Word Sense Disambiguation Corpora Acquisition via Confirmation Code

Wanxiang Che and Ting Liu*
Research Center for Social Computing and Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology, China
{car, tliu}@ir.hit.edu.cn

Abstract

Word Sense Disambiguation (WSD) is one of the fundamental natural language processing tasks. However, lack of training corpora is a bottleneck to construct a high accurate all-words WSD system. Annotating a large-scale corpus by experts costs enormous time and financial resources. Human Computation is a novel idea for integrating human resources behind the Web, which has been wasted, to solve practical problems that are difficult for computers. Based on human computation, we design a confirmation code system, which can not only distinguish between human beings and computers (the function of normal confirmation code system), but also annotate WSD corpora. The preliminary experimental result shows that the proposed method can annotate large-scale and high-quality WSD corpora within a short time. To the best of our knowledge, this is the first attempt to use confirmation code in natural language processing for corpora acquisition.

1 Introduction

It is a common phenomenon that a word has multiple senses in natural languages. The aim of word sense disambiguation (WSD) is to identify the correct senses of ambiguous words according to their surrounding contexts. WSD is a basic task of natural language processing. The state-of-the-art in WSD is dominated by supervised machine learning methods where a model is trained to recognize word senses in a given context based on various features. Although it is important to use powerful machine learning algorithms, latest studies have found that large-scale and high-quality corpora are more important for WSD (Agirre and Edmonds, 2006). Therefore, building such corpora is key challenge to be addressed.

Currently, corpora are created mainly through manual annotation by expert annotators. However, the cost of annotating a necessary size of corpora is prohibitive. Consequently, in the research field of WSD, some common ambiguous words are sampled and then annotated with a necessary amount of examples. The sampling method promotes the research in WSD algorithms. However, these algorithms are difficult to be used in practical applications due to lack of large-scale corpora in which all ambiguous words are annotated. For example, SemCor[1] corpus contains WSD annotation of about 250,000 words sampled from a subset of the Brown corpus. However, for most of the ambiguous words, the number of examples is still too small to train a high performance all-words WSD model. The best performance of Senseval-3 English all-words evaluation task (Snyder and Palmer, 2004) is only about 65%.

Semi-supervised methods have been applied to build large-scale corpora, such as bootstrapping (Yarowsky, 1995). However, the quality of corpora built with such methods is not high enough to train accurate WSD models. Therefore, the methods are not feasible to be used in practice.

Crowdsourcing is an "online, distributed problem-solving and production model. (Brabham, 2008)" A benefit of this distributed model is that a job can be shared amongst a wide variety of demographics, where such diversity would be difficult to obtain otherwise. Previous research has demonstrated the successful application of crowdsourcing in a variety of natural language processing areas including relevance evaluation (Alonso et al., 2008), machine

*Correspondence author: tliu@ir.hit.edu.cn

[1] http://www.cse.unt.edu/∼rada/downloads.html#semcor

translation (Ambati et al., 2010), and language processing (Callison-Burch and Dredze, 2010). However, the submissions of crowdsourcing are always needed to review to separate the legitimate work from the rest. Additionally, the crowdworkers are always motivated by money. These increase the extra cost of time and finance.

Human Computation is a novel method for collecting corpora (von Ahn, 2007). In this method, a computer asks a person or a large group of people to solve a problem, and then collects, interprets, and integrates their solutions. The methods of human computation include interactive online games, confirmation code, and so on. For instance, reCAPTCHA (von Ahn et al., 2008) system is a kind of confirmation code, also called CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). Confirmation code systems are widely used on the Web as security measures to prevent automatic programs from abusing online services by asking a question that computers cannot yet answer. In reCAPTCHA, to pass the confirmation stage, users must input the content of word images from scanned old books. Two scanned word images are shown to a user at the same time. The system knows the content of one word and does not know the other. If the user wants to pass the confirmation stage, he must input the correct content of the known word. Because the user does not know which word the system knows, he has to input the contents of both words. Thus, once the confirmation function is successfully achieved, the content of the unknown word can be obtained. Finally, reCAPTCHA helps to digitize plenty of old books that an optical character recognition (OCR) cannot decipher.

Different from crowdsourcing, human computation does not need to review the results again nor pay users. The method can take advantage of a larger range of people. However, human computation methods are rarely used in natural language processing tasks. The main reason is that natural language processing tasks are usually very complex. It is difficult to design an appropriate game and also be annotated by normal users. Seemakurty et al. (2010) designed a game which helps to collect WSD corpora. The game chooses two participants randomly and then shows a sentence to them. The two users are asked to input as many synonyms of the same ambiguous word in the sentence as possible within a limited period of game round time. Once there are identical input words by them, they are awarded scores and then a synonym of the ambiguous word is obtained. Based on these synonyms, the correct sense of the ambiguous word can be recognized. However, a big problem is how to attract a considerable number of users. In addition, it needs long time to collect these synonyms[2]. More seriously, the game can be cheated under some extreme circumstances, e.g., when all users just input the same words, or we use a robot to input all words in a dictionary quickly.

In this paper, we are inspired by the idea of reCAPTCHA system and propose a confirmation code based method to annotate WSD corpora with low cost. The method can help to collect large-scale and high-quality corpora within a short time. Preliminary experimental result shows that the method can achieve 80.65% accuracy on an annotated WSD corpus which is close to the inter-rater agreement of the corpus. It only needs about 8 to 10 seconds to annotate an example by a person.

2 System Description

A WSD confirmation code includes two questions. Each question consists of a sentence and a highlighted ambiguous word in the sentence. All senses of the ambiguous word are provided as optional answers. The system only knows the answer for one of the two questions, which is named as *known* question and the other is *unknown* question. A user needs to choose a word sense for each ambiguous word. The user can pass the confirmation stage if and only if his answer to the known question is correct. Like in reCAPTCHA, users do not know which one is known question. They must choose each word sense carefully in order to pass confirmation stage. Therefore, they provide the correct sense for the ambiguous word of unknown question. If WSD confirmation code system is widely used by lots of Web sites, we can easily collect large-scale corpora.

The data flow chart of the WSD confirmation code system is shown in Figure 1. ① two questions are randomly selected from known and unknown question databases respectively. ② the two questions are asked to a user and the user needs to answer them. ③ once the user's answer is correct,

[2]In their work, a game round is to be set 30 seconds, i.e. it needs 30 seconds to annotate the sense of an ambiguous word at least.

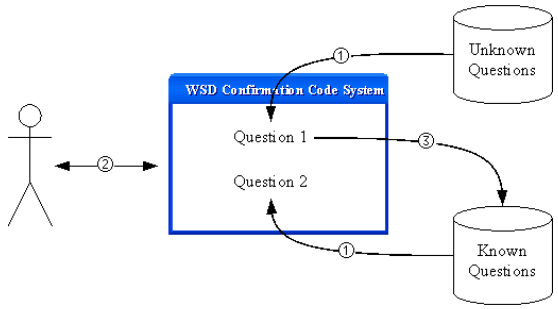

Figure 1: The data flow chart of WSD confirmation code system

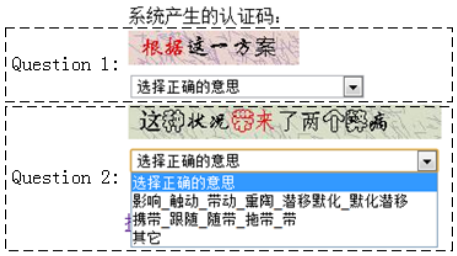

Figure 2: An example of WSD confirmation code system. Here, the two sentences are "根据这一方案 (according to the scheme)" and "这种状况带来了两个弊端 (this situation brings two drawbacks)" respectively. The two ambiguous words are "根据 (according to)" and "带来 (bring)". Here, "带来" has two senses: "影响 触动 ··· (influence)" and "携带 跟随 ··· (bring)". Because of the incompleteness of the thesaurus, there are some words whose senses cannot be found. Therefore, we add a "其它 (other)" option for every word since the current word sense does not belong to any of above options. The other Chinese sentences in the example instruct the users how to use the system. "系统产生的验证码" means "The confirmation codes provided by the system" and "选择正确的意思" means "Please choose the correct word sense".

i.e. it is equal to the answer of known question, he can pass the confirmation stage. Then we can know the answer of the unknown question, which becomes a new known question and can be added into the known question database. Otherwise, the confirmation stage cannot be passed and the user has to answer another pair of questions.

The known question is used to distinguish between human beings and computers, i.e. this is the function of commonly used confirmation code. Usually, it is either impossible or less possible for a computer to automatically choose correct answers. In order to further prevent automatic WSD programs from passing the confirmation stage, we convert original sentences into images with randomly distorted background and font. This method makes it impossible to recognize the contents of the original sentences and to do WSD automatically. If users correctly annotate the known word sense, the system can assume that they are human and gain confidence that they can also annotate the other word sense correctly.

Figure 2 shows an example of WSD confirmation code[3]. Sentences are in image forms and the target ambiguous words are highlighted with red font. The senses of these ambiguous words are shown in pull-down menus.

In order to improve the consistency of the final corpora, we allow each example to be annotated more than once. Then, a voting method can be used to determine the final word sense.

3 Experiment

3.1 Experimental Data

To evaluate the correctness of the corpus annotated by our WSD confirmation code method, we built a trial system and took advantage of an annotated Chinese all-words WSD corpus consisting of about 10,000 sentences containing about 200,000 words sampled from Chinese news documents. In these words, there are about 78,800 ambiguous words and all of which have been annotated with their corresponding senses by human experts. Among them, we randomly set 5,000 words as unknown questions and remaining as known.

3.2 Thesaurus

We use WordMap (Che et al., 2010) as a thesaurus to represent word senses. There are more than 100,000 Chinese words in WordMap. Each word sense belongs to a tree node with five levels. There are 12 top level nodes, such as "entity" and "human beings". There are about 100 second, 15,000 third, and more fourth and fifth level nodes. Under the fifth level nodes, there are some synonyms which have the same word sense. For instance, the word "材料" has two senses which are represented by five level nodes as follows:

1. 物 (entity) → 统称 (common name) → 物资 (goods) → 物资 (goods) → 材料 (material)

[3]In this paper, we use Chinese as an example. However, the method is not restricted to specific language.

# of times an example is annotated	# of annotated examples	# of correct annotated examples	Acc.
Random			41.01%
1	2,387	1,609	67.41%
≥2	336	271	80.65%

Table 1: Comparison of Experimental Results

2. 人 (human beings) → 才识 (ability) → 俊杰 (hero) → 人才 (talents) → 人才 (talents)

We can see that the two word senses belong to two top level nodes "物 (entity)" and "人 (human beings)" respectively. In each sense, the concept becomes more and more specific as the level increases.

However, the above sense representation method is not suitable to be shown as word sense options directly since it is too abstract to be understood by normal users. Therefore, we use synonyms of each word sense to represent the word sense. For instance, the synonyms of the two senses of the word "材料" are "材质 生料 质料 (materials)" and "人才 佳人 才子 奇才 天才 (talents)" respectively. Then we show these synonyms to the users for better understanding of the word sense.

3.3 Preliminary Results

We invited 20 volunteers to test the system. We collected 2,723 examples which passed the confirmation successfully. Table 1 shows the comparison of the experimental results.

From Table 1, we can see that when an example is annotated once, the accuracy (67.41%) is much higher than the random sense selection (41.01%)[4]. This tells us that the human efforts have a positive effect on the annotation. However, the accuracy is still not high enough and this can be attributed to volunteers' lack of experience in WSD. They maybe make mistakes. The accuracy, along with an increase of the annotation times, is improved. When an example is annotated more than once, the accuracy reaches 80.65% and is close to the inter-rater agreement (83.84%) of the original corpus.

On average, it needs about 8 to 10 seconds for a person to successfully input a WSD confirmation code. It is faster than common confirmation code systems (with six to eight randomly characters), which need 13.51 seconds on average (von Ahn et al., 2008). This is not surprising, because choosing an answer is faster than inputting some characters. So, in practice, the WSD confirmation code system can be adopted without reducing the quality of user experience.

[4]In WordMap, there are 2.44 senses for each ambiguous word on average. Therefore, the accuracy of random selection is 41.01%.

4 Conclusion and Future Work

To address the lack of WSD corpora, we propose a human computation based method. When users successfully input a confirmation code, they annotate a WSD example incidentally. The preliminary experiments show that the novel method can annotate large-scale and high-quality WSD corpora within a short time. As far as we know, there is no work done to annotate natural language processing corpora with confirmation code.

In the future, we plan to improve the annotation speed and reduce the complexity of confirmation process by showing two sentences with the same ambiguous words. Thus, users can easily compare the two sentences. More importantly, they only need to read the options once, which can save confirmation time further. We also can use unsupervised clustering method which determines senses that are very similar and displays only one of the alternatives. Secondly, we will apply this method to other languages. Our method is general enough and can be applied to any languages as long as the language has a thesaurus and some initial WSD corpora. Of course, a particular language WSD confirmation code system can only be used in Web sites of the same language because it is difficult for a normal user to perform WSD task on the foreign language that they are unfamiliar with. Thirdly, we can apply this method to other natural language processing tasks which need corpora acquisition such as co-reference resolution, named entity recognition, and parsing. Finally, we will use the corpora annotated by the confirmation code method to train a more effective WSD model.

Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 61133012, Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069), and Fundamental Research Funds for the Central Universities (HIT.KLOF.2010064).

References

Eneko Agirre and Philip Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer, 1 edition, July.

Omar Alonso, Daniel E. Rose, and Benjamin Stewart. 2008. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42:9–15, November.

Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2010. Active learning and crowd-sourcing for machine translation. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

D C Brabham. 2008. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):75–90.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. In *Workshop on Creating Speech and Language Data With Mechanical Turk at NAACL-HLT*.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Coling 2010: Demonstrations*, pages 13–16, Beijing, China, August. Coling 2010 Organizing Committee.

Nitin Seemakurty, Jonathan Chu, Luis von Ahn, and Anthony Tomasic. 2010. Word sense disambiguation via human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 60–63. ACM.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, July. Association for Computational Linguistics.

Luis von Ahn, Ben Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468.

Luis von Ahn. 2007. Human computation. In *Proceedings of the 4th international conference on Knowledge capture*, K-CAP '07, pages 5–6, New York, NY, USA. ACM.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA. Association for Computational Linguistics.