Journal of Computer Applications Research and Development (JCARD), ISSN 2248-9304(Print), ISSN 2248-9312 (Online), Volume 1, Number 1, April-June (2011)
pp. 49- 56 © PRJ Publication
http://www.prjpublication.com/JCARD.asp

QUANTIFYING THE FINEST SIMILARITY FOR CASE BASED REASONING TO IMPLEMENT WORD SENSE DISAMBIGUATION USING DIFFERENT LEARNING CLASSIFIERS

[1]Tamilselvi P, [2] S.K.Srivatsa
[1]*Research Scholar, Sathyabama Universtiy, Chennai, TN, India, tamil_n_selvi@yahoo.co.in*
[2]*St.Joseph Engineering College, Chennai, TN, India.*

ABSTRACT

In case base reasoning, solution for the current situation is derived using already existing similar cases. In this paper, case based approach is handled with bigram features get hold of pre-bigram and post-bigram, to identify the sense of a word in English language. Feature size of input and cases are taken as two, because of bigram features. Major task in the disambiguation process is text feature vectorization. Instead of considering the features as text, they are construed as vector form. To collect the similar cases of the ambiguous words, different distance measuring functions are used. To select the best case from the collected cases, that is, to make disambiguation, three different techniques, K-Nearest neighboring method, Baye's method artificial neural network are used. Among these, Baye's produced outstanding performance with 84.76% of disambiguation accuracy with pre-bigram.

Keywords: Word Sense Disambiguation, pre-bigram, post-bigram, similarity function, case based reasoning (CBR), Part-of-Speech (PoS), K-Nearest Neighboring method (KNN), Artificial Neural Network (ANN), Baye's method .

1. INTRODUCTION

One important problem of Natural Language Processing is figuring out what a word means when it is used in a particular context. The different meaning of a word is listed as its various senses in a dictionary. The task of word sense disambiguation is to identify the correct sense of a word in context. Improvement in the accuracy of identifying the correct word sense will result in better machine translation systems, information retrieval systems, etc. To assign an appropriate sense to an occurrence of a word in a given context many methods have been proposed to deal with the problem, including supervised learning algorithms (Leacock et al., 1998, Sin-Jae Kang et al, 2001), semi-supervised learning algorithms (Yarowsky, 1995, Zheng-Yu Niu et al, 2005), and unsupervised learning algorithms (Sch¨utze, 1998).

Text features are playing vital role in all kind of natural language processing (NLP) tasks such as word sense disambiguation (WSD), Machine translation (MT), information

retrieval (IR) etc. Commonly used text features are morphological text, PoS of the word, surrounding words, location collocations, verb-noun relation etc. (Hwee tou Ng, 2007) three different teams were involved to make the disambiguation process based on supervised learning concept. CITYU-HIF team used Naïve Baye's classifier with text features part of speech of the words, single words in the surrounding context classifier, HIT-IR-WSD team used support vector machine with a linear kernel function with text features POS of surrounding words, local collocations, single words in the surrounding context and syntactic relations and the third team PKU used support vector machine with maximum entropy classifier with POS of surrounding words, local collocations and single words in the surrounding context.

(Zheng-Yu Niu et al, 2007) used three types of features, part-of-speech of neighboring words with position information, unordered single words in topical context, and local collocations as same as the feature set used in (Lee and Ng, 2002) except the syntactic relations. If the occurrence frequency of a feature was less than three then, it was removed from feature set. (Marine et al, 2007) used word sense disambiguation task to improve the statistical machine translation. For doing WSD, they used the text features of position sensitive, syntactic and local collocation features for getting best disambiguation performance.

The aim of CBR is to reuse solutions of similar cases to solve the problem at hand (Weber et al, 2006). Clearly, the ability to compare text content is vital in order to identify the set of relevant cases for solution reuse. However a key challenge with text is variability in vocabulary which manifests as lexical ambiguities such as the polysemy and synonymy problems (Simpson, G.B, 1984). (Krisda Khankasikam, 2011) used case based reasoning to get solution (meta data) for the current situation. If the existing cases failed to produce the solution, case adaption was done to make changes in existing cases, in such a way to produce solution for the situation. To identify the similarity between the case and the input data, Euclidean distance was used. (Juan A. Recio-Garcia et al., 2010) used case based reasoning to infer knowledge from web pages. Cases were defined as a pair of problem-solution along with the vocabularies. Frequency of a term in a case was also calculated to group the content based on the context. It is stated that to measure the similarity between the cases and input, any one of the similarity function, namely, Euclidean, Cosine or KL-Divergence was used.

(Pedersen, 2001) experimented the use of bigrams for WSD with a decision tree and naive Bayes classifier. He tested different bigrams that occur close to the ambiguous words (within approximately 50 words to the left or right of the ambiguous word) as possible disambiguation features. He applied statistical method to disambiguate texts using decision tree with bigram concept. (Zhimao Lu et al., 2004) extracted mutual Information (MI) of the words as input vectors for back-propagation neural network. The network is tested with maximum feature sets varying from ten words from left and ten from right with respect to ambiguous word.

Common problems faced in natural language processing are data sparseness and inconsistency in vocabulary. When the number of features increases, the sparseness is unavoidable. Smoothing is really required to overcome the above problem for improving the performance. To avoid sparseness, bigram is adopted here. Text comparison is

required to disambiguate the input words. This paper presents a system to disambiguate words using case based reasoning with minimal features. With CBR, the solution for the problem is derived by comparing the current problem description with a set of past cases maintained in a case-bases, represented as distributed E-dictionaries (P.Tamilselvi et al, 2009, 2009). Rest of the paper is organized as follows: section 2 describes case based reasoning technique, section 3 describes system architecture, section 4 about the experimental results and section 5 with the conclusion.

2. ABOUT CASE BASED REASONING TECHNIQUE

Case-based Reasoning (CBR) is an approach in artificial intelligence that differs from other artificial intelligence approach (David, 1996, Krisda Khankasikam, 2011). Instead of depending on general knowledge of a domain or depending on knowledge gained by deduction from rule of a problem domain, CBR depends on knowledge that is previous experience of problem solving (S. Russell et al., 2002). In CBR, new problems are solved by remembering solution to problem which is similar to the current problems (I. D. Watson, 1997). As the problem cases and the remembered cases are often not perfectly matched cases the remembered solutions are modified in a way that at least parts of the case can be used.

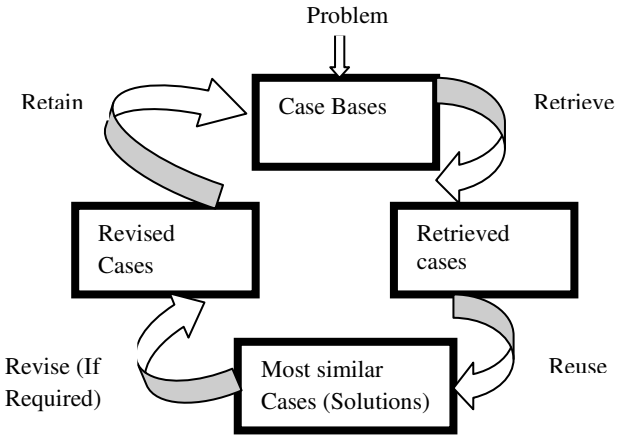

Fig-1: Schematic Cycle of CBR

The processes involved in CBR can be represented by a schematic cycle (Fig-1). Aamodt and Plaza [94] have described CBR typically as a cyclical process comprising *the four REs*:

- RETRIEVE the most similar case(s);
- REUSE the case(s) to attempt to solve the problem;
- REVISE the proposed solution if necessary, and
- RETAIN the new solution as a part of a new case.

A new problem is matched against cases in the case base and one or more similar cases are *retrieved*. A solution suggested by the matching cases is then *reused* and tested

for success. Unless the retrieved case is a close match the solution will probably have to be *revised* producing a new case that can be *retained*.

3. DISAMBIGUATION SYSTEM ARCHITECTURE

Disambiguation system uses minimal features as input; it uses case based approach together with the distributed E-Dictionaries for producing appropriate sense from the existing cases. Disambiguation system includes two steps, first, converting text input into vector forms, next, extracting most similar cases for solving the ambiguity by applying distance measuring functions. System Architecture is given in Fig-2.

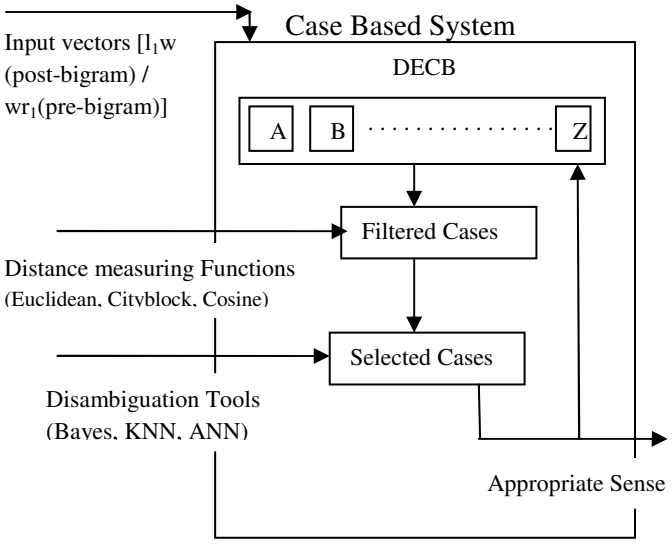

Fig-2: System Architecture

3.1 Vector Formation of text input

In general, input size in bigram is two, ambiguous word with either previous (post-bigram) or immediate next word (pre-bigram). When inputs are taken as text, process complexity would be high. To avoid it, weight of PoS of the inputs is taken in the form of vector of size 1×2.

word position	word	PoS	# of senses
3	work	VB	27
6	manager	NN	2
9	bank	NN	10

Fig-3: Ambiguous Words

Hence, weight assignment process is an important one. PoSs from Brown corpus are grouped (by eliminating two of them, POS and NPS) based on their nature, totally,

seventeen groups and weights are assigned for the groups with values .01 to .17. Input text features (PoS) are replaced by their relevant weight with the condition, 0.0001is assigned for absence of any feature (left or right word). For example, the sentence, 'He is working as a manager in a bank", here, all ambiguous words are isolated (Fig-3) and each of their relevant vectors are constructed as in Table-1.

Table1: Input Vectors

Features	Ambiguous words		
	work	manager	bank
Pre-Bigram	**.1500** .1200	**.1400** .1200	.0200 **.1400**
Post-Bigram	.1000 **.1500**	.0200 **.1400**	**.1400** .1200

3.2 Most similar case Extraction

In the case base, the first attribute represents the ambiguous word. Case filtration means collecting cases having their first column value as input ambiguous word. From the filtered cases, most similar cases are extracted using similarity measuring functions. For selecting the best case (correct sense of the ambiguous word) from the extracted cases, three different classifiers such as K-nearest neighborhood, Bayes method and artificial neural network are used.

4. EXPERIMENTAL RESULTS

Refined sentences of Brown Corpus (P.Tamilselvi, S.K Srivatsa, 2010) are considered for this research work. Totally, 1500 sentences are taken and 80% (1200) of sentences are treated as cases and remaining 20% (300) are taken for testing. Three different similarity-finding functions such as Euclidean, Cityblock and Cosine functions are used for case selection C_i, i=1,2,3. Three different classifiers are used to get the best case from the collected cases. KNN classifier, with k=1, is applied on three different set of cases collected by three different functions, to get the optimal (minimal) case as output for the current situation. Next, Bayes classifier is used for getting the optimal distance cases for the input. If tie exists between cases, best case will be selected on random basis. Same set of cases is taken as training data for the neural network. After completion of training, the input feature vector is simulated to the network to get the relevant output. Disambiguation accuracy of three different classifiers (KNN, Baye's & ANN) on three different similarity cases C_i, (i=1,2,3) with two different feature vectors (post-bigram (T1) and pre-bigram (T2)) is given in Table-2. From the table, it is clear that, Euclidean function with Baye's classifier with pre-bigram produced 84.76% of disambiguation accuracy.

Table-2: Disambiguation Accuracy in %

Similarity Function	Disambiguation Method	Post-Bigram	Pre-Bigram
Euclidean	K-NN	78.175	69.048
	Baye	76.19	**84.76**
	ANN	65.476	69.841
Cityblock	K-NN	78.175	72.619
	Baye	78.968	**81.746**
	ANN	32.143	69.048
Cosine	K-NN	81.746	**66.27**
	Baye	78.968	75.397
	ANN	68.254	63.492

Table-3: Performance comparison based on size of the sentences

Similarity Function	Disambiguation Method	Post-Bigram T1		Pre-Bigram T2	
		Seg1	Seg2	Seg1	Seg2
Euclidean	K-NN	83	78	83	69
	Baye	83	65	**100**	**85**
	ANN	58	76	67	70
Cityblock	K-NN	83	78	83	73
	Baye	83	79	**100**	**85**
	ANN	58	32	67	69
Cosine	K-NN	83	**82**	83	66
	Baye	83	79	**100**	75
	ANN	58	68	67	63

Sentences are considered into two segments, sentences having less than or equal to 10 words (seg1) and sentences having more than 10 words as (seg2). From Table-3, it is clear that the disambiguation accuracy produced by Baye's method for seg1 (length of the sentence $\leq$ 10 words) with T2 (with all three similarity functions) are almost 100%, (i.e) if the length of the sentence is small, accuracy level is to the maximum. For segment 2, Baye's classifier yields 85% accuracy with Euclidean and Cityblock functions, but with cosine, KNN produced 82% of accuracy with T1. Bayes classifier produced better results on the cases extracted by Euclidean function on both segments.

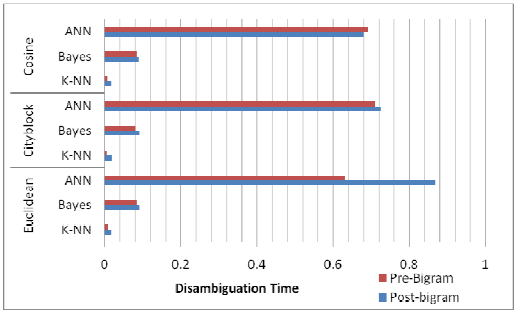

Chart-1: Disambiguation Time

Performance of disambiguation is also measured based on disambiguation time. For this, we used the basic command tic and toc (stop watch counter) to measure the processing time with an assumption that no other task should be assigned to CPU. Since the time given by tic-toc is not constant always, we made each disambiguation process for five times (no changes in output, only the time values got changed) and mean of them is treated as processing time, given in Chart-1. In the chart, it is clearly shown that KNN took less time but ANN took high processing time and Baye's method took a little more time than KNN. Even though Baye's method took more time than KNN, the disambiguation accuracy is 100% in seg-1 and 85% in seg-2, with minimal features.

5. CONCLUSION

Three different similarity functions Euclidean, Cityblock and Cosine are used for case selection from distributed E-case base. Cases are extracted by only two text features pre-bigram (ambiguous word + immediate next word) or post-bigram (preceding word + ambiguous word). Cases are processed with KNN and ANN for the ambiguous words in the list (prepared from input sentence). Among these, cases selected by Euclidean distance measuring function with Pre-bigram vector using Baye's Classifer produced 84.76% of disambiguation accuracy. This is suggested as better among all tested methods because, with all types of sentence (seg1 and seg2) the same combination produced 85% of disambiguation. Level of accuracy performance may be increased by raising the feature elements size as three or four in the row vector.

REFERENCES

[1]. Juan A. Recio-Garcia 1 and Nirmalie Wiratunga, 2010, Taxonomic Semantic Indexing for Textual Case-Based Reasoning, ICCBR'2010. pp.302-316.

[2]. Krisda Khankasikam, 2011, Metadata Extraction Using Case-based Reasoning for Heterogeneous Thai Documents, International Journal of Computer and Electrical Engineering, Vol.3, No.1, February, 2011

[3]. Marine CARPUAT Dekai WU, 2008, Evaluating the Word Sense Disambiguation Performance of Statistical Machine Translation, LREC 2008

[4]. S. Russell, and P. Norvig, 2002, Artificial ntelligence: A Modern Approach. 2nd ed, New York: Prentice-Hall.

[5]. P.Tamilselvi, S.K.Srivatsa, 2009, Decentralized E-Dictionary (DED) for NLP task, Proceedings of ICMCS International conference on Mathematics and computer Science, India, 2009

[6]. I. D. Watson, 1997, Applying Case-Based Reasoning: Techniques for Enterprise Systems. California: Morgan Kaufmann.

[7]. P.Tamilselvi, S.K.Srivatsa, 2010 , A Study on Lexicographical Information using open source lexical databases, Proceeding of NCRTCSE National conference on Recent Trends in Computer Science and Engineering, 2010

[8]. Simpson, G.B, 1984, Lexical ambiguity and its role in models of word recognition, Psychological Bulletin 92(2), 316–340

[9]. T. Pedersen, 2001, A decision tree of bigrams is an accurate predictor of word senses, in: Presented at Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics.

[10]. Weber, R.O., Ashley, K.D., Br¨uninghaus, S, 2006, Textual case-based reasoning. The Knowledge Engineering Review 20(03), 255–260.

[11]. Zhimao Lu, Ting Liu, and Sheng Li. 2004, Combining neural networks and statistics for chinese word sense disambiguation, *ACL* SIGHAN Workshop, pages 49-56.

[12]. Zheng-Yu Niu, Dong-Hong Ji, Chew Lim Tan, 2007, Learning model order from labeled and unlabeled data for partially supervised classification, with application to word sense disambiguation. Computer Speech & Language 21(4): 609-619.

[13]. Sin-Jae Kang, Jong-Hyeok Lee, 2001, "Ontology Word Sense Disambiguation by Using Semi-Automatically Constructed Ontology", Machine Translation Summit VIII, pp. 181-186.

[14]. David B. Leake, 1996, CBR in Context: The Present and Future, 1996, Menlo Park, AAAI Press/MIT Press.

[15]. Leacock, C., Chodorow, M., and Miller, G. A., 1998, Using corpus statistics and WordNet relations for sense identification. Computational Linguistics, 24(1).

[16]. David Yarowsky, 1995, "Unsupervised Word Sense Disambiguation Rivaling supervised methods" in proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL 1995).

[17]. Hinrich Schütze, 1998, "Automatic Word Sense Discrimination.", Computational Linguistics, 24(1).

[18]. Ng, Hwee Tou, & Chan, Yee Seng, 2007, English Lexical Sample Task via English-Chinese Parallel Text. Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007). (pp. 54 – 58).

[19]. Y. K. Lee and H. T. Ng., 2002, An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In Proc. of EMNLP.

[20]. Aamodt, A. and Plaza, E., 1994, Case-based Reasoning: Foundation issues, methodological variations and system approaches. AI- Communications, 7(1): pp 39-59.