Primitive-Based Word Sense Disambiguation For SENSEVAL-2

Lim Beng Tat, Prof Zaharin Yusoff, Dr Tang Enya Kong and **Dr Guo Cheng Ming**
Unit Terjemahan Melalui Komputer,
Universiti Sains Malaysia,
11800, Pulau Pinang.
{btlim, zarin, enyakong, cmguo}@cs.usm.my

Abstract

This paper describes a descriptive-semantic-primitive-based method for word sense disambiguation (WSD) with a machine-tractable dictionary and conceptual distance data among primitives. This approach is using unsupervised learning algorithm and focuses only on the immediately surrounding words and basis morphological form to disambiguate a word sense. This approach also agrees with past observations that human only requires a small window of a few words to perform WSD. (Choueka & Lusignan, 1985). In additional, this paper also describes our experience in doing the English all-word task in SENSEVAL-2. Then, we will discuss the results in the SENSEVAL-2 evaluation.

Apart from the description of current system, possibilities for future work are explored

1 Primitive-Based Word Sense Disambiguation

This system consists of three important components: machine-tractable dictionary, conceptual distance data and sense tagger that uses a simple summation algorithm.

1.1 Machine-Tractable Dictionary

The first one is Machine-Tractable Dictionary (MTD) such as WordNet and LDOCE (Longman Dictionary of Contemporary English) especially LDOCE has been used extensively in NLP research and provide a broad set of senses for sense tagging. MTD contains word senses and their definitions are defined in term of descriptive and tagged primitives (words attached with sense number). Primitives are a set of words derived from dictionary (Guo, 1989b) and it is used to define the definition of a word sense. (For further information about primitives, please refer to Wilks.Y (1977)). For example, father#1 has a definition defined by using four primitives that are '*title1*','*respect2*', '*priest3*' and '*church4*' (refer figure 1).

For SENSEVAL-2 competition, the pre-release WordNet1.7 was used for this purpose. After WordNet1.7 was downloaded, the entries including their definition, sense number and sense id in WordNet was extracted and written into a temporary file. Primitives (not tagged) were derived from the words used in the word senses' definition. Then, the first 7 words of the definition text of the WordNet dictionary were disambiguated using the information from an existing MTD (LDOCE) and the derived primitives (Guo, 1998a). The existing MTD (LDOCE) contained word senses and the words in their definition are already tagged.

Thus, a new MTD (the pre-release WordNet 1.7) was ready for the usage of tagging process.

1.2 Conceptual Distance Data

Conceptual distance data is showing the relatedness between two tagged primitives.

Basically, the conceptual distance data is calculated by using content terms in the definition to determine the relatedness measure between two primitives layer by layer. The definitions of the primitives are getting from the existing MTD (LDOCE). It is important to note that a tagged primitive is also a word sense. For example, the first and second layer of referential definition for word sense '*forecast2*' is:

forecast2 [def] predict1 in2 advance3
predict1 [def] make1 a2 prediction3 about4; tell1 in2 advance3
1-referential layer: forecast2 predict1 advance3
2-referential layer: predict1 make1 prediction3 tell1
(note:'advance3' is omitted because it has been counted in the first layer)

Formula used to compute the relatedness percentage:

% for first layer of the first target word sense and first layer of the second target word sense:

if q<(n1+n2)/2 then p1 = (q/((n1+n2)/2)*70%

if q>(n1+n2)/2 then p1 = 70%

% for other layers:

x1 = q1 / ((n3+n4)/2)

x2 = q2 / ((n1+n4)/2)

x3 = q3 / ((n2+n3)/2)

p2=((x1+x2+x3)/3)*30%

The total value = p1+p2

n1 = no. of the element in the first layer of first target word sense

n2 = no. of the element in the first layer of second target word sense

n3 = no. of the element in the second layer of first target word sense

n4 = no. of the element in the second layer of second target word sense

q, q1, q2, q3 = no. of common content terms for each comparison

p1, p2 = final value of the relatedness measure

1.3 Sense Tagger

The third one is sense tagger. Sense tagger will get the input from MTD and its conceptual distance data among primitives to do the word sense disambiguation. Currently, the tagger consists of three processes:

- Preprocess process.
- Dictionary look-up process
- Numerical calculation algorithm

In the preprocess process, test data, which is downloaded for the usage of SENSEVAL-2, is going through several processes before tagging process takes place. The first process is separating the given text into sentences using full stops as separator. After that, the words in the sentences that do not require tagging will be removed, leaving only the heads (words to be sense-tagged) behind. Then, each word in the sentences will be stemmed, leaving only morphological root. The list of heads is then cut into chunks of three successive heads to be tagged in seconds.

In dictionary look-up module, word senses with their definition for each of the words in a chunk is extracted from MTD.

After that, sense tagger will use numerical calculation algorithm to choose the suitable word sense for the words in the sentence. This algorithm is to compute the path value among the definition of the word senses in a sentence. This is done first, by summing up the semantic data from conceptual distance data when comparison among primitives in the definitions for the word sense pairs in the sentence. After that, the result of summation has to be multiplied with the distance value between the two words in the sentence. This distance value basically depends on total words in a sentence. For example, sentence "Father marry couple", the distance value for '*father*' and '*marry*' is 2 whereas the distance value for '*father*' and '*couple*' is 1. This is because word '*father*' is closer to the word '*marry*' than '*couple*'. Then this computation continues for the other of word sense pairs.

Finally, this algorithm will compare the path values for the combination of word senses in a sentence and find the highest path value. Then this algorithm assigns the best combination of senses to each word in the sentence.

For example, with reference to Figure 1, assume that words such as '*father*', '*marry*' and '*couple*' have two senses only.

In the first step, definition of sense 1 from '*father*' will compare with definition of sense 1 from '*marry*'.

father 1	marry 1	value
title1	take1	x1
	person2	x2
	marriage3	x3
respect2	take1	x4
	person2	x5
	marriage3	x6
priest3	take1	x7
	person2	x8
	marriage3	x9
church4	take1	x10
	person2	x11
	marriage3	x12
Total		x1+x2+...x12=X
Total comparison		12

(Note: x1, x2, ...x12 are the values accessed from conceptual data.)

In the second step, definition of sense 1 from '*father*' will compare with definition of sense 1 from '*couple*'. The total comparison is 4*4=16 and total value extracted from conceptual distance data is Y. Then in the third step, definition of sense 1 from '*couple*' will compare with definition of sense 1 from '*marry*'. The total comparison is 4*3=12 and total value extracted from conceptual distance data is Z. The calculations for second step and third are as same as the step.

So, the path value for father1 marry1 couple1 = 2(X/12) +Y/16 + 2(Z/12).

Formula used to compute path value:

$$\text{Path value} = \sum_{i=1}^{n} (\text{distance})(s\_i \,/\, \text{total comparison})$$

where n = the total of words sense pairs , s = the total summation of values getting from conceptual distance data for *i*-th of word sense pairs.

This process will continue for other combination of word senses:

father1	marry1	couple2
father1	marry2	couple1
father1	marry2	couple2
father2	marry1	couple1
father2	marry1	couple2
father2	marry2	couple1
father2	marry2	couple2

The total combination of word sense for this example is 2*2*2=8. Finally, this algorithm will compare the path values for the combination of word senses in a sentence and find the most suitable combination of word senses. (Please refer to Figure 2)

2 Result

System	Number of primitives	Course Grained Precision/ Recall	Fine Grained Precision/ Recall
usm1	492	35.5% / 34.7%	34.5% / 33.8%
usm2	478	37.0% / 37.0%	36.0% / 36.0%
usm3	4000	34.4% / 34.4%	33.6% / 33.6%

Table 1: SENSEVAL-2 English All Word Results

(note:usm = Universiti Sains Malaysia)

With reference to the above table, usm1, usm2 and usm3 are three systems that are different in the number of primitives used in MTD as well as in the conceptual distance data and also MTD used. MTD used in usm1 is less comprehensive compare to MTD used in usm2 and usm3. More comprehensive is meaning that each of the entries is represented by a more complete set of primitives. MTD used in usm2 and usm3 is the same. Because of we are focusing more on speed of the system, overall of the results decreases when only the head words are considered.

3 Future extension of the system

In order to improve the existing algorithm, we need to avoid repeated calculation especially repeated comparison among the primitives. The concept of dynamic programming is needed to reduce the calculation. Basically, by using this method, result of the calculation is stored in memory so that the result can be accessed easily later when it is needed. As a result, although this method will increase the memory usage, it can also increase speed of the calculation significantly especially when a long sentence is processed. This is important because since the speed of the algorithm is increasing, it can be used in the real time application such information retrieval system especially in the Internet.

In additional, the accuracy of the system can be increased because more words in a sentence can be considered when a target word is tagged.

It is also important to note that this system not only can be used for English language, it can also be used in the other languages such Bahasa Malaysia, Chinese and Japanese.

Conclusion

In this paper, we have illustrated the overall architecture of our application of unsupervised learning technique to word sense disambiguation. Besides that, we have also presented that how our application in handling the given sentence and how we manage to complete the English all task given by SENSEVAL-2 competition. In additional, we illustrated the improvement over the algorithm we have presented in this paper. This is to make the algorithm becoming more efficient and practical to implement in real time application.

References

Guo, C-M (1989a) "Constructing a Machine-Tractable Dictionary from Longman Dictionary of Contemporary English.". Doctoral dissertation. New Mexico State University.

Guo, C-M (1989b) "Deriving a natural set of semantic primitives from Longman Dictionary of Contemporary English." Proceedings of the Second Irish Conference on Artificial Intelligence and Cognitive Science. 218-227

Wilks, Y. (1977) "Good and Bad Arguments About Semantic Primitives." In Communication and Cognition, Vol 10, No 3/4.

Y. Choueka and S.Lusigna. (1985). "Disambiguation by Short Contexts". Computer and the Humanities. 19:147-157.

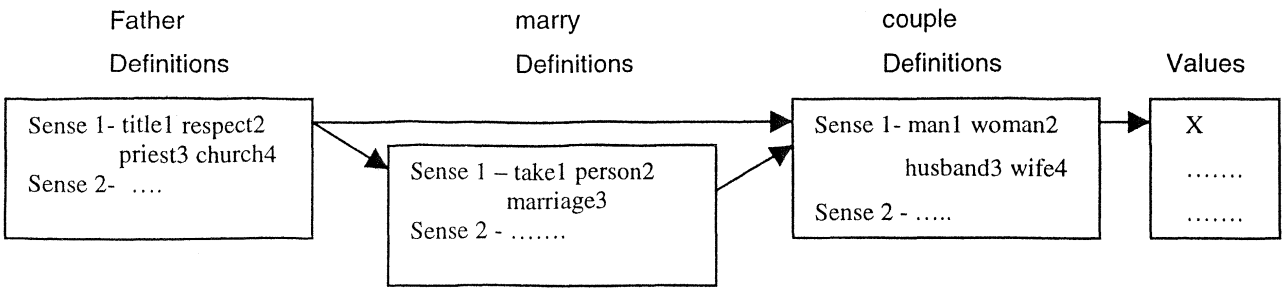

Figure 1: Example for sense-pair comparison among the words in a sentence.

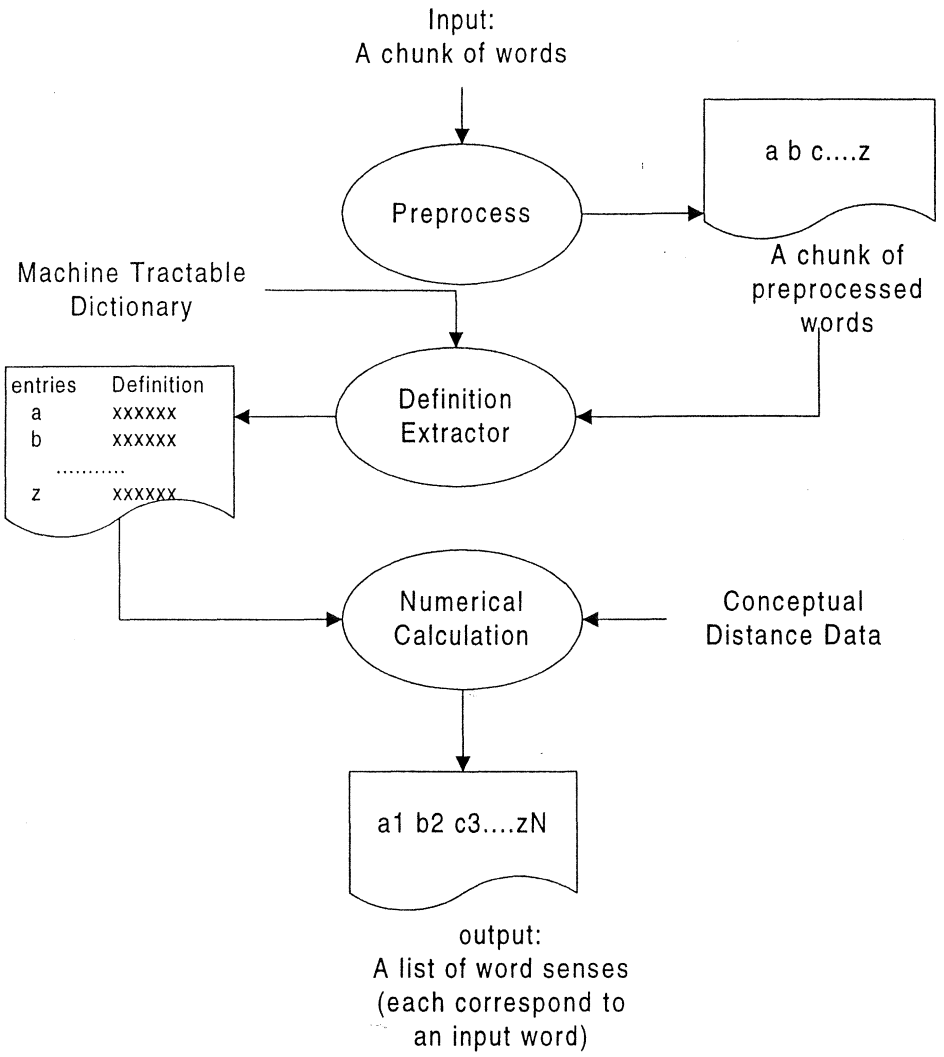

Figure 2: Sense Tagger