2011 International Conference on Information Communication and Management
IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore

Case Based Word Sense Disambiguation Using Optimal Features

P.Tamilselvi [1], S.K.Srivatsa [2]
[1] Sathyabama University, Chennai, Tamilnadu, India
[2] St.Joseph College of Engineering, Chennai, Tamilnadu, India

Abstract. Optimal features refer about achieving best accuracy with minimum features. In this paper, optimal features are used for word sense disambiguation using case based approach. Disambiguation is tried with two (bigram) and three (trigram) and four (n-gram) features. Feature size of input (ambiguous word) and cases are taken as two for bigram three for trigram and four for n-gram. Major task in the disambiguation process is 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 to solve the ambiguity, three different classifiers namely K-Nearest neighboring method, Baye's classifier and artificial neural network are used. Among these, Baye's classifier produced outstanding performance with 84.76% of disambiguation accuracy with pre-bigram.

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

1. Introduction

With the rapid development of artificial intelligence (AI) technology, rule-based reasoning (RBR) and case-based reasoning (CBR) are applied more on linguistic soft computing systems (LSCS) for information retrieval, Machine learning process etc. Generally, the major task, sense extraction for NLP tasks is very complicated and time consuming when a clumsy disambiguation algorithm applied with centralized dictionary entries. Experienced disambiguation algorithms are difficult to formalize. Moreover, text ambiguity itself is imprecise and uncertain sometimes. For these reasons, it is hard to reasonably represent and acquire the sense from a huge case library using obsolete high level system development. As for CBR, it is hard to perform disambiguation when size of case library is huge and the processing document is lengthy. So solving massive case selection problems mainly depends on reducing the case attributes and the document information is split into individual sentences and processed one sentence at a time. As a result, newly framed intelligent disambiguation system produced better performance with minimal text features. In general, it is hard to apply expert's experiences, knowledge, and the methods accumulated in practices to the development of intelligent disambiguation system effectively. Here, CBR is applied with minimal attributes for sense selection with different learning classifiers. 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). Rest of the paper is organized as follows: section 2 describe the related research works, section 3 describes system architecture, section 4 about the experimental results and section 5 with the conclusion.

[1] P.Tamilselvi, Mob No: (90) 9840494754; E-Mail address: nagas_67@hotmail.com
[2] S.K.Srivatsa, Mob No: (90) 9444181459; E-Mail address: profsks@rediffmail.com

2. Related Works

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, 2008) 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.

(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, minimal features bigram, trigram and n-gram with four features are adopted and compared here. 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 case-bases, represented as distributed E-dictionaries (P.Tamilselvi et al, 2009).

3. System Architecture

Disambiguation system architecture is described with feature vector representation and sense disambiguation.

Column	C1	C2	C3	C4	C5	C6
Fields	***case***	***Sense_value***	***Sense_tag***	$l_3l_2l_1$	***W***	$r_1r_2r_3$
Description	ambiguous word	WordNet Sense value	WordNet Sense tag	Weight of immediate three left words	Weight of ambiguous word	Weight of immediate right three words

Fig.1: Structure of distributed Case-Base

3.1. Feature vector representation

In general, input of bigram is ambiguous word with either previous (post-bigram – T1) or immediate next word (pre-bigram – T2), in trigram ambiguous word with two immediate right words (pre-trigram – T3), or with one word from left and right (in-trigram – T4) or with two immediate left words (post-trigram – T5) and in n-gram (T6), two left words one right word along with ambiguous word at third position. 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 (bigram) 1×3 (trigram) and 1×4 (n-gram). Hence, weight assignment process is an important task. PoS notations from Brown corpus are pursued to categorize (by eliminating POS and NPS) them into 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, ambiguous words {work, manager, bank} are isolated and each of their relevant vectors of bigram, trigram and n-gram are framed. Cases are in electronically distributed form. Totally 26 distributed casebases used to maintain cases based on their starting character. For example, the case 'bank' is maintained under the casebase 'B'. A case is represented with six fields (Fig-1). In the casebase, C1 is the case word, C2 & C3 together are solution for the case, C4, C5 & C6 are used for case feature vector (F1, F2, F3 and F4) representation (Table-1).

Table.1: Case feature vector representation

Feature type	F1	F2	F3	F4	Number of features taken
Post-Bigram	l_1	W	-	-	2
Pre-Bigram	W	r_1	-	-	2
Pre-Trigram	W	r_1	r_2	-	3
In-Trigram	l_1	W	r_1	-	3
Post-Trigram	l_2	l_1	W	-	3
n-gram	l_2	l_1	W	r_1	4

3.2. Sense disambiguation

Consider a document D is given as input for the system. Pre-disambiguation process is applied for the document to split it into sentences S_i (i=1,2,..N). Next, split S_i into individual morphological words W_j (j=1,2..,M), N is total number of sentences and M is total number of words in sentence S_i. Collect all ambiguous words of the W_i for disambiguation. Frame vectors of bigram, trigram and n-gram of ambiguous words (input vectors). Select the ambiguous word one at a time and find the appropriate casebase of it and filter the cases having C1 value as input ambiguous word and frame case vectors from their respective features. Three different distance metric functions such as Euclidean, cityblock and cosine are used to select the minimal distance of case with input word and finally three different classifiers, namely, K-nearest neighboring classifier (KNN), Baye's classifier and artificial neural network (ANN) are used to extract the best case from the minimal distance cases and produce the relevant case's solution part as sense for the input word. Sense disambiguation algorithm is given in Fig-2.

Table 2 Disambiguation Accuracy in %

Distance Metric Functions	Learning Classifier	T1	T2	T3	T4	T4	T6	Maximum
Euclidean	K-NN	78.18	69.05	77.39	71.83	71.04	**80.16**	80.16
	Bayes	76.2	**84.77**	72.62	76.2	78.97	80.96	***84.77***
	ANN	65.48	**69.85**	62.7	66.27	58.34	63.5	69.85
Cityblock	K-NN	78.18	72.62	77.39	71.83	71.04	**80.16**	80.16
	Bayes	78.97	81.75	72.62	76.2	**81.75**	77.39	81.75
	ANN	32.15	**69.05**	62.7	57.15	59.13	59.13	69.05
Cosine	K-NN	**81.75**	66.27	77.39	71.83	63.89	76.59	81.75
	Bayes	**78.97**	75.4	69.85	76.2	78.18	77.39	78.97
	ANN	**68.26**	63.5	66.27	63.5	59.13	62.7	68.26

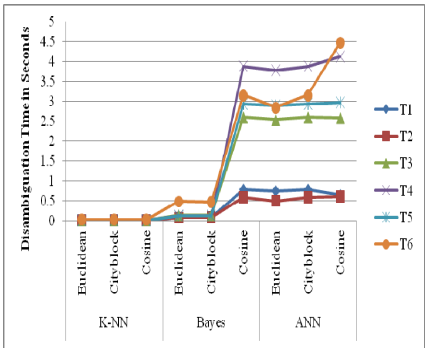

Chart-1: Disambiguation Time

1. *Get the document D with N sentences.*

$D = \sum_{i=1}^{N} S_i$

2. *For i=1:N*

$M_{S_i} = predis(S_i)$ % M_{S_i} *is the collection of words in* S_i

$A_{S_i} = amb(M_{S_i})$ % A_{S_i} *is the collection of ambiguous words in* M_{S_i}

$[row\ col] = size(A_{S_i})$ % $row = 1, col =$ *the number of ambiguous words in* S_i

for $j = 1:col$

$IP_{S_{ij}} = inp\_vec(A_{S_{ij}})$% *construct the input feature vector*

$FC_{S_{ij}} = case\_filter(IP_{S_{ij}})$% *get the cases from the casebase*

for k=1:3

$DM_{S_{ij_k}} = dist_{metric_k}(FC_{S_{ij}})$

for l=1:3

$sense_{S_{ij_k}} = classify_i(DM_{S_{ij_k}})$

end % for index l

end % for index k

end % for index j

end % for index i

Fig.2: Sense Disambiguation Algorithm

4. Experimental Results

Testing document is taken form Brown Corpus. Even though the sentences in Brown Corpus are sense tagged, we didn't consider the same PoS, because, individual words are framed as compound word and for that PoS is assigned. In our algorithm, all words should be individual, so parsing is done separately using Hidden Markov Model (P.Tamilselvi et al., 2010). Main purpose of this disambiguation system is to produce better result with minimal features. The performance of the system is measured using multiple scales. Disambiguation accuracy is measured and compared using three different distance metric functions together with three different learning classifiers (Table-2). From the table it is clear that, the minimal feature pre-bigram (T2) produced 84.77% disambiguation accuracy by Baye's classifier with Euclidean function. Sentence is categorized into two types based on its length, length with less than or equal to ten words are grouped into segment 1 (seg-1) and more than ten words are as segment 2 (seg-2). Disambiguation accuracy is measured based on the length of a sentence (Table-3). Table-3 values projects that pre-bigram (T2) vector

yielding 100% accuracy for seg-1 and 84.76% for seg-2 with Euclidean as well as City-block using Baye's classifier. Disambiguation time is also calculated using 'tic-toc' command with an assumption no other task should be assigned to CPU. Still, the disambiguation time is not fixed always, so we did the disambiguation process five times for each word and average of that is taken as disambiguation time for that word. Chart-1 shows the disambiguation time. KNN with all three functions for all 6 features took less time, Bayes classifier with Euclidean and cityblock functions took less time for all six types of features but cosine distance function took a little more time for the feature types T3, T4, T5 and T6. With ANN, the three functions took more time for T3, T4, T5 and T6 and overall accuracy performance is less.

5. Conclusion

In this paper, disambiguation system is implemented with three different set of features with three different distance measuring functions combined with three different classifiers for word sense disambiguation. The system is used on a document, means one paragraph. At the end of execution of the system, sentencewise ambiguous words and their sense are listed. Using Neural Networks with enormous number of features, accuracy measured from 33.93% to 97.40% for words with more than two senses and 75% of accuracy for words with two senses (A. Azzini, C. da Costa Pereira, M. Dragoni and A. G. B. Tettamanzi, 2008). Here,case based disambiguation accuracy of 84.77% made us to recommend pre-bigram feature type with Euclidean distance function together with Baye's classifier as better approach for disambiguation. The system can be extended for doing disambiguation process on a document with more than paragraph.

Table 3 Disambiguation Accuracy based on sentence length

Distance Metric Functions	Learning Classifier	Seg-1						Seg-2					
		T1	T2	T3	T4	T5	T6	T1	T2	T3	T4	T4	T6
Euclidean	K-NN	83.333	83.333	83.333	66.667	83.333	83.333	78.175	69.048	77.381	71.825	71.032	80.159
	Bayes	83.333	**100**	83.333	83.333	83.333	83.333	76.19	**84.76**	72.619	76.19	78.968	80.952
	ANN	58.333	66.667	54.167	58.333	16.667	41.667	65.476	69.841	62.698	66.27	58.333	63.492
Cityblock	K-NN	83.333	83.333	83.333	66.667	83.333	83.333	78.175	72.619	77.381	71.825	71.032	80.159
	Bayes	83.333	**100**	83.333	83.333	83.333	83.333	78.968	**84.76**	72.619	76.19	81.746	77.381
	ANN	58.333	66.667	54.167	66.667	41.667	54.167	32.143	69.048	62.698	57.143	59.127	59.127
Cosine	K-NN	83.333	83.333	83.333	66.667	83.333	83.333	81.746	66.27	77.381	71.825	63.889	76.587
	Bayes	83.333	**100**	83.333	83.333	83.333	83.333	78.968	75.397	69.841	76.19	78.175	77.381
	ANN	58.333	66.667	70.833	66.667	41.667	29.167	68.254	63.492	66.27	63.492	59.127	62.698

6. References:

[1] 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.

[2] 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.

[3] Zhimao Lu, Ting Liu, and Sheng Li. 2004, *Combining neural networks and statistics for Chinese word sense disambiguation*. In Oliver Streiter and Qin Lu, editors, *ACL SIGHAN Workshop 2004*, pages 49-56.

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

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

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

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

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

[9] 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.

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

[11] 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, 2001.

[12] 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

[13] P.Tamilselvi, S.K.Srivatsa, *Part-Of-Speech Tag Assignment Using Hidden Markov Model*, International Journal of Highly reliable Electronic System, Vol 3, No 2, 2010.

[14] A. Azzini, C. da Costa Pereira, M. Dragoni, and A. Tettamanzi. *Evolving Neural Networks for Word Sense Disambiguation*, HIS'08, pages 332-337, LNCS, Springer, Septermber 2008.