COMPARATIVE ANALYSIS OF DIFFERENT CLASSIFIERS FOR CASE BASED MODEL IN PUNJABI WORD SENSE DISAMBIGUATION

Himdweep Walia[1]*, Ajay Rana*, Vineet Kansal**
*Amity University, Noida, Uttar Pradesh, India
**IET, Lucknow, Uttar Pradesh, India

ABSTRACT
Research is being carried out for machines to be able to better decipher an ambiguous word. The majority of work done in Punjabi, a regional language of India and one among the 10 most spoken languages of the world, is limited to knowledge-based techniques. The implementation of Case Based Model to help decipher the Punjabi ambiguous word is new and hence the results determined can be beneficial exemplar in Punjabi Word Sense Disambiguation research. Vectorization of the sentence is done to use minimal features to help find the right context of the given ambiguous word. Four different measuring functions are used to measure the nearness of the given sample with respect to store sample, thereby using the concept of case-based reasoning. The collected sample is then subjected to four different classifiers, namely Naïve Bayes, k-Nearest Neighbor, Decision Tree and Artificial Neural Network to find the closest context. The experimentation shows the variation in results subject to the size of the vector.

KEYWORDS: Natural Language Processing, Word Sense Disambiguation, Punjabi language, Case Based Reasoning, Classifiers, Similarity Function.

MSC: 68T50

RESUMEN
Se desarrolla una investigación para máquinas que son capaces de descifrar mejor una ambigua palabra. La mayoría el trabajo se desarrolló con el Punjabi, un lenguaje de la regional de la India y que es una de las más utilizadas entre la 10 más habladas en el mundo, y que es limitada para la tecnología. La implementación de un Modelo Basado en Caso para ayudar a descifrar palabras ambiguas del Punjabi es nuevo y por lo tanto los resultados obtenidos pueden ser un beneficioso ejemplo en el marco de la investigación "Punjabi Word Sense Disambiguation". La vectorización de las sentencias es desarrollada para usar minimales estructuras para ayudar a hallar el contexto correcto de la palabra ambigua. Cuatro funciones de medición diferentes se usan para medir la cercanía de una muestra dada respecto a la muestra archivada, por lo que se usa el concepto de razonamiento basado en caso. La muestra obtenida es entonces evaluada usando cuatro clasificadores diferentes, nombrados Naïve Bayes, k-Vecinos Más Cercanos, Árbol de Decisión y Red Artificial Neuronal para hallar el más cercano contexto. La experimentación muestra que la variación en los resultados están sujetos al tamaño del vector.

PALABRAS CLAVE: Procesamiento del Lenguaje Natural, Desambiguación del Sentido de la Palabra, Lenguaje Punjabi, Razonamiento Basado en Caso, Clasificadores, Función de Similaridad.

1. INTRODUCTION

The natural language is the tool through which the humans are able to communicate with each other. Though the language has a vast vocabulary but even then a single word has multiple meanings and its usage is dependent on the context in which it is being used. Being the creator of the natural language, this differentiation comes easy to humans but coding the same intelligence into a machine is a different story and so Word Sense Disambiguation (WSD) [11] is catered as an NP-hard problem in the world of Natural Language Processing (NLP). To illustrate the point further, consider the word, "default" which can mean the option that would be taken if you do not pick something ("the default value is true") or it can mean to fail to meet a legal requirement ("he is going to default his next bank installment"). The idea is that the machine (computer) is able to differentiate the context in which the given ambiguous word is being used. This is

[1] himdweep@yahoo.com

necessary when we look at the application areas of NLP, which are translation, word generation, speech recognition and like-wise. The best way to find the context of the given ambiguous words is through the words surrounding this word. The machine is taught to pick a group of words including the ambiguous word and then decipher the correct meaning.

Due to the extent of its relevance in NLP, WSD is becoming a topic of research in all the languages of the world [9]. In our paper, we are trying to decipher the correct meaning of an ambiguous word in the language Punjabi. Punjabi is one of the regional languages of India and comes among the top 10 languages been spoken and written in the world. There is a large amount of literature written in Punjabi and number of publications including newspapers, magazines, journals, books, etc. is available. This work is being made available online to reach maximum readers. Most of the work in this language has been done in translation to various languages. In terms of WSD, the majority of the work has been limited to knowledge-based techniques. This has led to working on various supervised and un-supervised techniques to help decipher the Punjabi ambiguous words.

The supervised and un-supervised techniques are being used to help in deciphering the word. In a supervised approach, we train the machine with the set of training data and then the sample data is fed and based on the training the machine gives the result. In case of unsupervised approach, unlabeled data is used to find the result based on the stored data. When catering with NLP problems with WSD, the given text is laced with long sentences and this consequently becomes difficult to decipher. The conversion of the text into vectors helps in reducing the size of the context window [15, 21]. This reduction of size would improve the overall effectiveness of the classifiers being used. The vectors are then subjected to the Case Based Reasoning (CBR) methodology to find similar cases. CBR resolves the problem by finding a case similar to the given problem and then utilizing this knowledge to find the right context of the given ambiguous word. This was the approach which has been experimented with in this paper.

2. LITERATURE REVIEW

Natural Language Processing is the science of incorporating the same level of comprehension as humans have into machines. With the advent of Artificial Intelligence into our natural lives, this precedent has become all the more necessary. One major area of research has been Word Sense Disambiguation [11] which is a rather daunting scenario as deciphering of the right context of the word becomes essential in understanding the language better. Due to this, a lot of research has went into English, European languages and some Asian languages like Chinese and Japanese [9]. Work has been carried out in Indian Regional Languages [2] as well, primarily in Hindi. The majority of the work done in Indian Regional Languages like Tamil, Manipuri, Punjabi, Malayalam, Bangla, etc. is restricted to creation of dictionary and knowledge-based supervised methodology [20].

The language of choice for this paper is Punjabi (*script*, Gurmukhi), one of the regional languages spoken and well documented in India and all over world with considerable amount of population residing in Canada and United Kingdom. Two survey papers [8, 19] were referred to understand the nature of research being carried out in this language and it was noted that the majority of work involved knowledge-based and supervised methodologies. As Punjabi drives closely from Hindi [15], work done in this language along with English [16, 17, 18] has been studied for this paper. In order to decipher the correct context of the word, we need a context window [15] i.e. n-number of words surrounding the given ambiguous word. In the paper by Walia, et. al. [21], 3 different classifiers have been used, Naïve Bayes, k-NN, and, Decision Tree, and the results indicates that with a larger context window the results are better and Naïve Bayes shows the best accuracy out of the 3 classifiers. The inclusion of large number of words to help decipher the correct context of the word reduces the effectiveness of the classifiers. This increases the sparseness which is one of the major concerns when working on WSD.

CBR [1, 7] is a classic example of replicating the working of a brain where brain refers to similar situations as in present and respond likewise. It is noted that the present case is not exactly similar to as stored ones and therefore, we adapt the existing case to present one to find the solution. The concept of similarity function [17, 18, 23] is to understand how close we are with respected to the meaning that we have understood of the given ambiguous word. As the total cases are divided into two sets – one which represent the database i.e. stored cases and the other set – represents the testing cases. The similarity functions help to determine the closeness between the two. The bigrams [21] represents the abstruse word along with either its predecessor word or its successor word. In this paper, the author has compared the results obtained by applying the classifiers – Naive Bayes and Decision Tree on bigrams to decipher the correct meaning of the

given abstruse word. The concept of Case Based Reasoning [16, 17, 18], works on the idea of using previous similar cases to help decipher the meaning of the new cases. In the papers studied with this concept, the authors have identified the groupings of words, also called feature sets, to helps decipher the meaning of the abstruse words in the new cases.

The papers [22, 23] are the base papers for this work. In the paper [22], the authors have given the architecture of using CBR for the implementation of WSD. The paper concludes that ambiguous words with lesser context and having a larger context window will give better results. And in the paper [23], the authors have applied Euclidean similarity function to extract the similar cases with respect to the given input vector. Six different vectors (pre-bigram, post-bigram, pre-trigram, in-trigram, post-trigram and n.-gram) were experimented with. These vectors were then subjected to 3 classifiers, Naïve Bayes, k-NN and Decision Tree. 20 sentences (vectors) were used to help to decipher the correct context of four ambiguous words. The best results were shown by the Decision Tree classifier with 84.88% using pre-bigram vector, followed by Bayes classifier Tree using pre-bigram vector and then k-NN using n-gram vector, where 4 features were used.

3. CASE BASED REASONING

The Case Based Reasoning (CBR) is based on the principal of referring to old similar cases in order to find the solution to problem at hand. A repository of cases is prepared and when a new problem is given, the previous similar cases are retrieved, their solution sought, and if necessary, the solution of previous problem is adapted to find the solution for new problem. Then this new case is stored in the repository for future reference. Aamondt and Plaza [1], proposed the CBR cycle which defines the four important phases of CBR: retrieve, reuse, revise, and, retain

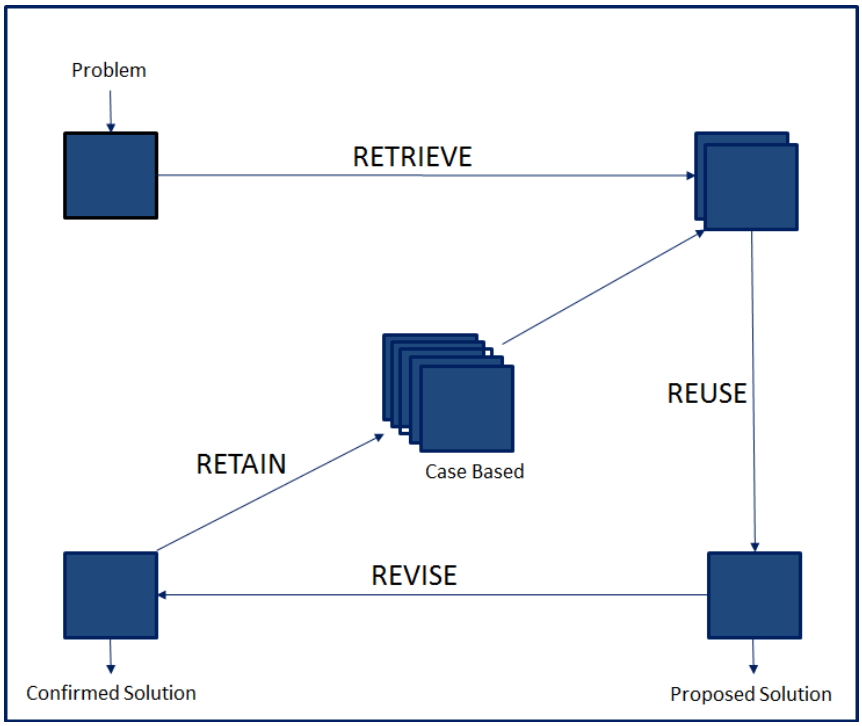

Fig. 1 Four phases of Case Based Reasoning (CBR) Cycle

A. Retrieve

Retrieval is the first phase of the CBR cycle where we search for similar case(s) which can be used to solve the given case in hand. The process starts with the given case's description and ends when we find previous similar case(s). A similarity function is used to calculate the similarity index between the cases. There are different similarity measuring metrics available like, Euclidean, City Block, Cosine, Correlation, Hamming, etc.

B. Reuse

The next phase is reuse where a solution to the given problem is proposed from the retrieved case(s). This phase works in two ways – method reuse and solution reuse. In method reuse, we use the procedure or method used by previous similar case(s) in order to arrive at new solution. This is helpful when we did not get a previous case similar to the present problem. In solution reuse, we can either reuse the solution of previous case if we have an exact match or we use the knowledge from previous case to arrive to the solution of new case.

C. Revise

Revise phase is essential as it evaluates the retrieved solution. This is done by testing the new solution in the real world. And if required the new solution is revised so as to accommodate as per the problem.

D. Retain

In this phase, the solution is stored along with the given problem as knowledge for future reference.

3. VECTORIZATION

The process of vectorization means to identify semantic markers alongside the ambiguous word. The semantic markers denote the words that are placed on the left hand side and on the right hand side of the given ambiguous word in the given case. For our study we are taking 2 words on the left side (L_{w1}, L_{w2}) and 2 words on the right side (R_{w1}, R_{w2}) along with the given ambiguous word. The Feature Vector Representation is defined in the table (Table 1) given below:

Table 1: Feature Vector Representation

Column	**Field**	**Explanation**
C1	Case	Ambiguous Word
C2	Sense_Value	Sense Value
C3	Sense_Tag	Sense Tag
C4	Lw_1, L_{w2}	Weight of two left words
C5	W	Weight of ambiguous word
C6	Rw_1, R_{w2}	Weight of two right words

After identifying the ambiguous words in the given corpus along with their semantic markers, we need to identify the number of features that we want to study for our case analysis. In this paper, we will be implementing the case based model by using 5 different vectors which would have two types of inputs, bigram and trigram. The difference would lie in the grouping of the words for input. Table 2 gives the interpretation of these different vector representations. The pre-bigram vector symbolizes the abstruse word followed by the immediate next word, is represented by T1. The post-bigram vector symbolizes the abstruse word followed by the immediate previous word, is represented by T2. The pre-trigram vector symbolizes the abstruse word followed by the immediate next two words, is represented by T3. The in-gram vector symbolizes the abstruse word with the preceded word and followed by the immediate next one word, is represented by T4. The post-trigram vector symbolizes the abstruse word along with the two previous words.

Table 2: Feature Types

	Feature	**F1**	**F2**	**F3**	**Features Taken**
T1	Pre-bigram	W	R_{w1}	-	2
T2	Post-bigram	L_{w1}	W	-	2
T3	Pre-trigram	W	R_{w1}	R_{w2}	3
T4	In-trigram	L_{w1}	W	R_{w1}	3
T5	Post-trigram	L_{w1}	L_{w2}	W	3

4. DIFFERENT CLASSIFIERS

In the paper, we have taken four different classifiers – Naïve Bayes, k-Nearest Neighbor, Decision Tree and Artificial Neural Network. The idea of using a classifier is to map the given input to obtain a discrete output value. The 3 classifiers, namely Naive Bayes, k-Nearest Neighbor and Decision Tree are supervised in nature while Artificial Neural Network is an un-supervised algorithm. The supervised approach dictates

learning from the training set and then implementing on the given data set whereas the un-supervised approach dictates of understanding the underlying structure of the given data set and then mapping the new data onto it. We are briefly discussing the 4 classifiers:

A. Naïve Bayes Classifier

The Naïve Bayes classifier is a statistical classifier, derived from Bayes Theorem. The classifier states that for a given attribute (in our case, vector), say X, and the value of X does not in any way interfere with any other attributes of the given class, C. In other words, the attribute X is independent of all the attributes belonging to the given class C. This property is known as the class conditional independence.

For the a given data set, say D, the posteriori probability of hypothesis H, P(H|D) according to Bayes Theorem is given by:

$$P(H \mid D) = (P(D \mid H).P(H)) \,/\, P(D)$$

where:

P(H/D) - the probability that the hypothesis holds given the observed data sample D

P(H) - prior probability of hypothesis H

P(D) - probability that sample data is observed

P(D|H) - probability of observing the sample D, given that the hypothesis holds

B. K-NN Classifier

The K-Nearest Neighbor is a supervised approach used to process the data. This classifier helps in locating the closest context of the given the ambiguous word. It mathematically calculates the distance between the given ambiguous word and the different meanings of the word available. The distance helps in forecasting the predicted meaning of the ambiguous word.

The following algorithm illustrates the steps required to find the value of "k" which determines the closeness between the actual and the predicted meaning of the ambiguous word.

Algorithm

Step 1: Start

Step 2: Input - sentence with ambiguous word

Step 3: Remove - stop words

Step 4: Vectorization – from given input

Step 5: For a given ambiguous word in its k^{th} sense

DO

Calculate the distance of the testset w.r.t. set of surrounding words using similarity function

Step 6: Form List - in descending order of the distances between the test data and set of surrounding words.

Step 7: Selection of *k* - such that k > 0.

Step 8: Select - the 'k' nearest neighbor

Step 9: Select from the given list the training vector which is nearest to the given test vector.

Step10: Stop

C. Decision Tree Classifier

Decision tree is one of the popular machine learning algorithms. The reason being that a decision tree subliminally performs variable screening or feature selection. The idea is to find out the set of features that can help in easily deciphering the ambiguous word and then help in developing the decision tree. The process of vectorization that we are using helps in forming different decision tree based on the number of words that we are picking surrounding the given ambiguous word. The traversal from one level to another depends on the addition of words. This is directly dependent on the fact that using how many surrounding words we are able to decipher the correct meaning.

A. ANN Classifier

Neural networks are probable in providing a systematic approach for arriving at a solution, similar to how a human being will do, thus the integration of ANN into the case retrieval phase of the CBR will help in finding better similar case(s) for the given problem.

The Artificial Neural Network (ANN) is an instance-based knowledge acquisition structure which is a potent data modelling tool specifically when the basic correlation among the data set is unknown. ANNs can categorize and absorb interrelated patterns between input data sets and equivalent target values.

Characteristics like generalization, parallelism, robustness, and adaptability are essential for solving real problems.

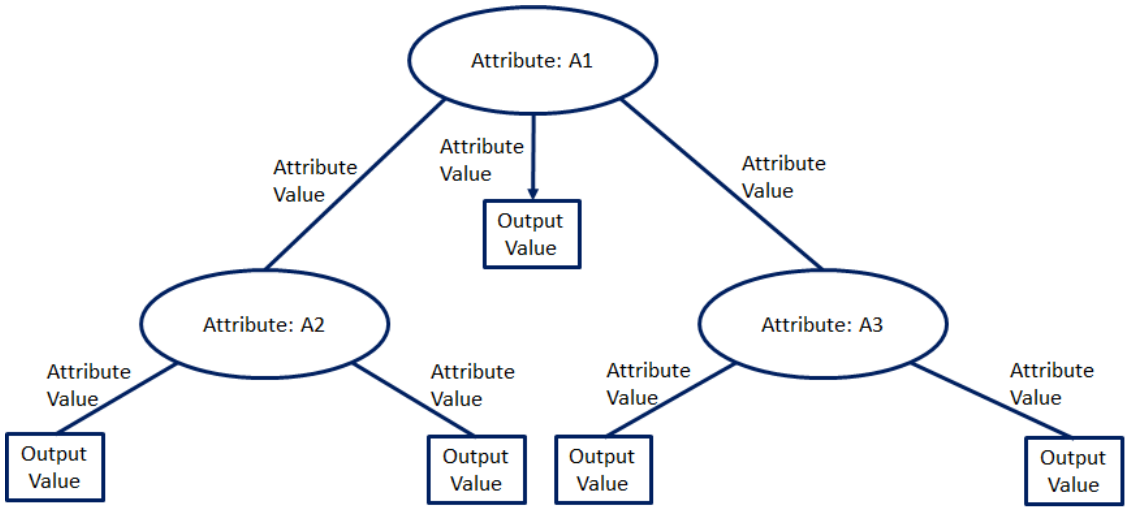

Fig. 2 Decision Tree with nodes signifying the attribute and its value along with branches showcasing the attribute value

ANN is a collection of nodes, which are organized into layers. Every node in a layer is connected to a node in the next layer through a weighted connection. The ANN has three layers, namely input layer, which contains as many nodes as the number of inputs, followed by hidden layer, which has arbitrary number of layers and also arbitrary number of nodes corresponding to the input modification, and finally the output layer, which has as many nodes as the number of outputs.

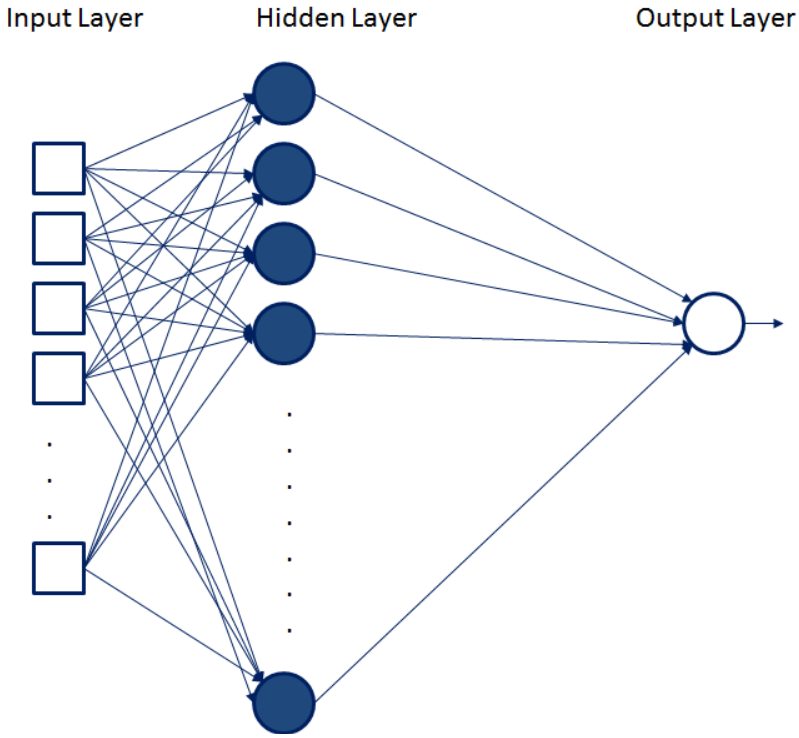

Fig. 3 A general description of Artificial Neural Network with hidden layer responsible for converting the inputs into outputs

The activation function is primarily used to convert the input into the output. In ANN we multiply the inputs with their corresponding weights and apply activation function to it, getting an output from the previous layer, which acts as an input to the next layer. For this paper, we are using the Back Propagation Neural Network (BPNN) which is based on the principal of gradient descent. The activation function used in BPNN is differentiable.

We have used four different similarity functions – Euclidean, Cityblock, Cosine and Hamming to measure the nearness of the given sample data with respect to the most similar data stored in the database.

A. Euclidean Distance

In mathematics, the Euclidean distance refers to a straight line between two points in the Euclidean space. Translating this definition with respect to our scenario specifies the distance between correct meaning of the ambiguous word and the meaning deciphered interpreting the context. Mathematically, Euclidean is defined as:

$$D(E) = \sqrt{\sum_{i=1}^{n}(Xi - Yi)^2} \quad (1)$$

B. Cityblock Distance

It is also referred as the Manhattan Distance and it denotes the absolute distance value between two points. In our case the two points refer to the actual meaning of the ambiguous word and the predicted one. Mathematically, City Block is defined as:

$$D\ (C) = \sum_{i=1}^{n} |Xi - Yi| \quad (2)$$

C. Cosine

Cosine is the measure of the angel between two vectors. It shows promising results in text mining and in our case we can feed different meanings of the ambiguous word and the predicted meaning. The angel between the different meanings and the predicted meaning will be recorded. The smallest angel would signify the result concluded. Mathematically, Cosine is defined as:

$$D\ (COS) = \frac{\sum_{i=1}^{n} Xi\ .Yi}{\sqrt{\sum_{i=1}^{n}(Xi)^2\ .\sum_{i=1}^{n}(Yi)^2}} \quad (3)$$

D. Hamming Distance

The Hamming distance is used to measure the distance between two sequences or in other words, the number of substitutions that can be made such that both the strings are same. In our case, the Hamming Distance helps in determining the approximate window size such that the predicted meaning can be deciphered as close to the actual meaning possible.

5. CASE BASED MODEL

The model that has been used in this paper comprises of four steps, shown in the figure (Fig. 4) below:

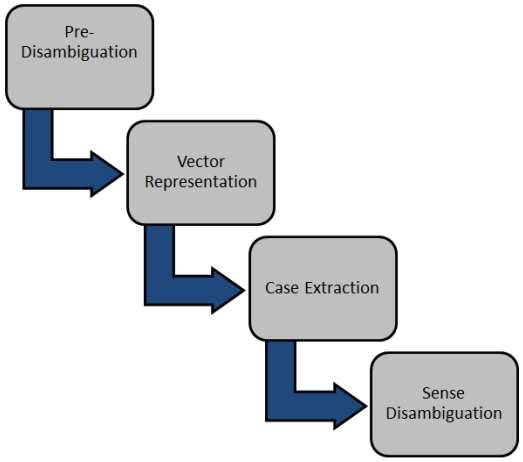

Fig 4: The Case Based Model representing the step-based process to decipher the correct meaning of the ambiguous word

The each step in the proposed model is explained below:

A. Pre-Disambiguation

The pre-disambiguation process refers to removal of stop words from the corpus. The Punjabi language has 184 classified stop words [10] which includes prepositions, conjunctions, etc. which when removed gives us "bag of words". These bags of words are then converted into vectors.

B. Vector representation

For our paper, we have used 5 different vectors (refer Table 2), namely, pre-bigram, post-bigram, pre-trigram, in-trigram, and post-trigram.

C. Case Extraction

Following this we move onto case extraction which has further 3 steps – firstly we identify the ambiguous word and in this paper, we took 4 ambiguous words. Secondly, we pull out all the similar examples having ambiguous words that we have sense-tagged and reserved as stored cases. Third and the final step is to use the similarity functions to draw out similar cases.

D. Sense Disambiguation

The final step of case based model is sense disambiguation where we are using 4 classifiers to decipher the correct meaning of the ambiguous word.

6. EXPERIMENTATION & RESULT

To carry out the experimentation, we have used the Punjabi Corpora which was acquired from Evaluations and Language Resources Distribution Agency, Paris, France [8] which has been sense-tagged with 100 ambiguous words**.** To highlight the importance of disambiguation we took four words of Punjabi having more than 2 different meanings to check whether we are able to decipher the correct meaning in the given sentence (vector). Table 3 gives the Punjabi ambiguous word along with its English translation and different contexts. We have acquired the meanings for these words from Punjabi WordNet [7, 9], an online repository created by Centre for Indian Language Technology (CFILT), Computer Science and Engineering Department, IIT Bombay, Mumbai, Powai.

Table 3: Punjabi Words with different senses

Punjabi Word	English Translation	Sense 1	Sense 2	Sense 3	Sense 4	Sense 5
ਉੱਤਰ	uttar	answer to a question	north direction	dislocation of a joint, shoulder or knee	descending the stairs	-
ਕੱਚਾ	kachha	undercooked food	lacking practical training or experience	not ripe	able to be eradicated or rooted out	not complete knowledge
ਹਾਰ	haar	unsuccessful ending to a struggle or contest	jewelry consisting of a chord or chain	the act of inflicting corporal punishment with repeated blows	-	-
ਉਲਟ	utla	altogether different in nature or quality or significance	meaning is opposite to original word	-	-	-

The process of retrieval of the correct sense of the given ambiguous word can be traced out with the help of the proposed algorithm. To implement the algorithm, we have divided the 100 sentences taken from the Punjabi Corpora, having the said 4 ambiguous words into two sets:

1. First Set - 70 sentences. Used for building the various cases for word sense disambiguation and stored the cases for CBR.
2. Second Set – 30 sentences. Used for experimentation to see how the classifiers retrieve the cases and decipher the word sense in the context itis used.

A. Algorithm

The steps of the algorithm are as follows:

Step 1: Start

Step 2: Create a case repository to carry out CBR

Step 2.1: The first set of 70 sentences were used to build the case repository database along with their sentences with respect to their meaning in the sentence

Step 2.2: The second set of 30 sentences were picked one by one to find the meaning of the 4 ambiguous words in the current context in the sentence

Step 3: Select a sentence having the ambiguous word from 30 sentences.

Step 4: Apply pre-disambiguation process i.e. removal of stop words

Step 5: Convert given sentence into vector

Do

Step 5.1: With respect to the ambiguous word, create a pre-bigram i.e. ambiguous word plus immediate next word

Step 5.2: With respect to the ambiguous word, create a post-bigram i.e. ambiguous word plus immediate previous word

Step 5.3: With respect to the ambiguous word, create a pre-trigram i.e. ambiguous word plus immediate next two words

Step 5.4: With respect to the ambiguous word, create a in-gram i.e. ambiguous word plus one previous word and one next word

Step 5.5: With respect to the ambiguous word, create a post-trigram i.e. ambiguous word plus two previous words

Step 6: Use Similarity Functions to extract similar cases

Step 6.1: Euclidean - The similarity is calculated by finding the distance between correct meaning of the ambiguous word and the meaning deciphered interpreting the context.

Step 6.2: Cityblock – The similarity is calculated by referring the two points as the actual meaning of the ambiguous word and the predicted one.

Step 6.3: Cosine – The similarity is calculated by finding the angel between the different meanings and the predicted meaning of the given ambiguous word.

Step 6.4: Hamming – The similarity is calculated by determining the approximate window size such that the predicted meaning can be deciphered as close to the actual meaning as possible.

Step 7: Use classifiers to decipher the correct meaning of the ambiguous word from the given vector and selected case.

Step 7.1: Naïve Bayes – This classifier helps in locating the closest context of the given ambiguous word by finding an attribute independent of all the attributes in the vector.

Step 7.2: k-NN - This classifier helps in locating the closest context of the given ambiguous word by calculating the distance between the given ambiguous word and the vector found in the case repository.

Step 7.3: Decision Tree – This classifier helps in locating the closest context of the given ambiguous word by finding the set of features that helps in deciphering the correct context.

Step 7.4: ANN – This classifier helps in locating the closest context of the given ambiguous word by using the Back Propagation Neural Network (BPNN) which is based on the principal of gradient descent.

Step 8: Verified the deciphered meaning of the ambiguous word in the current context and gathered the accuracy data for each of the classifiers

Step 9: Stop

B. Process

The process is followed as described below:

In order to understand how this algorithm is implemented, consider the Punjabi ambiguous word, "kachha)". The word "kachha" had the maximum contexts of all the four words taken for experimentation for this paper. This primarily became the reason of showcasing the results of this word. Table 4 shows the five different contexts of this word, along with the sentences to show its usage. These sentences are part of the repository created to apply the case based reasoning approach to decipher the correct meaning

Table 4: Ambiguous word "kachha" usage in sentences

Context	Meaning	Punjabi Sentence	English Translation
1	undercooked food	ਕੁਝ ਕੱਚੀਆਂ ਸਬਜ਼ੀਆਂ ਸਲਾਦ ਦੇ ਰੂਪ ਵਿਚ ਖਾਦੀਆਂ ਜਾਂਦੀਆ ਹਨ।	Some raw vegetables are eaten in the form of salad.
2	lacking practical training or experience	ਅਨੁਭਵ ਵਿਚ ਕੱਚਾ ਹੋਣ ਦੇ ਕਾਰਣ ਰਾਮੂ ਨੂੰ ਨੌਕਰੀ ਨਹੀਂ ਮਿਲੀ।	Due to lack of experience, Ramu did not get the job.
3	not ripe	ਰਾਮ ਕੱਚਾ ਅੰਬ ਖਾ ਰਿਹਾ ਹੈ।	Ram is eating unripe mango.
4	able to be eradicated or rooted out	ਇਸ ਸਾੜੀ ਦਾ ਕੱਚਾ ਰੰਗ ਇਕ ਧੋ ਤੇ ਹੀ ਉਤਰ ਗਿਆ ।	This saari's colour faded out after one wash only.
5	not complete knowledge	ਤੁਸੀ ਇਸ ਕੰਮ ਵਿਚ ਕਿਸੇ ਕੱਚੇ ਵਿਅਕਤੀ ਦੀ ਰਾਇ ਨਾ ਲਵੋ	Do not take advice of an inexperienced person for this work.

To illustrate further, consider the following sentence:

ਜ਼ਿਆਦਾਤਰ ਕੰਪਨੀਆਂ ਕੱਚਾ ਮਾਲ ਅਯਾਤ ਕਰਦੀਆਂ ਹਨ।

After removing the stop words from the sentence, we convert it into vectors. Table 5 shows the five different vectors created for the given sentence.

Table 5: Five different vectors created for the given sentence

	Feature	F1	F2	F3
T1	Pre-bigram	ਕੱਚਾ	ਮਾਲ	-
T2	Post-bigram	ਕੰਪਨੀ	ਕੱਚਾ	-
T3	Pre-trigram	ਕੱਚਾ	ਮਾਲ	ਅਯਾਤ
T4	In-trigram	ਕੰਪਨੀ	ਕੱਚਾ	ਮਾਲ
T5	Post-trigram	ਜ਼ਿਆਦਾ	ਕੰਪਨੀ	ਕੱਚਾ

Following this, the Similarity Functions are used to extract similar cases. Then we use classifiers to decipher the correct meaning of the ambiguous word from the given vector and selected case.

C. Experimentation

For this paper, we have taken four Punjabi ambiguous words, which were subjected through different classifiers and the results obtained are illustrated in the given tables. Table 6 shows the results displayed on using Naïve Bayes classifier. Table 7 shows the results displayed on using K-NN classifier. Table 8 shows the results displayed on using Decision Tree classifier. Table 9 shows the results displayed on using ANN classifier.

Table 6: Accuracy obtained by using Naïve Bayes Classifier

Punjabi Word	Similarity Function	T1	T2	T3	T4	T5
ਉੱਤਰ	Euclidean	83.03	82.2	81.66	82.45	80.21
	Cityblock	82.76	81.43	81.8	76.4	81.2
	Cosine	81.67	80.6	79.9	77.7	79.65
	Hamming	80.44	80.12	78.8	77.45	78.48
ਕੱਚਾ	Euclidean	83.23	82.23	84.23	84.23	84.23
	Cityblock	83.22	81.33	83.9	83.9	83.4
	Cosine	83.4	82.3	82.5	80.45	83.24
	Hamming	81.98	80.79	81.93	82.67	81.34
ਹਾਰ	Euclidean	85.19	84.12	83.67	86.12	84.12
	Cityblock	83.21	83.11	82.78	83.67	82.4
	Cosine	84.2	83	83.6	83.65	81.66
	Hamming	83.45	81.2	82.8	82.4	81.67
ਉਲਟ	Euclidean	84.54	82.54	84.54	84.67	83.66
	Cityblock	81.34	82	83.67	81.8	81.4
	Cosine	80.3	81.67	83.1	82.1	81.77
	Hamming	81	79.45	86	81.43	81.69

Table 7: Accuracy obtained by using K-NN Classifier

Punjabi Word	Similarity Function	T1	T2	T3	T4	T5
ਉੱਤਰ	Euclidean	77.18	68.09	77.41	77.98	72.02
	Cityblock	78.2	67.4	79.8	70	73.4
	Cosine	75.34	69.4	74.6	71.9	71.78
	Hamming	73.4	61.4	75.4	71.32	71.6
ਕੱਚਾ	Euclidean	77.18	68.09	77.41	76.98	72.02
	Cityblock	78.2	74.6	77.4	74.4	74.9
	Cosine	74.2	84.2	71.4	76.3	54.45
	Hamming	75.6	71.45	73.4	73.45	78.45
ਹਾਰ	Euclidean	73.21	74.89	72.39	74.88	75.21
	Cityblock	74.3	76.4	74.45	74.5	73.5
	Cosine	71.9	75.4	73.6	77.88	73.45
	Hamming	72.6	72.6	80.3	73.9	72.45
ਉਲਟ	Euclidean	73.27	75.22	74.45	76.32	76.12
	Cityblock	74.7	73.4	77.99	77.4	78.5
	Cosine	71.3	78.45	73.8	73.4	73.45
	Hamming	71	72.6	71.9	72.67	73.4

Table8: Accuracy obtained by using Decision Tree Classifier

Punjabi Word	Similarity Function	T1	T2	T3	T4	T5
ਉੱਤਰ	Euclidean	83.23	82.2	82.2	82.2	82.2
	Cityblock	80.45	83.2	81.1	81.4	81.2
	Cosine	82.4	81.9	81.9	81.4	81.8
	Hamming	81.2	79.6	80.9	81.45	81.3
ਕੱਚਾ	Euclidean	84.21	82.2	82.2	82.2	82.2
	Cityblock	83.89	83.2	84.4	82.12	83.2
	Cosine	81.5	81.5	83.6	82.2	83.25
	Hamming	82.49	82.45	79.11	81.8	82.45
ਹਾਰ	Euclidean	86.1	86.1	85.9	85.98	86.02
	Cityblock	85.45	85.9	84.44	84.4	82.6
	Cosine	85.5	85.34	86	83.45	82.49
	Hamming	81.5	83.33	83.6	82.67	79.56
ਉਲਟ	Euclidean	85.54	84.44	86.54	84.24	85.54
	Cityblock	86.3	82.2	85	81.3	84.9
	Cosine	82.9	82.67	84.4	82.3	81.2
	Hamming	82.11	82.2	82.99	81.3	82.55

Table 9: Accuracy obtained by using ANN Classifier

Punjabi Word	Similarity Function	T1	T2	T3	T4	T5
ਉੱਤਰ	Euclidean	73.6	69.8	76.3	72.4	80.89
	Cityblock	74.3	69.8	77.2	74.6	79.6
	Cosine	73.9	66.7	73.4	78.4	79.34
	Hamming	71.6	61.4	73.2	71.6	79.23
ਕੱਚਾ	Euclidean	78.4	82.34	77.4	78.4	81.3
	Cityblock	75.4	80.67	77.9	77.45	81.6
	Cosine	77.6	80.3	74.5	77.9	80.67
	Hamming	76.9	84.67	79.45	76.9	82.32
ਹਾਰ	Euclidean	74.1	77.29	81.34	75.11	81.34
	Cityblock	74.4	77.4	81.9	73.2	79.4
	Cosine	72.1	75.39	82.4	74.45	79.12
	Hamming	72.9	74.5	79.0	72.5	82.43
ਉਲਟ	Euclidean	74.0	81.7	73.2	76.3	76.1
	Cityblock	74.8	81.7	74.9	77.4	79.4
	Cosine	75.1	80.34	71.6	77.4	73.45
	Hamming	74.1	80.49	71.39	71.4	73.67

D. Result

To understand the results better, we have plotted graphs with respect to the 4 classifiers. Off the 4 words chosen for the disambiguation process, we are plotting graphs for the word “kachha”. The idea was to understand how by using different similarity functions for a given classifier alter the decision making capability.

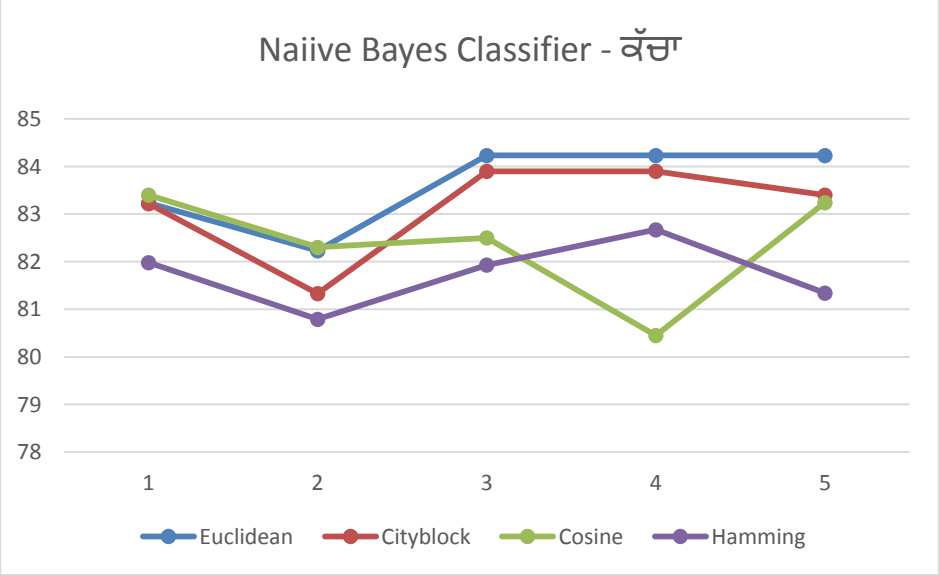

Fig. 5: Results shown by Punjabi ambiguous word “kachha” using Naive Bayes Classifier with respect to

Accuracy obtained and number of different senses

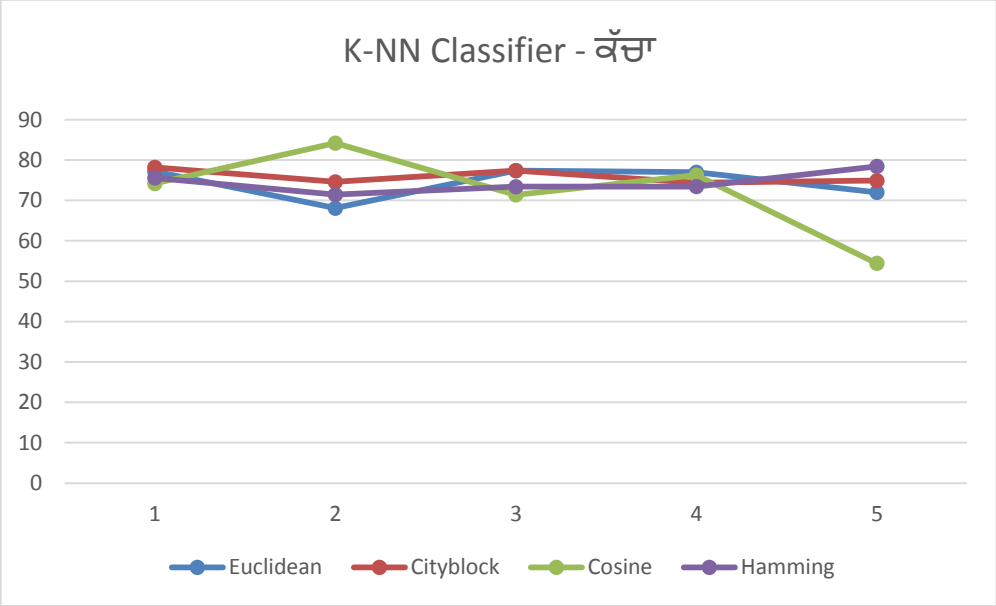

Fig. 6: Results shown by Punjabi ambiguous word “kachha” using k-NN Classifier with respect to Accuracy obtained and number of different senses

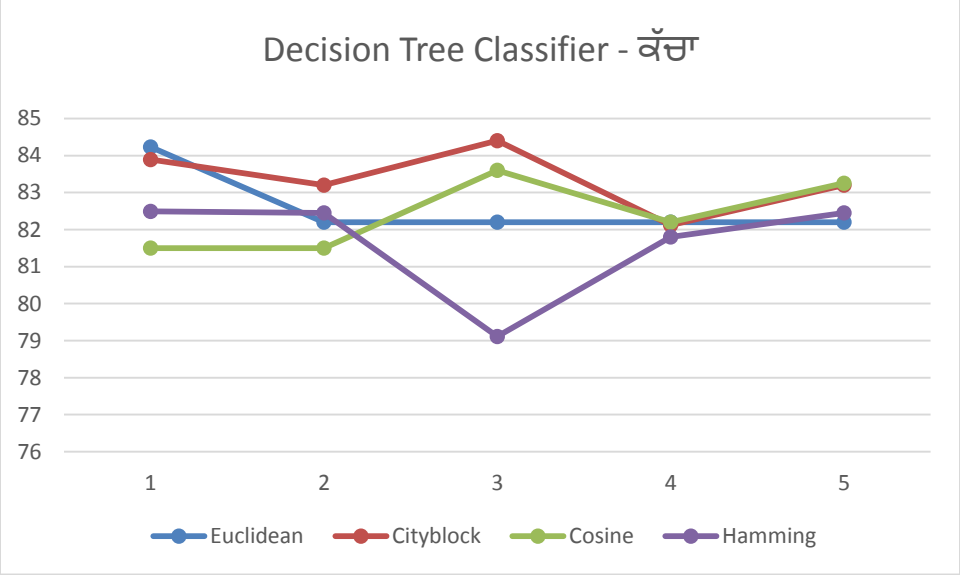

Fig. 7: Results shown by Punjabi ambiguous word “kachha” using Decision Tree Classifier with respect to Accuracy obtained and number of different senses

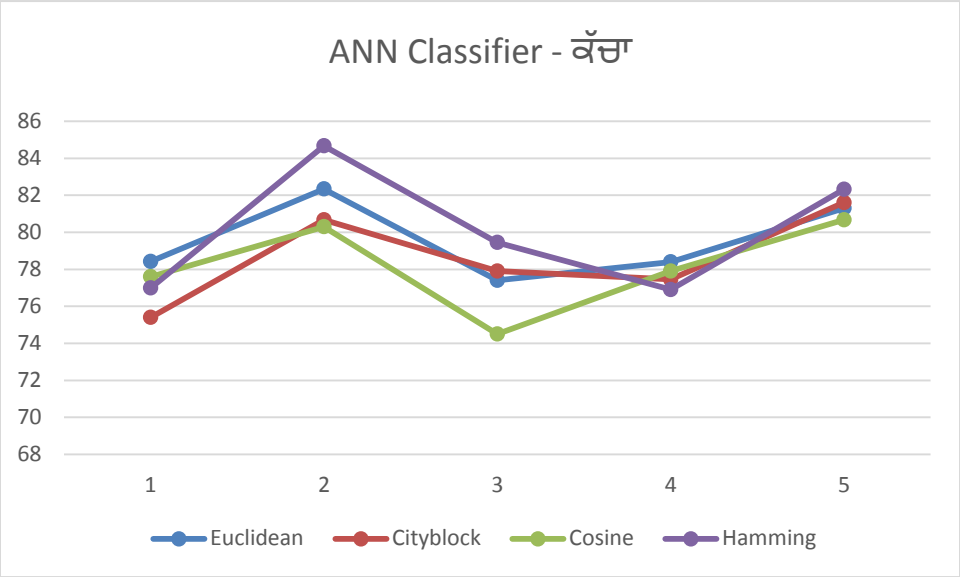

Fig. 8: Results shown by Punjabi ambiguous word "kachha" using ANN Classifier with respect to Accuracy obtained and number of different senses

The observations made on the word "kachha" were broadly seen in other three words as well. This helped in generalizing the conclusion.

7. CONCLUSION

To write the results, we plotted the graph for the word "kachha" with respect to the four classifiers. The conclusion drawn from the experimental results were as follows: For T1 (Pre-bigram) the highest accuracy was observed in Euclidean metrics using decision tree classifier. For T3 (Pre-trigram) the highest accuracy for the four words used for disambiguation was observed in Euclidean metrics using Decisions Tree classifier. For T4 (In-trigram) the highest accuracy was observed in Euclidean metrics using Naive Bayes classifier. For T2 (Post-bigram) and T5 (Post-trigram) the results were inconclusive. And of the 3 results (T1, T3, T4), T4 (84.23%) showcases the highest accuracy i.e. using Euclidean metrics with Naive Bayes classifier. This result is different from the base paper [23] used and this can be attributed to the fact that in base paper, 20 sentences (vectors) from the corpus were taken whereas in this paper we took 30 sentence (vectors). This is also the primary reason of having inconclusive results for T2 and T4. If we further increase the number of sentences then the results can be more conclusive. Due to limited work done in Punjabi WSD, this work will help in improvising the disambiguation process.

ACKNOWLEDGEMENTS: This work was supported by Amity Institute of Information Technology, Amity University Uttar Pradesh, where the author[1] is registered as scholar and is grateful for the support.

RECEIVED: MAY, 2019.
REVISED: DECEMBER, 2019.

REFERENCES

[1] AAMODT A. and PLAZA E. (1994): Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. **AI Communications, IOS Press**, 7, 39-59.

[2] BANSAL M (2015): Word Sense Disambiguation: Literature Survey for Indian Languages. **International Journal of Advanced Research in Computer Science and Software Engineering**, 5, Issue 12.

[3] BISWAS S. K., BARUAH B., SINHA N. and PURKAYASTHA B. (2015): A hybrid CBR classification model be integrating ANN into CBR. **International Journal Services Technology and**

Management, 21, Nos. 4/5/6.
[4] CHEN D. and BURRELL P. (2001): Case-Based Reasoning System and Artificial Neural Networks: A Review. **Neural Computer & Applications**, 10, 264-276.
[5] KAUR J. and GUPTA V. (2010): Effective Approaches for extraction of Keywords. **International Journal of Computer Science**, 10, 144-148.
[6] KAUR J. and SAINI J. R. (2016): Punjabi Stop Words: A Gurmukhi, Shahmukhi and Roman Scripted Chronical. **Proceedings of the ACM Symposium on Women in Research,** 32-37.
[7] KOLODNER J. L. (1992): An Introduction to Case-Based Reasoning. **Artificial Intelligence Review**, 6, 3-34.
[8] KUMAR R. and KHANNA R. (2011): Natural Language Engineering: The Study of Word Sense Disambiguation in Punjabi. **International Journal of Engineering Sciences**, 1, 230-238.
[9] KUMAR R., KHANNA R. and GOYAL V. (2012): A Review of Literature on Word Sense Disambiguation. **International Journal of Engineering Sciences**, 6, 224-230.
[10] NARANG A., SHARMA R. K. and KUMAR P. (2013): Development of Punjabi WordNet. **CSI Transactions on ICT**, 1, 349-354.
[11] NAVIGLI R. (2009): Word Sense Disambiguation: A Survey. **ACM Computing Surveys**, 41, 10:1-10:69.
[12] PEDERSEN T. (2001): A decision tree of bigrams is an accurate predictor of word senses. **Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics**, **Pittsburgh.**
[13] BHATTACHARYYA P (2013): Punjabi Wordnet. Available: http://tdil-dc.in/indowordnet/index.jsp **Consulted** 16-8-2018.
[14] BAKER P. and HARDIE A. (2004): The EMILLE Corpus incorporating the CIIL Corpora. Obtained from Evaluations and Language Resources Distribution Agency, Paris, France. Available: http://catalog.elra.info/en-us/repository/browse/ELRA-S0408/ **Consulted** 16-8-2018
[15] SINGH S. and Siddiqui T. J. (2012): Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation. **International Conference on Information Retrieval & Knowledge Management, Malaysia**.
[16] TAMILSELVI P. and SRIVASTSA S. K. (2011): Case Based Word Sense Disambiguation Using Optimal Features. **International Conference on Information Communication and Management**, **Singapore.**
[17] TAMILSELVI P. and SRIVASTSA S. K. (2011): Word Sense Disambiguation using Case Based Approach with minimal Features Set. **Indian Journal of Computer Science and Engineering**, 2, 628-633.
[18] TAMILSELVI P. and SRIVASTSA S. K. (2011): Quantifying the finest similarity for Case Based reasoning to implement Word Sense Disambiguation using Different Learning Classifiers. **Journal of Computer Applications Research and Development**, 1, 49-56.
[19] WALIA H., RANA A. and KANSAL V. (2017): Different Techniques Implemented in Gurumukhi Word Sense Disambiguation. **International Journal of Advanced Technology in Engineering and Science**, 5, 40-46.
[20] WALIA H., RANA A. and KANSAL V. (2017): A study on different Word Sense Disambiguation Approaches and their application on Indian Regional Languages. **International Conference on Technology and Trust, Greater Noida.**
[21] WALIA H., RANA A. and KANSAL V. (2018): Word Sense Disambiguation: Supervised Program Interpretation Methodology for Punjabi Language. **7th International Conference on Reliability, Infocom Technologies and Optimization, Noida**.
[22] WALIA H., RANA A. and KANSAL V. (2019): Case Based Interpretation Model for Word Sense Disambiguation in Gurmukhi. **9th International Conference on Cloud System and Big Data Engineering, Noida.**
[23] WALIA H., RANA A. and KANSAL V. (2019): Case Based Construal using Minimal Features to decipher ambiguity in Punjabi Language. **International Conference on Artificial Intelligence, Dubai**.