DOI 10.29042/2018-3916-3918

Helix Vol. 8(5): 3916- 3918

N-gram Based WSD for Improving Accuracy of Machine Translation using TM

[*1]Sunita Rawat, [2]Manoj Chandak, [3]Tabassum Khan

[1]Research Scholar, G. H. Raisoni College of Engineering, Nagpur

[2] Supervisor, G. H. Raisoni College of Engineering, Nagpur

[3]Assistant Professor, G. H. Raisoni College of Engineering, Nagpur

Email: ssunitarawatt@gmail.com, chandakmb@gmail.com, tabassum.khan@raisoni.net

Received: 09th July 2018, Accepted: 14th August 2018, Published: 31st August 2018

Abstract

Word Sense Disambiguation (WSD) concerns with selecting an accurate sense of a term automatically in the given situation. It is very significant and challenging problem in several applications of natural language processing. We have used a probabilistic model in our system. Word Sense Disambiguation is done based on n-grams (bigrams and trigrams). The objective is to evaluate the performance of the system with various kinds of an input string which have an ambiguous word. For resolving the probability of the sense of an ambiguous word in the given sentence we use a Naïve Bayes classifier. We have use translation memory (TM) to speed up the translation process. A classical translation memory chooses contender translations into a target language (Hindi) by getting similar translations to an entered text from available pairs of earlier translated segments.

Keywords: Word Sense Disambiguation, Machine Translation, Naïve Bayes Classifier, Translation Memory.

Introduction

In natural language processing Word Sense Disambiguation is a topic that has been learned from so many years. The objective of word sense disambiguation is to choose the accurate sense of an ambiguous word in a specified context. In reality automatic word sense disambiguation persists to be an open problem has raised an immense attention in the community of computational linguistics, hence, various techniques has been commenced in the previous decades [1]. Rivalries like Senseval and lately SemEval1 have moreover encouraged the invention of innovative methods for word sense disambiguation, offering a motivating atmosphere for checking those methods. In spite of the word sense disambiguation task has been learned for a decade of time, the probable sentiment is that word sense disambiguation has to be included into actual applications for example information retrieval, lexicography, machine translation systems, knowledge mining, automatic answer machines, semantic interpretation, etc. [1]. So many studies on this matter have confirmed that those applications have advantage from word sense disambiguation [2, 3].

To decide the probability of a sense of an ambiguous word, the statistical dictionary is given as input to Naïve Bayes classifier. We make a classification model which depends on the probability of finding correct sense of each ambiguous word as well as the words which enclose it. We know that some other classification models are also present for instance, support vector machine [4] and conditional random field [5]. On the other hand, while we have preferred a probabilistic model founded on independent description so as to determine the accurate target sense, we consider that the Naïve Bayes classifier entirely suits with this type of technique.

The objective of this research is to appraise to what level every word, in a surrounding of the ambiguous word, adds to getting better the process of the word sense disambiguation. To speed up the translation process, we have make use of translation memory (TM). Where, translation memory is a database which stores pair of original (source segment) and translated (target segment) text.

Related Work

The choice of the suitable sense for a specified ambiguous word is usually done by taking into account the words nearby the ambiguous word. An inclusive analysis of a number of techniques possibly covered in [1]. While possibly seen, lots of work has been made on verdict the most excellent supervised learning technique for word sense disambiguation (such as, see [6, 7, 8, 9]) however regardless of the large variety of learning algorithms, it has been found that several classifiers for instance Naïve Bayes are extremely competitive and their performance essentially based on the depiction schema and their feature preference method.

In literature some other works are explained where for solving the problem of word sense disambiguation parallel corpora was used [10, 3]. These kinds of methods are normally used to

 Copyright © 2018 Helix ISSN 2319 – 5592 (Online)

discover the most excellent sense in the given language.

Word Sense Disambiguation

WSD is an essential work in natural language processing because of the reality that the several meanings are corresponds to an ambiguous word in the given source language. For example the word "date" which possibly will have numerous meanings. Let we choose one among these available meanings, that is, "Saumya likes dates", here date is one kind of fruit. "Ashish went on date", in this context date is related to romantic meeting. "Rashmi has exam on date 27th", this date is related to calendar date. Consequently, the capability for disambiguating an ambiguous word in given source language is important for the job of machine translation [11].

The Probabilistic Model

Consider an input sentence in English, whose representations are taken by using *n*-grams. Suppose S = {a$w1$, a$w2$, …, awk, …, a$wk+1$, …, a$w|S|$} be the *n*-gram representation of English sentence created by taking collectively all the *n*-grams, where awk is the representation of an ambiguous word. Our objective is to find an appropriate candidate for the polysemous word awk. Hence, we have planned to use a Naïve Bayes classifier which considers the probability of specified awk. A proper explanation of the classifier is :

$$p(t_i^k|S) = p(t_i^k|w_1, w_2, \ldots, w_k, \ldots)$$

To find N candidate meanings of the polysemous word awk corpus is used. Corpus gives all the probable meanings for awk with the corresponding details. Thus, we can use word that match with the equivalent class of the ambiguous word.

N-gram Model

N-grams are just entirely groupings of neighboring letters or words of size n that are present in the source text. Consider an example, specified the word apple, for this the "bigrams" or 2-grams are ap, pp, pl and le. We can moreover count the word boundary – that would enlarge the list of bigrams to #a, ap, pp, pl, le and e#, where # represents a word boundary.

Translation Memory

A translation memory is basically a record of earlier translated pairs of corresponding source language and target language segments, usually, sentences. When source language sentence is given to the translation memory for translation, will find in the database and selects samples which intimately match with the given sentence. Concept is translator may use earlier translated text as model. Consider a case, where a translator has to translate an input text so, TM first checks whether it has translated any such sentence or part of it before and if so it shows that as a suggestion. It is then up to the translator to use the existing translations or translate the sentence from scratch [12].

System Architecture

In our system first of all ambiguous words in the given string get catch. Ambiguity of all found words gets clear by applying N-gram model. For speed up the translation in our system we have use translation memory.

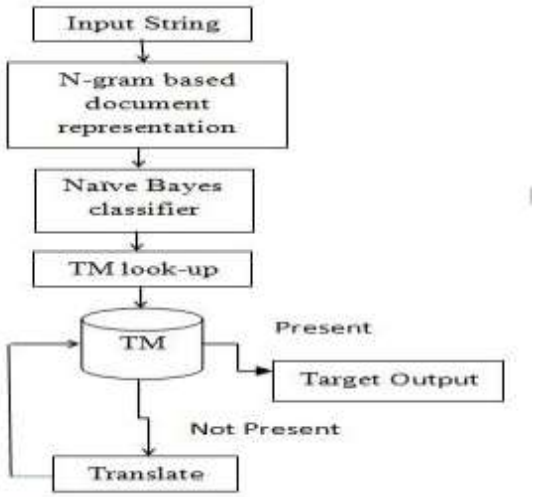

Figure 1. System Architecture

In our system we have use bigram and trigram model.

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Above bigram formula gives the probability of coming two words together. Consider following examples,

- <s> I am tourist </s>
- <s> tourist I am </s>
- <s> I do not like cold drink </s>

P(I|<s>)=2/3=.67
P(tourist|<s>)=1/3=.33
P(am|I)=2/3=.67
P(</s>|tourist)=1/2=0.5
P(tourist|am)=1/2=.5P(do|I)=1/3=.33

In the above example, we have considered three sentences. So, probability of coming "I" at the beginning of sentence is .67, as among three two sentences are starting with "I". Probability of starting sentence with "tourist" is .33. Probability of coming "I" and "am" together is .67, whereas probability of coming "am" and "tourist" together is .5. We can calculate probability matrix also for same.

 Copyright © 2018 Helix ISSN 2319 – 5592 (Online)

	i	want	to	see	Taj	Mahal
i	7	947	0	23	0	0
want	2	0	783	1	5	836
to	3	0	2	761	0	0
see	0	0	3	0	58	349
Taj	11	0	64	0	0	65
Mahal	10	0	86	0	0	0

Fig 2: Probability Matrix

Now consider Trigram model, here the probability of a word, conditioned on two previous words

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

P(tourism brings wealth and fame to the country)
= q(tourism|*,*)
x q(brings|*,tourism)
x q(wealth|tourism, brings)
x q(and|brings, wealth)

Therefore, by using bigram and trigram model we find the probability of the correct sense to be considered for ambiguous word.

While using translation memory input string is considered as „input"; and the samples preferred are known as „matches", in spite of whether they are useful or not to the translator. In some instance, an accurate translation is available so may place into the target text. Or else, fractional matches will be available which can be utilized to direct the translator.

This kind of fractional matching is generally known as „fuzzy matching". It allows predictable matches to be ordered with respect to the „fuzziness" or the degree of similarity in relation to the input sentence.

Conclusion

WSD is the task of deciding accurate meaning of the word (among available all meanings) in the specified context. In this paper, we presented a word sense disambiguation technique which uses *n*-grams for input sentences containing ambiguous word. Specifically, we used a Naive Bayes classifier for finding the probability of the accurate sense of an ambiguous word in the source language. Naïve Bayes is the simplest classification algorithm. It is easy to implement and requires only small amount of training data. It exhibits high accuracy and speed when applied to large databases. We have also used the concept of translation memory to speed up the translation process. Advantage of using TM is reduced translation cost as well as time.

References

1. Aguirre, E. & Edmonds, P. 2006. Word Sense Disambiguation: algorithms and applications. Dordrecht: Springer.

2. Carpuat, M. & Wu, D. 2007. Improving statistical machine translation using word sense disambiguation. 2007. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). Prague, Czech Republic, 61-72.

3. Chan, Y., Ng, H. & Chiang, D. 2007. Word sense disambiguation improves statistical machine translation. 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 33-40.

4. Cortes, C. & Vapnik, V. 1995. Support-vector networks. Machine Learning, 20 (3), 273–297.

5. Lafferty, J.D., McCallum, A. & Pereira, F.C.N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Eighteenth International Conference on Machine Learning, ICML "01. Massachusetts, USA, 282–289

6. Florian, R. & Yarowsky, D. 2002. Modeling consensus: Classifier combination for word sense disambiguation. ACL-02 Conference on Empirical Methods in Natural Language Processing, Philadelphia, USA, 10, 25–32.

7. Lee, Y.K. & Ng, H.T. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. ACL-02 Conference on Empirical Methods in Natural Language Processing, Philadelphia, USA

8. Mihalcea, R.F. & Moldovan, D.I. 2001. Pattern learning and active feature selection for word sense disambiguation. Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2). Toulouse, France, 127–130.

9. Yarowsky, D., Cucerzan, S., Florian, R., Schafer, C. & Wicentowski, R. 2001. The Johns Hopkins Senseval2 system descriptions. Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2). Toulouse, France, 163–166.Ng, H. T., Wang, B. & Chan, Y. S. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. 41st Annual Meeting of the Association for Computational Linguistics (ACL"03). Sapporo, Japan, 455–462.

10. Sinha, R., McCarthy, D. & Mihalcea, R. 2010. Semeval-2010 task 2: Cross-lingual lexical substitution. NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions. Colorado, USA, 76–81.

11. Edward K. Whyman and Harold L. Somers. 1999. Evaluation Metrics for a Translation Memory System. Published in Journal: Software-Practice & Experience. Volume 29, Issue 14, 1999, pg 1265 – 1284.

 Copyright © 2018 Helix ISSN 2319 – 5592 (Online)