	back	Noun	not	Preposition	straight	Adjective	nil	nil	Noun
3	Please send my books back.	send	verb	my	Pronoun	books	Noun	back	Adverb	nil	nil	nil	nil	nil	nil	Adverb

5.4. Testing Method

The ten-fold cross-validation testing method was used for the experiment. For assessing the model's predictions for the class, this testing technique is the most popular data resampling process. The dataset is divided into ten sections for cross-validation with ten folds at random. In nine of these 10 portions, the training data is used, and in the tenth, the testing data is used. The tenth part is checked after each repetition of the procedure [21]. Assume, for instance, that k is equal to 10 and we have a dataset of 100 statements, numbered S1 through S100. The value of k determines how many folds are utilized to partition the dataset. The data is first shuffled, then it is divided into ten folds. The number of folds used to divide the dataset depends on the value of k. We first shuffle the data before dividing it into ten folds. We will do ten folds, each of which will include ten sentences, for a total of one hundred sentences.

It always leaves one fold for validation in every iteration. The data is provided to the DT, SVM, and NB algorithms. The training set was created for each classification algorithm so that a model could be learned. As a result, it is possible to group data instances into recognized classes. Any algorithm's association procedure involves the following steps:

1. Creating an ARFF or CSV-formatted training dataset.
2. Attribute categorization and identification.
3. With the training data set, the learning model is built.
4. The AmbiF model's final prediction for the current data set.

5.5. Experimental Result and Analysis

To observe the text classification objectives, all supervised WSD algorithms are assessed. Many instances that were appropriately identified served as the basis for evaluating each method. The accuracy has been considered in order to determine how well each classification method performs. The primary result of this study was a comparison of the popular supervised ML algorithms (mentioned above) in terms of precision and accuracy. The effectiveness of the algorithms was compared using a variety of statistical methods. They included F1-score, recall, and precision. Screenshot of the dataset in Weka tool is shown in Figure 22.

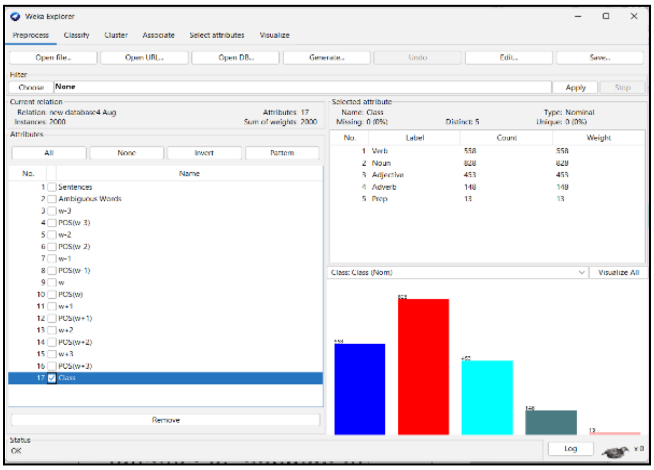

Figure 22. Screenshot of frequency graph of class attribute with respect to total number of part of speech for the dataset.

Using the following formula, one may determine the overall accuracy:

$$Accuracy=[true\ positive + true\ negative]/N \quad (20)$$

Where, $N =$

$$[true\ positive+true\ negative+false\ positive+false\ negative] \quad (21)$$

The experiment is conducted, and the results are generated, on the given dataset. Each experiment is assessed using stratified ten-fold validation to make sure the findings are unbiased. The model was trained to increase precision once the dataset was analyzed. To categorize the test data into the noun, verb, adjective, and adverb classes, 2000 sentences in total are analyzed. The accuracy of the algorithms are based on the total number of correctly classified sentences.

Table 4 lists the results of supervised classification techniques, including the AmbiF model that was suggested.

Table 4. Ten-fold cross-validation test results.

S. No.	Algorithm	Accuracy Percentage
1.	LR	68.4783
2.	IBK	71.5839
3.	RF	72.6708
4.	NB classifier	78.1056
5.	SMO	82.2981
6.	DT (J48)	84.6273
7.	Hybrid Approach	84.7826

From the Table 4, we see that in AmbiF model, the total number of correctly classified sentences are increased with respect to all other grouping methods. The percentage of sentences that are successfully identified in the hybrid model is 84.78%, whereas it varies between 68.47 and 84.62% in other studied approaches.

The detailed accuracy of AmbiF model is compared with other methods with respect to recall, precision, and F1-score. The accuracy reported in Table 5 shows the weighted accuracy value under Precision, recall, and F-score for all the analyzed methods including hybrid model.

6. Comparative Analysis

We are analyzing and comparing the algorithms discussed for text classification and predicting their performance evaluation based on the correct and incorrect classification of ambiguous words. This evaluation is done using features extracted from the dataset. The outcome of this testing determines the effectiveness of each algorithm on the datasets and provides information on the accuracy of each algorithm. For this testing, a dataset of 2000 sentences was prepared and used to test the performance of various ML algorithms, including NB, DT, RF, KNN, SVM, LR, and NNs. The results are compared in terms of accurately classified sentences. The results of the comparisons are shown in Table 5.

Table 5. Weighted accuracy under precision, recall, and F1-score.

S. No.	Algorithm	Precision	Recall	F1-score
1.	LR	0.691	0.685	0.686
2.	KNN (IBK)	0.720	0.716	0.716
3.	RF	0.763	0.727	0.714
4.	NB classifier	0.786	0.781	0.780
5.	SVM (SMO)	0.824	0.823	0.823
6.	DT (J48)	0.848	0.846	0.846
7.	AmbiF model (proposed techniques)	0.849	0.848	0.848

Table 5 shows the accuracy values for the ten-fold cross-validation test performed for all the six algorithms and also for our hybrid model [6, 8]. The efficiency of AmbiF model is improved with the reported higher accuracy of all other analyzed algorithms. The AmbiF model's accuracy under the "F1-measure" is stated to be 0.848. Thus it could be a better approach with improved accuracy for the English-to-Sanskrit translation. Figure 23 displays a comparison chart for a selection of successfully categorized texts.

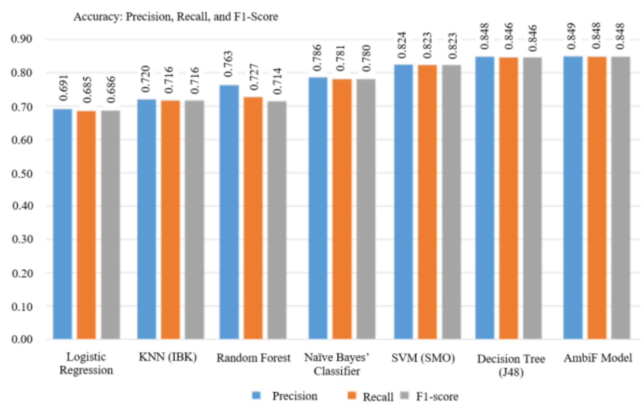

Figure 23. Accuracy chart under precision, recall, and F1-score.

The Figure 23 shows the accuracy measures of six analyzed supervised text classification algorithms including with the hybrid model. The LR approach yields the minimum result, while the DTs approach yields the maximum result among all the analyzed approaches. The result is further improved in the hybrid approach, which achieves a result of approximately eighty-five percent.

7. Conclusions and Future Scope

The goal of this research is to create a hybrid model for the expectation of WSD problems by using ML classification techniques with appropriate features. The pre-processed dataset was prepared and the efficiency of the various supervised classification approaches are checked. These algorithms are LR, NB, KNN, SVM, DT, and RF using the WEKA tool of ML software was tested. Precision, recall, and F1-measure were used to assess how well the strategies performed. The accuracy of every supervised algorithm under study ranges from 68.4783 to 84.6273 percent. The AmbiF model has been established for better performance in enhanced WSD techniques. The current work is in the direction of building a more accurate algorithm for the English-to-Sanskrit MT. The disambiguation accuracy is improved and reported 84.7826 percent.

All the validations were implemented only on 17 attributes. In future we will increase our data size and window size to improve the accuracy of the developed hybrid model AmbiF. After the pre-processing, the classification result may be more accurate because the data will be consistent.

Acknowledgement

This paper is the part of the Post-Doctoral Study from Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA, USA. We are thankful to Dr. Ajith Abraham for his continuous guidance, support, and funding.

Author's Contribution

All the authors contributed equally to the manuscript. The contributions of each author are as follows: The first author confirmed the study conception, design, and selection of the machine learning tool. The third author was responsible for data collection, conducting the experiment, and interpreting the results. The second author prepared the draft manuscript. The fourth author reviewed the results and proofread the manuscript. Finally, all the authors have approved the final version of the manuscript.

References

[1] Abraham A., Gupta B., Maurya A., Verma S., Husain M., and Ali A., "Naïve Bayes Approach for Word Sense Disambiguation System with a Focus on Parts-of-Speech Ambiguity Resolution," *IEEE Access*, vol. 12, pp. 126668-126678, 2024. https://ieeexplore.ieee.org/document/10663552

[2] Alian M. and Awajan A., "Generating Sense Inventories for Ambiguous Arabic Words," *The International Arab Journal of Information Technology*, vol. 18, no. 3A, pp. 446-451, 2021. https://doi.org/10.34028/iajit/18/3A/8

[3] Alloghani M., Al-Jumeily D., Mustafina J., Hussain A., and Aljaaf A., *Supervised and Unsupervised Learning for Data Science*, Springer, 2020. https://link.springer.com/chapter/10.1007/978-3-030-22475-2_1

[4] Aung N., Soe K., and Thein N., "A Word Sense Disambiguation System Using Naïve Bayesian Algorithm for Myanmar Language," *International Journal of Scientific and Engineering Research*, vol. 2, no. 9, pp. 1-6, 2011. https://www.uit.edu.mm/wp-content/uploads/2020/05/NTTA-4.pdf

[5] Bahadur P. and Chauhan D., "Machine Translation-A Journey," *in Proceedings of the Science and Information Conference*, London, pp. 187-195, 2014. https://ieeexplore.ieee.org/document/6918189?denied=

[6] Bates S., Hastie T., and Tibshirani R., "Cross-Validation: What does it Estimate and how Well does it Do it?," *Journal of the American Statistical Association*, vol. 119, no. 546, pp. 1434-1445, 2024. https://doi.org/10.1080/01621459.2023.2197686

[7] Boyle B., *Support Vector Machines: Data Analysis, Machine Learning and Applications*, Nova Science Publishers, 2011.

[8] Brownlee J., "A Gentle Introduction to K-Fold Cross-Validation," *Machine Learning Mastery*, 2018. https://scholar.google.com/citations?user=hVaJhRYAAAAJ&hl=en&oi=sra

[9] Chauhan N., Decision Tree Algorithm. Explained-KDnuggets, https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html, Last Visited, 2024.

[10] Dhar S., Roy K., Dey T., Datta P., and Biswas A., "A Hybrid Machine Learning Approach for Prediction of Heart Diseases," *in Proceedings of the 4th International Conference on Computing Communication and Automation*, Greater Noida, pp. 1-6, 2018. https://ieeexplore.ieee.org/abstract/document/8777531

[11] Escudero G., Marquez L., and Rigau G., "Naive Bayes and Exemplar-Based Approaches to Word Sense Disambiguation Revisited," *arXiv Preprint*, vol. arXiv:cs/0007011, pp. 1-5, 2000. https://arxiv.org/abs/cs/0007011

[12] Frank E., Hall M., Holmes G., Kirkby R., Pfahringer B., Witten I., and Len T., *Data Mining and Knowledge Discovery Handbook*, Springer, 2010.

https://link.springer.com/chapter/10.1007/978-0-387-09823-4_66

[13] Gawande U., Kale S., and Thaokar C., *Recent Advances in Material, Manufacturing, and Machine Learning*, CRC Press, 2023.

[14] Guo G., Wang H., Bell D., Bi Y., and Greer K., "KNN Model-Based Approach in Classification," *in Proceedings of the OTM Confederated International Conferences, on the Move to Meaningful Internet Systems*, Sicily, pp. 986-996, 2003. https://link.springer.com/chapter/10.1007/978-3-540-39964-3_62

[15] Hotho A., Staab A., and Stumme G., "Ontologies Improve Text Document Clustering," *in Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, pp. 541-544, 2003. https://ieeexplore.ieee.org/document/1250972

[16] Jabbar M. and Samreen S., "Heart Disease Prediction System Based on Hidden Naïve Bayes Classifier," *in Proceedings of the International Conference on Circuits, Controls, Communications and Computing (14C)*, Bangalore, pp. 1-5, 2016. https://ieeexplore.ieee.org/document/8053261

[17] Jeyanthi S., Efficient Classification Algorithms Using SVMs for Large Datasets, Master Theses, Indian Institute of Science, 2007. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3f9c48f6cf894d30b9c22d6ced2c5c77882e4bd6

[18] Kesavaraj G. and Sukumaran S., "A Comparison Study on Performance Analysis of Data Mining Algorithms in Classification of Local Area News Dataset Using WEKA Tool," *International Journal of Engineering Sciences and Research Technology*, vol. 2, no. 10, pp. 2748-2755, 2013. https://scholar.google.co.in/citations?user=FXbZ6e0AAAAJ&hl=en

[19] Kumar N., Kumar V., Kumar G., and Sharma R., "The Role of Human Computer Interaction," *Mukt Shabd Journal*, vol. 11, no. 6, pp. 284-287, 2022. https://app.box.com/s/eh96kvxeiov6ek6vjnqhdrbsi5cc3vus

[20] Le C. and Shimazu A., "High WSD Accuracy Using Naive Bayesian Classifier with Rich Features," *in Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation*, Tokyo, pp. 105-114, 2004. https://aclanthology.org/Y04-1011.pdf

[21] Liu Y., Wang X., Wang L., and Lv Z., "A Bayesian Collocation Method for Static Analysis of Structures with Unknown-But-Bounded Uncertainties," *Computer Methods in Applied Mechanics and Engineering*, vol. 346, pp. 727-745, 2019. https://doi.org/10.1016/j.cma.2018.08.043

[22] Martin-Valdivia M., Urenna-Lopez L., and Garcia-Vega M., "The Learning Vector Quantization Algorithm Applied to Automatic Text Classification Tasks," *Neural Networks*, vol. 20, no. 6, pp. 748-756, 2007. https://doi.org/10.1016/j.neunet.2006.12.005

[23] Maurya A. and Bahadur P., *Proceedings of Second Doctoral Symposium on Computational Intelligence*, Springer, 2021. https://colab.ws/articles/10.1007%2F978-981-16-3346-1_56

[24] Mitchell T., *Machine Learning*, McGraw-Hill Science, 1997. https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf

[25] Navigli R., "Word Sense Disambiguation: A Survey," *ACM Computing Surveys* (CSUR), vol. 41, no. 2, pp. 1-70, 2009. https://dl.acm.org/doi/abs/10.1145/1459352.1459355

[26] Pal A., Kundu A., Singh A., Shekhar R., and Sinha K., "A Hybrid Approach to Word Sense Disambiguation Combining Supervised and Unsupervised Learning," *International Journal of Artificial Intelligence and Application*, vol. 4, no. 4, pp. 89-101, 2013. https://arxiv.org/pdf/1611.01083

[27] Palanati D. and Kolikipogu R., "Decision List Algorithm for Word Sense Disambiguation for Telugu Natural Language Processing," *International Journal of Electronics Communication and Computer Engineering*, vol. 4, no. 6, pp. 176-180, 2013. https://ijecce.org/Download/conference/NCRTCST-2/38NCRTCST-13084.pdf

[28] Pavlyshenko B., "Using Stacking Approaches for Machine Learning Models," *in Proceedings of the 2nd International Conference on Data Stream Mining and Processing*, Lviv, pp. 255-258, 2018. https://ieeexplore.ieee.org/document/8478522

[29] Ramachandran R. and Mammone R., Modern Methods of Speech Processing, Springer, 2012. https://www.amazon.in/Processing-Springer-International-Engineering-Computer/dp/1461359627

[30] Ranjan Pal A. and Saha D., "Word Sense Disambiguation: A Survey," *International Journal of Control Theory and Computer Modeling*, vol. 5, no. 3, pp. 1-16, 2015. https://www.airccse.org/journal/ijctcm/papers/5315ijctcm01.pdf

[31] Samuel A., "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 44, no. 3, pp. 535-554, 1959. https://people.csail.mit.edu/brooks/idocs/Samuel.pdf

[32] Settouti N., El Amine Bechar M., and Amine Chikh M., "Statistical Comparisons of the Top 10 Algorithms, in Data Mining for Classification

Task," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.4 , no. 1, pp. 46-51, 2016. DOI:10.9781/ijimai.2016.419

[33] Singh R., Ghosh K., Nongmeikapam K., and Bandyopadhyay S., "A Decision Tree Based Word Sense Disambiguation System in Manipuri Language," *Advanced Computing: An International Journal*, vol. 5, no. 4, pp. 17-22, 2014. DOI:10.5121/acij.2014.5403

[34] Sinha R. and Jain A., "AnglaHindi: An English to Hindi Machine-Aided Translation System," *in Proceedings of 9th Machine Translation Summit: System Presentations*, New Orleans, pp. 1-5, 2003. https://aclanthology.org/2003.mtsummit-systems.15

[35] Szepesvari C., *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Springer, 2010. https://link.springer.com/book/10.1007/978-3-031-01551-9

[36] Tiwary U. and Siddiqui T., *Natural Language Processing and Information Retrieval*, Oxford University Press, 2008. https://dl.acm.org/doi/book/10.5555/1481140

[37] Venkatesan N. and Priya G., "A Study of Random Forest Algorithm with Implementation Using WEKA," *International Journal of Innovative Research in Computer Science and Engineering*, vol. 1, no. 6, pp. 156-162, 2015. https://www.ioirp.com/Doc/IJIRCSE/i6/JCSE242.pdf

[38] Verma S. and Chandran S., "Contactless Palmprint Verification System Using 2-D Gabor Filter and Principal Component Analysis," *The International Arab Journal of Information Technology*, vol. 16, no. 1, pp. 23-29, 2019. https://ccis2k.org/iajit/PDF/January%202019,%20No.%201/11209.pdf

[39] Wang L., "Research and Implementation of Machine Learning Classifier Based on KNN," *IOP Conference Series: Materials Science and Engineering*, vol. 677, no. 5, pp. 1-6, 2019. DOI:10.1088/1757-899X/677/5/052038

[40] Wang S. and Wang S., *Interdisciplinary Computing in Java Programming*, Spriger, 2003. https://link.springer.com/chapter/10.1007/978-1-4615-0377-4_5

[41] Zenko B., Todorovski L., and Dzeroski S., "A Comparison of Stacking with Meta Decision Trees to Bagging, Boosting, and Stacking with other Methods," *in Proceedings of the IEEE International Conference on Data Mining*, San Jose, pp. 669-670, 2001. https://ieeexplore.ieee.org/document/989601

[42] Zhang Z., "Introduction to Machine Learning: K-Nearest Neighbors," *Annals of Translational Medicine*, vol. 4, no. 11, pp. 1-7, 2016. https://pmc.ncbi.nlm.nih.gov/articles/PMC4916348/

Ajith Abraham received the B.Tech. Degree in electrical and electronic engineering from the University of Calicut, in 1990, the M.Sc. degree from Nanyang Technological University, Singapore, in 1998, and the Ph.D. Degree in computer science from Monash University, Melbourne, Australia, in 2001. He is the Vice Chancellor of Bennett University, New Delhi. Prior to this, he was the Dean of the Faculty of Computing and Mathematical Sciences, FLAME University, Pune, and the Founding Director of the Machine Intelligence Research Labs (MIR Labs), USA, a not-for-profit scientific network for innovation and research excellence connecting industry and academia. He also held two international university professorial appointments, such as, a Professor of Artificial Intelligence with Innopolis University, Russia, and the Yayasan Tun Ismail Mohamed Ali Professorial Chair of artificial intelligence with UCSI, Malaysia. He is involved in a multidisciplinary environment. He has authored/coauthored more than 1,500+ research publications out of which there are more than 100 books covering various aspects of computer science. One of his books was translated into Japanese and a few other articles were translated into Russian and Chinese. He has more than 60,000 academic citations (H-Index of more than 115+ as per Google Scholar). He has given more than 250 plenary lectures and conference tutorials (in more than 20 countries). He was the Chair of the IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing (which has over more than 200 members), from 2008 to 2021, and served as a Distinguished Lecturer for the IEEE Computer Society Representing Europe, from 2011 to 2013. He was the Editor-in-Chief of Engineering Applications of Artificial Intelligence (EAAI), from 2016 to 2021, and serves/served on the Editorial Board for over 15 international journals

Bineet Kumar Gupta is pursuing Post_Doc in Deep Learning from MIR Labs Washington USA. His area of research is Deep Learning, Information Retrieval and Security, Software Engineering and ICT. Dr. Gupta has wide experience of guiding PG and Doctoral students of Computer Science stream. He has published more than 50 research papers in various journals, conferences and symposium of national and international recognition.

Satya Bhushan Verma is working as an Associate Professor and Head of Department of Computer Science and Engineering, Institute of Technology in Shri Ramswaroop Memorial University Lucknow-Deva Road, UP, India. Dr. Satya Bhushan Verma has 7 years of teaching experience at undergraduate, post graduate level and research experience. He completed Doctorate in Computer Science and Engineering from NIT, Durgapur, and Master of Technology in Computer Science and Engineering and Bachelor of Technology in Computer Science and Engineering. His area of research is Blockchain, IoT, Computer Vision, Biometric, Pattern Recognition and ICT. He has actively been involved in number of international committees as abstract reviewer, workshop reviewer and panel reviewer. He has published several research papers in various journals, conferences and symposia of national and international recognition. Currently, he is working in the area of Computer Science and Engineering Blockchain Technology, IoT, Computer Vision, Biometric, Pattern Recognition and ICT.

Archana Sachindeo Maurya is working as an Assistant Professor at Shri Ramswaroop Memorial University Deva Road. She has completed her master's degree in Computer Application from IGNOU. She is currently working in the areas of Machine Translation, Machine Learning and Natural Language Processing. She has 10+ years of experience in teaching.

Mohammad Husain is holding Ph.D. Degree in Computer Science and Engineering 2008. He did his M. Tech. in Computer Science 2005 and Bachelor of Engineering in Computer Science and Engineering in 1990. He has about 33 years of professional experience in the field of IT and Academics. His area of research is Information Retrieval, Web Mining, and Big Data. Currently he is working in the Department of Computer Science, Faculty of Computer and Information Systems with Islamic University of Madinah, Kingdom of Saudi Arabia, Postal Code. 42351; P.O. Box. 170. He also has industrial exposure and has experience in Software Design and Development. He has published about 159 papers in different National/International Journals/proceedings. Dr. Husain has guided 14 Ph.D. and 11 M. Tech. Students. He has authored a book on Principles of Programming Languages and contributed number of book chapters. He is also an active member of various national/international journals, committees and societies.

Arshad Ali joined Islamic University of Madinah in December 2014 as Assistant Professor and then promoted as Associate Professor in July 2018 and later in June 2023 he is promoted as full Professor of Information Technology in Faculty of Computer and Information Systems, Islamic University of Madinah. He finishes his BSc in Mathematics and Statistics from University of Punjab, Lahore, Pakistan in 2000 and completed his Masters in Computer Sciences from Iqra University, Lahore, Pakistan. In 2005, he moved to Birmingham, UK for further studies. He joined Aston University, Birmingham, UK and obtained his MSc Telecommunication Technology in 2007. In 2007, he joined Geotechnical Group, Department of Engineering, and University of Cambridge as Research Ass. (2007-2009). In 2009, he was awarded a Ph.D. (2009- 2012) scholarship from the Lancaster University, UK and he awarded Ph.D. in 2012. He worked on the UK-NEES Project and Designed Communication System for live experimentation between UK Universities (Cambridge, Oxford and Bristol).

Sami Alshmrany is Associate Professor of Information Technology since October 2022. He finished his BSc in Computer since from University of King Abdulaziz, Jeddah, Saudi Arabia in 2008. In 2009, he moved to Adelaide, Australia for further studies. He joined Flinders University, Adelaide, Australia and obtained his MSc in Information Technology in 2013. In 2018 he finished his PhD in Information Technology from Flinders University, Adelaide, Australia.

Sanjay Gupta is a Corporate Vice President in HCL Technologies. He is a dynamic leader and a founding member of the infrastructure division at HCL Technologies Ltd (HCL), He holds his bachelor's degree in Electronics and Communication from Pune University. Sanjay has more than 27 years of experience in the IT Industry, having played various roles in Business Development, Service Delivery, Strategic Initiatives and Program Management. He is currently leading HCL's foray in new strategic locations in Asia such as Vietnam, South Korea, Taiwan and Bangladesh. Sanjay Gupta was leading multiple strategic initiatives across the organization such as the launch of New Vistas Program by creating new delivery centers of 15,000 employees in three locations (Lucknow, Nagpur and Madurai) in India. Sanjay will bring in his extensive experience of the Indian subcontinent to replicate HCL's success story in Bangladesh, Vietnam, South Korea and Taiwan.