Case Based Interpretation Model for Word Sense Disambiguation in Gurmukhi

Himdweep Walia
Department of CSE
DTC
Gr. Noida, India
himdweep@yahoo.com

Ajay Rana
Amity Technical Placement Centre
Amity University
Noida, India
ajay_rana@amity.edu

Vineet Kansal
Department of CSE
IET
Lucknow, India
vineetkansal@yahoo.com

***Abstract* – Language is a medium through which we can communicate our thoughts with each other. To duplicate the same communication between a human and a machine, we require Natural Language Processing. Under Artificial Intelligence, natural language processing is one of the major case study. What makes this case study more complex is the fact that a word has multiple meanings and the way in which the word is being used (context) explains the meaning of the word. This phenomenon where a single word could have multiple meaning and the correct meaning is deduced from the context it is being used as is known as Word Sense Disambiguation. In this paper, we are using Case Based Reasoning interpretation model for word sense disambiguation on Indian Regional Language - Gurmukhi, popularly known as Punjabi. The basic idea behind case based reasoning approach is to apply the solution of previously solved problems in finding the solutions for new problems. The inspiration for the case based reasoning approach came from the role of reminding in human reasoning.**

Keywords – Natural Language Processing, Word Sense Disambiguation, Case Based Reasoning, WSD in Punjabi, Sense-tagged corpora

I. INTRODUCTION

Word Sense Disambiguation (WSD) is an imperative study under Natural Language Processing (NLP). Since every language has ambiguous words, it is important to understand that in which context the word is being used and this information becomes necessary in natural language understanding, natural language generation, speech recognition, machine translation and other similar domains of NLP. To illustrate the meaning of the word "ambiguous", consider the word "bimonthly". It can mean twice each month or every two months. In a similar way, every spoken / written language have many ambiguous words which can be a little complicated for a machine to easily figure out. For our study, we are working on Gurmukhi, and it too has many ambiguous words. Consider the word, "uttar", which has four meanings, "climbing down", "answer", "direction" and "dislocation of bone".

The task of making a machine understand the correct context of the given ambiguous word involves the use of specialized algorithms. These algorithms could be either time consuming when dictionary-based techniques are implemented or difficult to formalize when using machine learning techniques. In this paper, we are using Case Based Reasoning (CBR) for WSD which uses the concept of referring to old similar cases in order to give the solutions to new problems.

The remaining paper is divided into the following sections: Section II discusses the work done in Gurmukhi (popularly known as, Punjabi) language and other related work in CBR. Section III explains the Case Based Interpretation Model using minimal feature set in WSD. The following section discusses the CBR Cycle, and the next section discusses CBR Cycle's execution and implementation of the system. After this Section VI showcases the results and the final section concludes the paper.

II. LITERATURE REVIEW

WSD has been a topic of research in all languages over the last few years [2, 3]. Lexical resources have been developed for English, European Languages, Asian languages and even Indian languages [4, 5]. For this paper, our language of focus is Gurmukhi [6, 7], spoken in the state of Punjab and some parts of its neighbouring states. Various knowledge-based and machine learning based algorithms have been implemented to find the right context of the ambiguous word [7].

The common problems with NLP are data sparseness and inconsistency in vocabulary. To remove the inconsistency, WSD algorithms are applied by considering the adjacent words in the sentences or highly repeated words in the document. To remove ambiguity of a word and infer the right context in which the word is used, the supervised approaches and unsupervised approaches can be applied [1]. The supervised algorithms require datasets that are annotated i.e. marked, giving the meaning of the ambiguous word for training while in the case of unsupervised algorithms we do not need any training, therefore the need of annotated dataset is not there.

In his paper, T. Pedersen [14], experimented with the use of bigrams (i.e. two words - one ambiguous word and other pre or post word following the ambiguous word) for WSD using the Decision Tree and the Naive Bayes classifier. In his paper, he studied various combinations of

978-1-5386-5933-5/19/$31.00 ©2019 IEEE

bigrams that appeared near to the ambiguous words (with the range of 40-50 words on either side - right or left, of the ambiguous word) and decided on them as the feature set. He then applied statistical methods to find the meanings i.e. removing the ambiguity from the texts using decision tree with bigram concept.

S. Singh and T. Siddiqui [13], in their paper on Hindi WSD, have discussed the advantages of using the optimal window size, stemming and stop words in finding the right meaning of the given word with better accuracy.

With the increase in the number of features, the sparseness is reduced which means that smoothing is required in order to improve the effectiveness and performance of the disambiguation algorithms. To avoid sparseness, minimal features such as bigram, trigram and n-gram are used. This paper presents an approach to disambiguate words using CBR with minimal features.

In their papers, P. Tamilselvi, and S.K.Srivastsa [15, 16] have used different set of features for word sense disambiguation. In the paper [16], they have considered two feature elements (pre-bigram and post bigram) in each row vector. They have then processed each row vector with k-NN and ANN for finding the right context of ambiguous words. The results for Pre-bigram have been the most promising. In the other paper [15], they have applied three different set of features (bigram, trigram and n-gram) with three different distance measuring functions in disambiguation system for WSD combined with three different classifiers. Here, the best results were shown by pre-bigram feature type with Euclidean distance function together with Bayes' classifier.

III. CASE BASED INTERPRETATION MODEL USING MINIMAL FEATURE SET

The case based interpretation model is built on two basic laws of the nature of the world. The first law states that the world is regular: "similar problems have similar solutions". This means that the starting point of any new problem is always found in the solutions of similar prior problems. The second law states that most of the problems are similar in nature and type and thus repetitiveness is there with minor differences. This means that every new problem is going to be quite similar to past and / or current problems. As the two laws hold, it becomes imperative to have a good understanding of the current reasoning so that they can be effectively be implemented: thus CBR becomes an effective reasoning strategy.

The fact that in our day-to-day lives, we use case-based reasoning to work out our problems. This analogy is being developed in AI also to solve problems. We humans have a tendency to find solution for a given problem in hand even if we do not have sufficient and minimum requirements to work it out. We are able to draw similarities from previous such situations and even with limited and uncertain knowledge we can fathom results. This build up is continuous and steadily we are able to arrive at more precise solutions to our problems and hence becoming better in solving future problems. They use past situations to interpret new situation (much similar to what lawyers do). They create an equitable solution to a new problems (much similar to what labor mediators do). All of these qualities are most needed or desirable for real-world AI systems. Case-based reasoning follows this same logic. In case-based reasoning, the same logic is applied, a reasoner remembers the previous situation which is quite similar to the current situation. The reasoner then uses the previous situation to solve the current problem.

CBR can therefore be pertained in the advancement of AI technology. Here we will be giving an overview of five different problems that can be perfected by using case based interpretation model:

A. Acquiring knowledge or building knowledge base

The foremost concern in a knowledge-based systems is to determine the way through which we can write the rules which would in turn help in taking decisions. The knowledge required to build the rules is laborious and unreliable hence making rule acquisition process hard. Though it is not easy to make rules but the tougher part is that there is no quantitative measures that can help to ensure that the defined rules will give the most accurate results. The complexity of brain is very hard and the way it functions, with the rules it follows, can be difficult to write and can be of enormous proportions.

The process of designing the rules, stems from understanding the specific condition and then trying to break it down into sub-parts such that simple rules can be generated from them. There could be certain conditions that are common in different scenarios and thus are already incorporated thus making the process a little simplified.

The different cases that are available, fall is various domains. It is possible that in some domains all the cases are not listed while in others, the cases could be all there but some of them cannot be processed as they fall under the exceptional category. When cases like these occur then the best possible way is to implement case engineering which helps in streamlining the information and only elaborate the part which has the required information pertaining to it.

B. Upkeep of knowledge

The foremost condition in building a good AI application is creating its initial knowledge base. The process of understanding the problem often lacks clarity which results in repeated updation of the knowledge base. The CBR systems also supports incremental learning. CBR systems can be set up with few limited set of ``seed cases,'' to be enhanced with new set of additional cases if the preliminary case library turns out to be inadequate in practice. A CBR system needs to handle the categories of problems that really happen in practice, whereas generative systems must account for each and every possible combination of problems that might be possible in principle.

C. Enhancing problem-solving competence

People achieve reasonable problem-solving competence regardless of the fact that routine problems in every day

reasoning, such as elucidation and planning, are hard. Reuse of previous elucidations helps in increasing problem-solving competence by building on earlier reasoning's instead of repeating earlier work. In addition to successful cases CBR saves failed scenarios as well so that it can caution of probable problems to evade.

D. *Enhancing quality of solutions*

Rules will be imperfect when the doctrines of a domain are yet to be fully understood. In such situations, the solutions recommended by cases can be more precise than the solutions proposed by chains of rules, as cases imitate what actually occurs (or fails to occur) in a particular set of situations. For example in medical reasoning, narratives about specific cases go beyond classified knowledge and suggest as "as-yet-unorganized suggestion"'.

E. *User acceptability*

A major problem in implementing successful AI systems is users' acceptability. No system is worthwhile without its user's acceptability of its results. To have faith in the system's conclusions, a user should be convinced that the systems conclusions are deduced in a sensible way. This is a problem for other AI approaches: such as neural network systems cannot arrange for reasons of their choices, whereas rule-based systems clarify their choices by referring to their rules, which the user does not totally comprehend or agree. While the fallouts of CBR systems are centered on authentic earlier cases that can be offered to the user to provide convincing and persuasive backing for the system's deductions.

IV. THE CBR CYCLE

Aamondt and Plaza proposed the CBR Cycle which underlines the four important phases of CBR process, namely, retrieve, reuse, revise, and retain, refer Fig. 1.

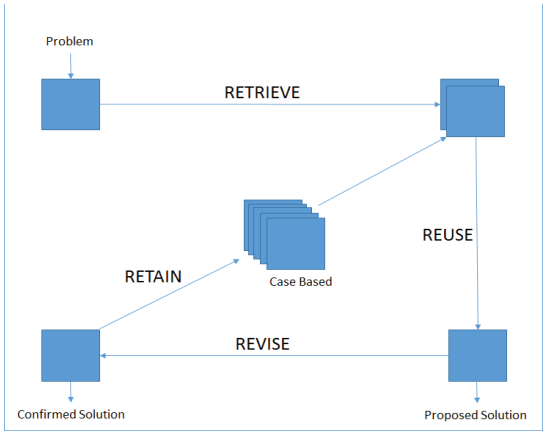

Fig. 1. CBR Cycle

A. *Retrieve*

In Retrieve phase of CBR for a given problem, the closest resembling case available in the case repository / knowledge base is retrieved which assists in solving it. A case is defined as the description of problem, the solution, and the series of steps executed to reach the solution. Consider the scenario where Lalit, for instance, has to make a cup of ginger tea. As Lalit is a novice in the kitchen and can remember making a regular tea only. He uses the same series of steps to prepare ginger tea for himself.

B. *Reuse*

In Reuse phase of CBR the solution from the previous retrieved case is mapped to the target problem to find out the solution. This process may comprise of adjusting the solution as required to fit the new circumstances. For example, Lalit must adjust his retrieved solution to prepare his ginger tea by adding ginger to tea.

C. *Revise*

In Revise phase of CBR cycle the previously mapped solution to the target circumstances is tested in the real world (or a simulation). If required the case is revised with the updated solution and its data. Now Lalit modified his tea recipe by adding ginger after adding the tea leaves. This did not allow the ginger to be incorporated into the tea. This recommends the following revision in the case / recipe that is to add ginger before adding the tea leaves so that ginger is also properly incorporated in the tea.

D. *Retain*

In this retain phase of CBR the solution is successfully modified to the target problem and hence subsequent knowledge as a new case in memory is stored so that it can be reused in future. For example, Lalit now registers his new-found method for making ginger tea, hence enriching his set of stored experiences in knowledge repository which is preparing him for future tea-making requirements.

V. EXECUTION AND IMPLEMENTATION

In this paper we have followed three major steps to disambiguate the given word. The three major steps are - pre-disambiguation process, case extraction, and, disambiguation process.

Fig. 2 CBR process for WSD

The feature vector representation and sense disambiguation are the two main parts of disambiguation system architecture. Feature Vector Representation is defined in the table (TABLE I).

A. *Pre-Disambiguation Process*

Let the input sentence be S with words W_i (i=1…n) words, represented in equation 1.

$$S = \sum_{i=1}^{n} (W_i) \quad (1)$$

In pre-disambiguation process, the given input sentence S is tokenized. All the stop words from the sentence are removed such that we are left with "bags of words".

Now, this bags of words, BOW, will have words, W_j, m>=n, j=1…m, represented in equation 2.

$$BOW = \sum_{j=1}^{m} (w_j, m \geq n) \quad (2)$$

TABLE I FEATURE VECTOR REPRESENTATION

Column	Fields	Description
C1	Case	Ambiguous word
C2	Sense_Value	Sense Value
C3	Sense_Tag	Sense Tag
C4	L_{w1} L_{w2} L_{w3}	Weight of Three Left word
C5	W	Weight of Ambiguous Word
C6	R_{w1} R_{w2} R_{w3}	Weight of Three Right word

In next step we perform curtailing or morphological analysis on the corpus. In this process we reduce the target word whose context is to be found out to its base form which will therefore make it easier for us to disambiguate and find its correct context in which it has been used. Now such ambiguous words in the input sentence are secluded using marked corpus [9] and Indo WordNet [8]. After that the case feature representation is created which includes post bigram, pre bigram, pre trigram, in trigram and post trigram as illustrated in the table (TABLE II).

TABLE II FEATURE TYPE

	Feature Type	F1	F2	F3	No. of Features Taken
T1	Post-Bigram	L_{w1}	W	-	2
T2	Pre-Bigram	W	R_{w1}		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

B. Case Extraction

In case extraction process the word whose meaning needs to be clarified is chosen from corpus. This ambiguous word along with the associated words in the sentence are given to the CBR system. There are three steps for case extraction, which includes, instance identification, instance filtering and lastly, instance selection. In first step of case identification, the precise circulated E-dictionary which is having ambiguous word is selected. The cases with the ambiguous words are selected, and are referred as case identification. In second step of case filtering from the set of identified cases, the cases with ambiguous word's PoS are taken for disambiguation process. After this step the third step is started in which alike cases are selected using likeness determining functions for example Euclidean functions. Result of these are handed over as input for disambiguation process.

C. Disambiguation Process

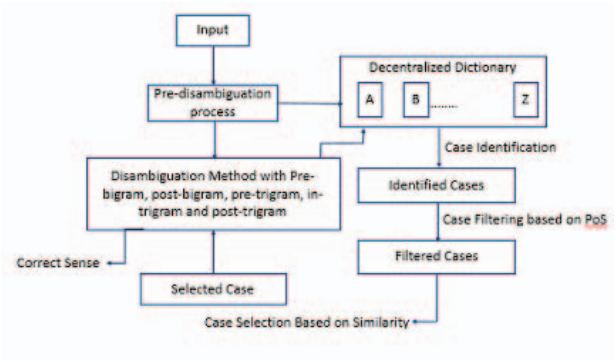

Fig. 3. CBR Process for WSD

Disambiguation has emerged as one of the critical tasks for NLP. For this purpose we have various algorithms like, naïve Bayes, Decision tree, random forests, k-NN and many more. One such more approach that is being followed to disambiguate an ambiguous word is case based interpretation model. This approach works on the principle that of the different variants available for a given instance, the closest gets selected and this task is carried out by using the minimal feature sets: T1 (Post Bigram), T2 (Pre Bigram), T3 (Pre Trigram), T4 (In Trigram), T5 (Post Trigram). The complete process of Disambiguation is shown in figure (Fig. 3).

Fig. 4 illustrates an example of the case data base with the following sentences:

1. ਰਾਮ ਪੌੜਿਆਂ ਉੱਤਰ ਰਿਹਾ ਹੈ।

2. ਪੌੜਿਆਂ ਉੱਤਰ ਦਿਸ਼ਾ ਵੱਲ ਹਨ।

3. ਮੈਂ ਪ੍ਰਸ਼ਨ ਦਾ ਉੱਤਰ ਸੋਚ ਕੇ ਲਿਖਿਆ।

4. ਮਹਿੰਦਰ ਡਿੱਗ ਗਿਆ ਤੇ ਉਸ ਦਾ ਘੁਟਣਾ ਉੱਤਰ ਗਿਆ।

5. ਸੂਰਜ ਉੱਤਰ ਵੱਲ ਜਾ ਰਹਿਆ ਹੈ।

The table (TABLE III) shows the meaning inferred in CBR Case no (refer Fig 4) used for word sense disambiguation.

For sentence number 5 there was no case in the data base but a similar case with some information was there and hence using the similarity it concluded the meaning. After verification of this new case it gets added to CBR repository and is further used for WSD.

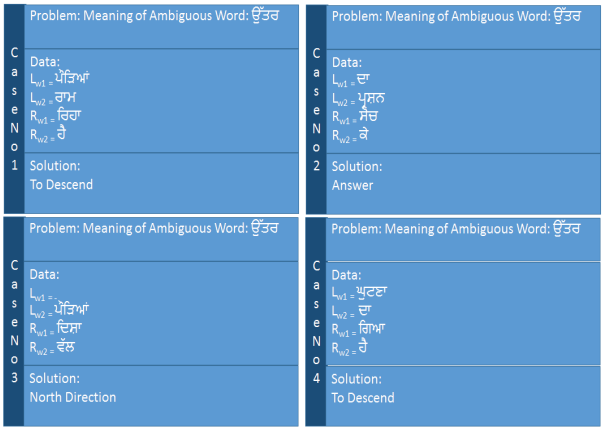

Fig. 4 Example

TABLE III MEANING OF DISAMBIGUOUS WORDS

Sentence Number	Case Number	Meaning
1	1	Descending
2	3	North
3	2	Answer
4	4	Dislocate
5	3	North

VI. RESULTS AND DISCUSSION

For this paper, we are using a sense-tagged corpora [9]. This corpora was manually sense-tagged with 100 words. In order to explain how the case based interpretation model works, we have taken into consideration three instances, namely, "ਉੱਤਰ (uttar)", "ਕੱਚਾ (kacha)", and "ਹਾਰ (haar)". The first instance, "ਉੱਤਰ", has four different interpretations (to descend, North direction, dislocation and answer), while the second instance, "ਕੱਚਾ", has three interpretations (unripe, nausea, crude) and the third instance, "ਹਾਰ", also has four different interpretations (necklace, defeat etc).

Further, these three words were tested using five different feature sets: T1 (Post Bigram), T2 (Pre Bigram), T3 (Pre Trigram), T4 (In Trigram), and T5 (Post Trigram) with respect to their different interpretations. This methodology was adopted in order to ensure that the interpretation model was able to rightly pick the correct interpretation of the given instance. The average of the number of times the system was able to give the correct answer is shown below in TABLE IV.

VII. CONCLUSION

In this paper, we have used cased based interpretation model to try to find the closest interpretation of the ambiguous word using the minimal feature set. The five different feature sets used are Post Bigram (T1), Pre Bigram (T2), Pre Trigram (T3), In Trigram (T4), and Post Trigram (T5). As seen in TABLE IV, it is evident that the result displayed by T2 (Pre-bigram) is the most promising.

The idea of using case based interpretation model for WSD in Gurmukhi was prompted by the fact that the humans are able to conclude the right context of the word mostly be their past remberance of the word in a similar situation. The same approach let to the work in this paper.

TABLE IV CALCULATED RESULTS

Ambiguous Word	No of Senses of the Word	No of Instances Tested	Feature Sets				
			T1	T2	T3	T4	T5
ਉੱਤਰ	4	153	56.67	65.4	52.67	64.32	54.2
ਕੱਚਾ	3	122	43.23	55.82	56.34	50.12	49.66
ਹਾਰ	4	164	63.42	69.34	68.52	67.34	66.90

ACKNOWLEDGEMENTS

The author[1] expresses gratitude towards the management of Delhi Technical Campus, Gr. Noida, who were gracious enough to give the consent to pursue doctorate and have been provided enough cooperation in the same regard.

The author[1] also expresses gratitude towards Amity University, Uttar Pradesh, where the author is registered as scholar and for providing with the library facility and extensive online resources for research.

REFERENCES

[1] R. Navigli, "Word Sense Disambiguation: A Survey", ACM Computing Surveys, Vol. 41, no.2, Article 10, February, 2009.
[2] J. Kaur, "Word Sense Disambiguation (WSD)", International Journal for Technological Research in Engineering, Volume 1, Issue 5, January 2014.
[3] R. Kumar, R. Khanna, and V. Goyal, "A Review of Literature on Word Sense Disambiguation", International Journal of Engineering Sciences, Vol. 6, July 2012.
[4] M. Bansal, "Word Sense Disambiguation: Literature Survey for Indian Languages", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 12, December 2015.
[5] H. Walia, A. Rana, and V. Kansal, "A study on different Word Sense Disambiguation Approaches and their application on Indian Regional Languages", Proceedings of International Conference on Technology and Trust (ICTT'17), 28-29 December, 2017.
[6] R. Kumar, and R. Khanna, "Natural Language Engineering: The Study of Word Sense Disambiguation in Punjabi", IJES, July 2011.
[7] H. Walia, A. Rana, and V. Kansal, "Different Techniques Implemented in Gurumukhi Word Sense Disambiguation", International Journal of Advanced Technology in Engineering and Science, Volume 05, Issue 06, June 2017.
[8] Punjabi WordNet. [Online]. Available: http://tdil-dc.in/indowordnet/index.jsp
[9] Punjabi Corpora. Obtained from Evaluations and Language Resources Distribution Agency, Paris, France.
[10] A. Narang, R. K. Sharma, and P. Kumar, "Development of Punjabi WordNet", Springer, CSIT, December 2013.

[11] J. Kaur, J. R. Saini, "Punjabi Stop Words: A Gurmukhi, Shahmukhi and Roman Scripted Chronical", Proceedings of the ACM Symposium on Women in research, March 2016, Page 32-37.
[12] J. Kaur, and V. Gupta, "Effective Approaches for extraction of Keywords", International Journal of Computer Science, Issue 6, Vol. 7, November 2010.
[13] S. Singh, and T. J. Siddiqui, "Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation", Proceedings of the International Conference on Information Retrieval & Knowledge Management, CAMP2012, 13-15 March, 2012, Malaysia, pp.1-5, IEEE Explorer.
[14] T. Pedersen, "A decision tree of bigrams is an accurate predictor of word senses", Presented at Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics, 2001
[15] P. Tamilselvi, and S.K.Srivastsa, "Case Based Word Sense Disambiguation Using Optimal Features", Presented at International Conference on Information Communication and Management, 2011.
[16] P. Tamilselvi, and S.K.Srivastsa, "Word Sense Disambiguation using case based Approach with minimal Features Set, Indian Journal of Computer Science and Engineering, 2011.