Combining ConceptNet and WordNet for Word Sense Disambiguation

Junpeng Chen
Computer School, Wuhan University,
Wuhan, P. R. China
chenjp@whu.edu.cn

Juan Liu*
Computer School, Wuhan University,
Wuhan, P. R. China
liujuan@whu.edu.cn

Abstract

Knowledge-based Word sense Disambiguation (WSD) methods heavily depend on knowledge. Therefore enriching knowledge is one of the most important issues in WSD. This paper proposes a novel idea of combining WordNet and ConceptNet for WSD. First, we present a novel method to automatically disambiguate the concepts in ConceptNet; and then we enrich WordNet with large amounts of semantic relations from the disambiguated ConceptNet for WSD. The evaluation experiments on the Semeval-2007 coarse-grained all-words disambiguation task show that the enriched WordNet can significantly improve the performance of knowledge-based WSD methods.

1 Introduction & Motivation

1.1 Introduction

Word sense Disambiguation (WSD) is a task of selecting the proper sense of ambiguous terms in the text. WSD is an intermediary step within many Natural Language Processing (NLP) applications, such as text summarization, machine translation, text processing, and so on. Finding a solution to the WSD problem is essential or even compulsory for such NLP applications.

There are two main classes of WSD approaches: supervised WSD, and knowledge-based WSD. The former employs statistical learning to learn a classifier from training data. The preparation for these training data may be laborious and erroneous. Moreover, building a manually annotated corpus, as required by supervised WSD methods, to cover all word senses in lexicon is expensive, and even infeasible. In contrast, knowledge-based WSD methods rely on the use of wide-coverage knowledge resources. These methods do not need the human labeled training data. The widely used knowledge resource in such methods is WordNet (Fellbaum, 1998). However, WordNet-based WSD methods usually achieved lower performance compared to supervised methods, mainly due the fact that the lexical and semantic knowledge contained in WordNet is not sufficient for WSD.

Therefore, many methods (see Section2) have emerged to enrich WordNet with the lexical and semantic knowledge for WSD purpose, such as by using Wikipedia, on-line lexicons, domain, and so on. The literature research results show that the use of enriched knowledge does improve the performance of WordNet based WSD systems, in terms of both accuracy and coverage.

However, most of the present methods mainly focus on enriching WordNet with limited lexical and taxonomic knowledge. Though Wikipedia contains many semantic relations, the knowledge extracted from Wikipedia may contain noises, for some of them are derived from weak semantic links, and lack of confidence.

In this paper, we propose to use ConceptNet (Havasi&Alonso, 2007) with the large amounts of semantic relations between concepts, to enrich WordNet. For there are many ambiguous concepts existed, ConceptNet cannot be directly used to enrich WordNet. Thus, we develop a novel methodology to automatically disambiguate the ambiguous ConceptNet concepts. By using the disambiguated ConceptNet, we can enrich semantic relations in WordNet. To evaluate performance of WSD based on the extended WordNet, we implement a simple knowledge-based algorithm and its extension, embedded with WordNet, WordNet+ConceptNet, respectively. The comparison results show that using WordNet along with disambiguated ConceptNet can make even simple knowledge-based algorithms achieve state-of-the-art performances.

*the corresponding author

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 686–694,
Chiang Mai, Thailand, November 8 – 13, 2011.

1.2 ConceptNet

ConceptNet is a large collaborative Web knowledge resource, which encompasses commonsense knowledge about the spatial, physical social, temporal and psychological aspects of everyday life (e.g., airplane is capable of flight, plane relates to geometry). It is useful in a wide variety of applications such as speech recognition (Lieberman et al., 2005), intelligent user interface (Speer et al., 2009), machine translation (Caseli et al., 2010), and so on.

Until 2011, ConceptNet contains nearly one million of assertions represented as triplets like *<concpet1, relation, concept2>*, to define the concrete semantic relations between two specific concepts. All the assertions are organized as a semantic network, where a node stands for a concept, and an edge stands for a relation between two concepts. There are above 400,000 concepts in ConceptNet, each of them are denoted as either a word or a phrase, and labeled by a unique identifier. In addition, ConceptNet defines nearly thirty kinds of semantic relations, such as CapableOf (agent's ability), SubeventOf (event hierarchy), MotivationOf (affect), DesireOf (want to), and so on, most of which are not included in WordNet. Therefore, if extending WordNet with the large amounts of semantic relations contained in ConceptNet, it is desirable to improve the performances of WordNet-based WSD methods.

However, ConceptNet cannot be directly used for WSD purpose due to the existence of polysemy and synonymy of the concepts in it.

Polysemy is the tendency for ConceptNet concepts to have multiple senses. For example, a ConceptNet concept *plane* has two word senses: "a fixed-wing aircraft", or "an unbounded two-dimensional shape". Which sense should be used depends on the considered assertion including the concept. Therefore, we should first assign the correct senses for those concepts in the assertions before using them to enrich WordNet.

Synonymy is another tendency for the concepts in ConceptNet to have a common word sense. For example, the concept *airplane* has only one sense: "a fix-wing aircraft", which is also the first sense of the concept *plane*. It is obvious that the concepts related to *airplane* should have the same relation with *plane*. However, it is not the case in ConceptNet. Concept *airplane* has an atLocation relation with the concept *airplane hangar*, whereas *plane* has not such relation with *airplane hangar*. This leads to the inconsistency of knowledge base. Therefore assigning the correct senses for the ambiguous concepts in the assertions to find the synonym concepts will improve the quality of the knowledge base.

2 Related Work

Up to now, there are many approaches to enrich the knowledge of WordNet for WSD tasks. For instances, Magnini&Cavagli à(2000) proposed to use domain knowledge to assign domain labels to most WordNet synsets. Some researchers (Mihalcea&Moldovan, 2001; Navigli, 2009; Hwang et al., 2011) proposed to enrich semantic relations by means of the disambiguation of the glosses of WordNet or other machine-readable dictionaries. Some other researches (Agirre et al., 2000; Cuadros&Rigau, 2008) extract semantic relations from Web to enrich WordNet. However, all above methods mainly aim to enrich lexical and taxonomic resources. Therefore some recent work (Mihalcea, 2007; Ponzetto&Navigli, 2010) exploits Wikipedia, a large collaborative Web encyclopedia, to extract the knowledge for WSD. However, the type of semantic relations extracted from Wikipedia is uncertain. Moreover, it is hard to know which semantic relations are transitive or belong to the same type (e.g. isA, part of).

Different with the existed methods, we propose to use Conceptnet to enrich the knowledge in WordNet, which has several advantages over previous works. First, ConceptNet is a large-scale commonsense knowledge base for many aspects of everyday life, such as spatial, physical social, temporal, psychological, and so on. Injecting such knowledge from ConceptNet into WSD system can effectively relieve the knowledge acquisition problem in WSD. Second, the semantic relations from some of the previous work such as Wikipedia are extracted in an indirect way, each of which has no a clear relation type. Thus some of those semantic relations are too weak to be filtered (Ponzetto&Navigli, 2010). In contrast, the semantic relations in ConceptNet are directly defined as assertions, each of which has a very clear relation type. Therefore, the semantic relations in ConceptNet are expected to be more robust than the others.

3 Disambiguating ConceptNet

For there are many ambiguous concepts existed in ConceptNet, it cannot be directly used to enrich WordNet for WSD. It is necessary to disambiguate the ambiguous ConceptNet concepts.

Assume concept *plane* have two senses. The first one is "a fixed-wing aircraft"; and the second one is "an unbounded two-dimensional shape". Given a ConceptNet assertion <*plane, usedFor, fly*>, one can easily judge the appropriate sense of *plane* in this assertion to be the first sense of *plane*. That is because people do not simply regard a sense of a word as an abstract symbol, but a concrete entity that has many properties. For the first sense of *plane*, people may think that it is an *aircraft*, also named *airplane*, and has *wings*, etc. For the second sense of *plane*, people may think that it is a *shape*, or a *form*, and relates to *mathematics*, etc. The concepts *aircraft*, *airplane*, and *wing* relate to *fly* more closely than *shape*, *form*, and *mathematics*. By integrating such information, people can easily know the correct sense of *plane* in the assertion <*plane, usedFor, fly*>.

We propose a method to simulate the human's processing of disambiguating ambiguous concepts by three steps. Given a ConceptNet assertion <*c, relation, d*>, where concept *c* is ambiguous (the cases of *d* being ambiguous or both *c* and *d* being ambiguous are the similar), in order to disambiguate *c* in this assertion, firstly, we construct a word sense profile (WSP) for each sense of *c*. A WSP is a set of terms (words) relating to a sense, it describes the sense in a whole (Section 3.1). Secondly, we measure the relatedness between the terms in WSP with *d* in the same assertion based on NGD (Section 3.2). Thirdly, we filter out the noisy terms in WSP, which would decrease the performance of ConceptNet disambiguating (Section 3.3).

As a result, we calculate the score of the WSP for each sense, and choose the sense with the lowest WSP score as appreciated one for the ambiguous concept in the assertion. Therefore, for each ambiguous concept of every ConceptNet assertion, we can assign the appropriate sense to it according to the WSP scores; and the resulted ConceptNet can be used to extend WordNet.

3.1 Constructing Word Sense Profile

As we all know, WordNet is structured as a semantic network in which nodes stand for a concept **sense**, and are linked by a small set of semantic relations such as hypernymy, hyponymy, meronymy, and so on. For ambiguous concepts with multiple senses, there are multiple nodes in the network. The concept sense is represented by a **synset** (a set of words sharing a common meaning, each word is called a synonymy in the synset). For an example, for concept *plane*, we can use plane_n^1 as the label of one of its senses, in which the subscript and superscript indicate its part of speech (e.g. "n" stands for noun) and sense no., respectively; and its synset is denoted as plane_n^1 = (airplane, aeroplane, plane), illustrating that this synset is consist of three synonymys: *airplane*, *aeroplane* and *plane*. Moreover, each synset has a textual definition, namely **gloss**. For instance, the gloss of synset plane_n^1 is "an aircraft that has a fixed wing and is powered by propellers or jets".

Given a WordNet synset *S*, we make use of the following knowledge resources to construct its **Word Sense Profile**, WSP(*S*).

Synonymy: all synonyms in *S*. For an example, three synonyms in the synset plane_n^1 will all be included in the WSP(plane_n^1).

Hypernymy/Hyponymy: all synonyms in the hypernym synset *H* of *S* (e.g., *S* is a kind of *H*) or in the hyponym synset *H* of *S* (e.g., *H* is a kind of *S*). For instance, the hypernym of synset plane_n^1 is $\text{heavier - than - air craft}_n^1$ = (heavier-than-air craft), then the synonym *heavier-than-air craft* will also be included in WSP(plane_n^1).

Meronymy/Holonymy: all synonyms in synsets *M* which has a meronymy (e.g., *M* is a part of *S*) or a holonymy (e.g., *S* is a part of *M*) relation with *S*, will be containd in WSP of S. For example, given a synset plane_n^1, one of the meronymies plane_n^1 is pod_n^4=(pod, fuel pod), so *pod* and *fuel pod* will also be included in WSP(plane_n^1).

Gloss: the set of words in the gloss of *S*. For example, the gloss of synset plane_n^1 is "an aircraft that has a fixed wing and is powered by propellers or jets". After removing the stop words, the remaining words will be included in WSP(plane_n^1).

Indirect Resources: Besides above direct relations in WordNet, we also use some indirect ones that are derived from the transitivity of WordNet semantic relations, to construct the WSP. Given a synset *A*, if *A* has a direct relation with synset *B*, *B* has a direct relation with synset *C*, and then *A* has an indirect relation with *C*. Therefore, *C* is regarded as an indirect resources for *A* and all synonyms of *C* are also added in the WSP(*A*).

For a synset *S*, WSP(*S*) is defined as the set of words obtained from direct or indirect resources.

3.2 Measuring Relatedness

Given a ConceptNet assertion <*c, relation, d*>, after getting the WSP of each sense of the am-

biguous concept *c*, we need to know which sense is the most likely one in this assertion by calculating a score for the WSP. To do so, we first measure the relatedness between a term in the WSP and *d*, and then compute the arithmetic mean of the values as the score of the WSP.

Normalized Google Distance (NGD) (Cilibrasi et al., 2007) was proposed to measure semantic relatedness between two terms using the vast available knowledge on the Web. Concretely, NGD takes advantage of the number of hits returned by search engine such as Google, to compute the distance between terms. Small NGD value indicates close relatedness, while large value suggests the opposite. Given a term pair <*x, y*>, the normalized Google distance between *x* and *y*, NGD(*x, y*), can be obtained as follows:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (1)$$

Where *f(x)* is the number of Google hits for the term *x*, *f(y)* is the number of Google hits for the term *y*, *f(x, y)* is the number of Google hits for both terms *x* and *y*, and *N* is the number of web pages indexed by Google. The smaller the NGD score, the more related the two terms are.

According to the definition, it is desirable to use NGD to measure the relatedness between any term in the WSP and *d*.

Note that we always compute the NGD scores in the context of ConceptNet assertion <*c, relation, d*>. That is, we only need to consider the relatedness between *d* and the term in WSP(*c*). Therefore, we sometimes simply say the NGD score of a term in this paper without confusion.

3.3 Filtering the Noise Terms

Ideally, the NGD score of any term in the WSP of the correct sense is lower than that in the WSP of incorrect sense, thus we can simply use the arithmetic mean of the scores to evaluate the relatedness of WSP and the assertion. However, it is inevitable that there are some noisy terms in WSPs, which will dramatically decrease the performance of disambiguating ConceptNet. Therefore, we need to pay efforts to reduce such noises.

There are two kinds of noises: those from WSP of the correct sense, and those from WSP of the wrong sense(s). Suppose we have an assertion <*c, relation, d*>, where concept *c* is ambiguous and has two senses c_n^1 and c_n^2. Corresponding to the assertion, the correct sense of *c* is c_n^1, and a wrong sense of *c* is c_n^2.

Then the first kind of noises are from WSP(c_n^1), which do not have close relation to *d*. Thus their NGD scores should be high, though they have a WordNet relation to c_n^1. For an example, given a ConceptNet assertion <*plane, isA, vehicle*>, the correct sense of *plane* is $plane_n^1$ (a fixed-wing aircraft). Term *navigation light* is in WSP($plane_n^1$), however there is no close relatedness between *navigation light* and *vehicle*, resulting that it is a noisy term in WSP($plane_n^1$), belonging to the first group. Obviously, this kind of noises have high NGD scores and would increase the score of the WSP of the correct sense, thus correspondingly decrease the probability of selecting the correct sense as the appreciated sense of ambiguous concept.

The second kind of noises are from WSP(c_n^2), which have close relation to *d* whereas occur in the incorrect WSP. Since they are closely related to *d*, their NGD scores are usually low, hence lower the score of the WSP of the wrong sense, which may lead to selecting the wrong sense as the appreciated one for ambiguous concept. Such noises mainly come from the ambiguous terms contained in the WSPs. For an instance, *basketball* has two word senses: $basketball_n^1$ ("a game played on a court by two opposing teams of 5 players"), or $basketball_n^2$ ("an inflated ball used in playing basketball"). Based on WordNet, it is easy to get WSP($basketball_n^1$) = (game, handball,…), and WSP($basketball_n^2$) = (ball, handball,…). We can see that *handball* occurs in both WSPs. In fact, *handball* is itself ambiguous and has two senses: $handball_n^1$ ("a small rubber ball used in playing the game of handball"), $handball_n^2$ ("a game played in a walled court"). Given a ConcepNet assertion <*basketball, isA, popular sport*>, it is clear that the correct sense of *basketball* should be $basketball_n^1$, while $basketball_n^2$ is not the appropriate one for this assertion. It is desirable that all terms in WSP($basketball_n^2$) have high NGD scores. However, it is not the case for term *handball*. According to $handball_n^1$, we know that the NGD score of *handball* should be low, for *handball* and *popular sport* frequently co-occur in the Web. There is no problem for the occurrence of *handball* in WSP($handball_n^1$). However, it is because the computation of NGD score does not consider different senses of the same term in different occurrences, different *handball*s in different WSPs actually have the

same NGD scores though they correspond to different senses. As a result, the term "*handball*" in WSP(basketball_n^2) becomes a noise to the assertion <*basketball, isA, popular sport*> for it improperly lower the score of WSP(basketball_n^2) and increase the risk of assigning the sense basketball_n^2 to concept *basketball* in the assertion.

Of course, in real applications, it is impossible to know which kind of noises exist and where they are in advance. Moreover, whether a term in a WSP is a noise may depend on the specific ConceptNet assertion, just as the *handball* in above example. We simply try three filters by using different strategies in the calculation of WSP scores: TopN filter, Threshold filter and Top r% filter.

TopN filter only selects the top *n* terms with the smallest NGD scores to take part in the calculation of WSP score. Threshold filter only retains the terms with NGD scores lower than a predetermined threshold value *t* to compute WSP score. Top r% filter supposes that the number of noise terms may be proportionate to the size of WSP, and only selects the top *r%* terms with the smallest NGD scores to compute the WSP score.

Obviously, all of above strategies aim to reducing the first kind of noises by disregarding the terms with larger NGD scores. In fact, it is hard to filter out the second kind of noises for most of them are related to the specific concept sense, which keeps unknown before the calculation of WSP scores. Furthermore, there is also the dilemma due to the filtering that the second kind of noises are more likely to be retained for they have lower NGD scores.

Nevertheless, we could still avert the second kind of noises to some extent, not by filtering, but by preventing them from being added into the WSP. Now that the second kind of noises mainly comes from the ambiguous terms, we can analyze which relationships defined in Section 3.1 may introduce more ambiguous terms thus could not be taken into consideration. We will address this problem in Section 4.1.3.

4 Experiments

Our evaluation experiments consist of two steps. The first step is to evaluate the intrinsic quality of the disambiguated ConceptNet (Section 4.1), including the selection of the noise filters and analysis of effects of different combinations of WordNet relationships on the disambiguation results. The second step is to evaluate the impact of combining disambiguated ConceptNet and WordNet for coarse-grained WSD by comparing different methods (Section 4.2).

4.1 Evaluation of the Disambiguating ConceptNet

To our best knowledge, there is no literature work for ConceptNet disambiguation before. So we first generate the test bed as the gold standard for evaluations; and then compare our disambiguated concepts to this gold standard to see how many of them are matched. We also compare different noise filtering strategies mentioned in Section 3.3 and accordingly choose the best one as the final noise filter. To avoid introducing more second kind of noises when generating WSPs, we also investigate different combinations of WordNet relationships mentioned in Section 3.1 by comparing by disambiguating accuracies.

4.1.1 Gold Standard Generation

To evaluate our ConcepNet disambiguating methods, we created a gold standard data as follows. First, we randomly selected a set of 1,000 assertions from the ConceptNet, and checked whether the included concepts have multiple senses in WordNet. By doing so, we found 425 ambiguous concepts with more than one WordNet senses, which are contained in 365 of the 1,000 assertions. Then we asked a language expert to annotate the WordNet senses for the ambiguous concepts in these ConceptNet assertions. To see whether the annotations are convincible, we asked a different expert to tag the same 425 ambiguous concepts in the 365 assertions independently. The kappa coefficient (Carletta, 1996), which is used to calculate the inter-annotator agreement, show that the two annotations achieved a perfect agreement with coefficient as 0.9. Therefore, we use the first annotation results as the gold standard to evaluate our methods.

4.1.2 Comparison of Three Filter Methods

To compare different filtering strategies mentioned in Section 3.3, we first construct WSP for each sense of the 425 concepts, and then calculating the WSP scores after noise filtering by different strategies separately; finally, we annotate each concept with a sense according to the WSP scores. We also did the concept annotation by calculating the WSP scores without noise filtering for reference. In the construction of WSP, we made use of the following six WordNet resources: synonymy, hypernymy, hyponymy,

meronymy/holonymy, gloss, and hyponymy → gloss (the gloss of the hyponymy; the similar denotations are also used in the following). Section 4.1.3 will show the reason.

To investigate the significance of the WSP score based annotation method against the random or dominant annotation, we also assigned a sense to each concept, by randomly selecting one from the available senses and by selecting the most frequent sense, respectively.

Table 1 shows annotation accuracies of different methods, where "MFS BL" and "Random BL" correspond to the performances of random and dominant annotation respectively. In the experiments, we set n=10 for TopN filter, t=0.2 for Threshold filter, and r=20 for Top r% filter. These parameter values were set by our experience, to achieve the best performances on the 425 concepts.

Filter	Accuracy
TopN Filter	**82.4%**
Threshold Filter	61.4%
Top r% Filter	60.5%
No Filter	44.0%
MFS BL	57.4%
Random BL	20.8%

Table1. Performance of the disambiguating ConceptNet using different filter methods

From Table 1 we can see that, the annotation results of WSP based methods are much more significant than the random method, and the use of noise filtering can significantly enhance the accuracy of the annotation. Without noise filtering, the WSP based method shows even worse performance than the MFS baseline (-17.4%), which demonstrates that there are actually some noise terms in WSP that decrease the performance of ConceptNet disambiguating.

In addition, we also see that though all of the filter methods can effectively filter out the noise terms from the WSP thus improve the performance of WSP method, TopN filter is significantly better than the other two (+21.9% and +21.0% compared to Top r% Filter and Threshold Filter respectively). This suggests that TopN would be the most appropriate filter method for removing the noise terms from WSPs. We think the result might due to the following factors. Firstly, Threshold filter cannot effectively remove the second kind of noise terms, because it may filter out too much right terms from the WSP of a wrong sense, while remain terms belonging to the second kind of noise. Therefore, there are not enough right terms to relieve the second kind of noise terms for the wrong sense. Secondly, the sizes of the WSPs may vary wildly. For the senses whose WSP size are small, Top r% filter may remain too few terms (e.g. only two or three terms), thus the WSP scores are very sensitive to the remain noises. Finally, for each sense, TopN filter remains a fixed number of terms. If the second kind of noise terms exists, there might be enough right terms to relieve the impact of those noise terms if we set the proper value to parameter n.

Therefore, in the comparative experiments, we only consider the TopN filter. In order to get a proper value of n, we investigated the performance of the filter by ranging n from 1 to 50 stepped by one and found that it is appropriate for TopN filter to set n in [6, 11]. In our work, we set n=10.

4.1.3 Investigation on the Resources Combination

Just mentioned in Section 3.3, WordNet resources defined in Section 3.1 may introduce the second kind of noises. Due to the fact that this kind of noises is hard to be filtered out, we should prevent such noises from entering the WSPs as possible as we can. Therefore, we tried different combinations and evaluate our ConceptNet disambiguating method, to investigate which combination is the best for our task.

Although hypernymy, hyponymy, and meronymy/holonymy are transitive, and can generate the indirect WordNet resources, the number of meronymy/holonymy is far below than those of the other two in the WordNet. Thus, we ignore the indirect WordNet resources derived from meronymy/holonymy. As a result, ten WordNet resources are used as the candidate resources to construct WSP. Five of them are direct WordNet resources (synonymy, hypernymy, hyponymy, meronymy/holonymy, and gloss), and five are indirect WordNet resources: hypernymy → hyponymy, hypernymy → gloss, hypernymy → hypernymy, hyponymy→hyponymy, and hyponymy→gloss.

Table 2 summarizes the highest accuracies of our ConceptNet disambiguating method with different combinations of WordNet resources against the 425 annotated concepts. The results show that our method achieves the highest accuracy with the six WordNet resources: hyponymy → gloss, gloss, hyponymy, hypernymy, mer-

onymy/holonymy, and synonymy. The combination of the six resources is abbreviated as "six resources" (We also tried to combine the direct resources with any of other four indirect resources, the results show that they cannot improve and even decrease the performance. Especially, hypernymy→hyponymy dramatically decreases accuracy (-8.94% compared to the "six resources"). Therefore, we do not list the accuracies of combining other indirect resources with the direct ones). We also noticed that based on the "six resources", the accuracy will be decreased by adding any new indirect WordNet resource. All of above facts imply that the indirect WordNet semantic resources except hyponymy→gloss may contain much more noise terms than the direct resources, especially the second kind of noises. Therefore, in order to avoid introducing too many noises, we chose the combination of "six resources" to construct WSP, and then disambiguate the ConceptNet for extending WordNet purpose.

Resources combination	**Accuracy**
Hyponymy→gloss	68.2%
Hyponymy→gloss + gloss	73.4%
Hyponymy→gloss + gloss + hyponymy	76.2%
Hyponymy→gloss + gloss + hyponymy+ Hypernymy	78.1%
Hyponymy→gloss + gloss + hyponymy+ hypernymy+ meronymy/holonymy	80.7%
Hyponymy→gloss + gloss + hyponymy+ hypernymy+ meronymy/holonymy+ synonymy	**82.4%**
Six resources+ hyponymy→hyponymy	80.5%
Six resources+ hypernymy→hypernymy	80.2%
Six resources+ hypernymy→gloss	79.8%
Six resources+ hypernymy→hyponymy	73.4%

Table2. The highest accuracies of disambiguating ConceptNet with different size of WordNet resources

4.2 Evaluation of WSD Methods

After disambiguated, the ConceptNet can be used to extend WordNet for WSD. In order to evaluate the impact of combining disambiguated ConceptNet and WordNet, we performed the comparative experiments on the Semeval-2007 coarse-grained all-words WSD dataset (Navigli et al., 2007) . We have chosen coarse-grained word sense disambiguation because the meanings of the ambiguous concepts in the ConceptNet assertions are naturally coarser than those in WordNet are. For example, given a ConceptNet assertion <*rain, isA, water*>, assigning either the first sense ("water falling in drops from vapor condensed in the atmosphere") or the second sense ("drops of fresh water that fall as precipitation from clouds") in WordNet to *rain* is suitable.

Since the aim of our experiment is to evaluate the impact of extended knowledge resource on WSD performance, the WSD algorithm is not the core of our work. Anyway, we implemented a simple knowledge-based algorithm, namely GM (Galley&McKeown, 2003), and our extending version. GM algorithm processes text sequentially, and compares current word to all of the previous words. If one of the senses of the current word has a semantic relation (synonymy, hypernym, hyponym, hypernymy→hyponymy) with any senses of previous words, then there is a weighted semantic edge between these two senses. After processing the whole text, a disambiguated graph is built, whose nodes represent the word senses and edges stand for the four kinds of semantic relations. Finally, for each sense of a target word, all scores of the edges linked to it are summed up as its score. The sense with the highest score is chosen as the correct sense for the target word.

In addition, we simply extend GM algorithm by considering more semantic relations. The extended algorithm is called as ExtGM. In details, in the semantic network of WordNet, if the length of the shortest path between two senses is not greater than four, we also consider that there exists a semantic relation between them, and assign the weight of this semantic relation as the inverse of the length. Since we did not focus on the WSD algorithms, the values of the two parameters (length, weight) of the implemented algorithms were not optimized.

Resource	**Method**	**P**	**R**	**F_1**
WordNet	GM	86.9	55.0	67.4
	ExtGM	**87.4**	70.6	78.0
WordNet + ConceptNet	GM	83.7	73.6	78.3
	ExtGM	85.5	**79.9**	**82.6**
WordNet + Wikipedia	Degree	87.3	72.7	79.4
	MFS BL	77.4	77.4	77.4
	Random BL	63.5	63.5	63.5

Table3. Performance on Semeval-2007 coarse-grained all words WSD (nouns only subset)

Table 3 shows the evaluation results of GM and ExtGM on Semeval-2007 coarse-grained all-words dataset, by using different knowledge resources: WordNet, WordNet+CocneptNet, where

P, R, and F_1 represent precision, recall and F1-measure respectively. We also use the random chosen sense (Random BL) and the most frequent sense (MFS BL) as baselines.

From this table, we can see that enriching WordNet with semantic relations from ConceptNet yields a significantly improvement against only using WordNet.

We also listed the result of Degree on WordNet+Wikipedia (Ponzetto&Navigli, 2010) in Table 3, from which we can see that compared to Degree, our method attains a slight variation in precision, but a significantly high increase in recall. The result shows that ConceptNet can increase recall for WSD more effectively than Wikipedia, though the size of Wikipedia is larger than that of ConceptNet. The reason may be that ConceptNet focuses on basic, unspoken knowledge which is obvious or common sense, therefore knowledge in ConceptNet may be more frequently used than those in Wikipedia.

Resource	**Method**	**P**	**R**	**F_1**
ConceptNet	ExtGM	**85.4**	**46.4**	**60.1**
ConceptNet(MFS)	ExtGM	80.9	43.4	56.5
	MFS BL	77.4	77.4	77.4

Table4. Performance on Semeval-2007 coarse-grained all words WSD (nouns only subset, and ConceptNet Only)

We further evaluate ExtGM on the two different ConceptNet: ConceptNet disambiguated by our method; ConceptNet disambiguated by MFS strategy, which assigns the most frequent sense to each ambiguous ConceptNet concept. The results are shown in Table 4, which illustrates that our method can attain significantly high increase in precision and recall. This proves that our ConceptNet disambiguating method is effective.

Algorithm	**Nouns only F_1**	**All word F_1**
ExtGM	**84.1**	**82.8**
SUSSX-FR	81.1	77.0
NUS-PT	82.3	**83.2**
MFS BL	77.4	78.9
Random BL	63.5	62.7

Table5. Performance on Semeval-2007 coarse-grained all-words WSD with MFS as a back-off strategy when no sense is assigned

Finally we compare the ExtGM with WordNet+ConceptNet to state-of-the-art WSD systems: SUSSX-FR (Koeling&McCarthy, 2007) and NUS-PT (Chan et al., 2007), which are the best unsupervised and supervised WSD systems participating in the Semeval-2007 coarse-grained all-words WSD task, respectively. Since the Semeval-2007 organizers allowed the algorithms to use the MFS as a back-off strategy when they did not return an answer, we apply this rule to ExtGM. Table 5 shows the results for nouns (1,108) and all words (2,269). The performance of ExtGM with WordNet+ConceptNet is significantly better than the best unsupervised system, and is not statistically different from the best supervised system NUS-PT. The result shows that enriching WordNet with the disambiguated ConceptNet can effectively improve the performance of knowledge-based WSD algorithms. In addition, using such enriched WordNet, even a simple knowledge-based algorithm can achieve state-of-the-art performance.

5 Conclusions

In this paper, we first proposed a novel method for the automatic disambiguation of a large-scale common sense knowledge base, namely ConceptNet. Then we used the disambiguated ConceptNet to enrichment WordNet. Our experiments show that enriching WordNet with the disambiguated ConceptNet can significantly improves the performance of knowledge-based WSD methods. On one hand, even a simple knowledge-based WSD algorithm using the enriched WordNet can perform as well as the highest-performing supervised ones. On the other hand, more sophisticated approaches (Agirre&Soroa, 2009; Navigli&Lapata, 2010) may achieve even higher performance by using such enriched WordNet. Moreover, the proposed ConceptNet disambiguating method can be easily applied for other knowledge resources to improve their quality too. We notice that ConceptNet is a multilingual common sense knowledge base, while we only concentrate on English Word Sense Disambiguation in this paper. It would be interesting to explore the impact of this knowledge resource in a multilingual setting.

Acknowledgements

The work was partially supported by the National Natural Science Foundation of China (60970063, 60773010), the Ph.D. Programs Foundation of Ministry of Education of China (20090141110026), and the Fundamental Research Funds for the Central Universities (6081007).

References

Agirre, E., O. Ansa, et al. 2000. Enriching very large ontologies using the WWW. Proceedings of the ECAI 2000 workshop "Ontology Learning".

Agirre, E. and A. Soroa 2009. Personalizing PageRank for word sense disambiguation. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Athens, Greece, Association for Computational Linguistics**:** 33-41.

Carletta, J. 1996. "Assessing agreement on classification tasks: the kappa statistic." Comput. Linguist. **22**(2): 249-254.

Caseli, H. d. M., B. A. Sugiyama, et al. 2010. Using common sense to generate culturally contextualized machine translation. Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas. Los Angeles, California, Association for Computational Linguistics**:** 24-31.

Chan, Y. S., H. T. Ng, et al. 2007. NUS-PT: exploiting parallel texts for word sense disambiguation in the English all-words tasks. Proceedings of the 4th International Workshop on Semantic Evaluations. Prague, Czech Republic, Association for Computational Linguistics**:** 253-256.

Cilibrasi, R.L., et al. 2007. "The Google Similarity Distance." IEEE Transactions on Knowledge and Data Engineering **19**(3): 370-383.

Cuadros, M. and G. Rigau 2008. KnowNet: building a large net of knowledge from the web. Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1. Manchester, United Kingdom, Association for Computational Linguistics**:** 161-168.

Fellbaum, C., Ed. 1998. WordNet. An electronic lexical database, MIT Press.

Galley, M. and K. McKeown 2003. Improving Word Sense Disambiguation in Lexical Chaining. In Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico**:** 1486--1488.

Havasi, C. and R. S. a. J. Alonso 2007. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In Proceedings of Recent Advances in Natural Languge Processing 2007.

Hwang, M., C. Choi, et al. 2011 "Automatic Enrichment of Semantic Relation Network and Its Application to Word Sense Disambiguation." IEEE Transactions on Knowledge and Data Engineering **23**(6): 845 - 858

Koeling, R. and D. McCarthy 2007. Sussx: WSD using automatically acquired predominant senses. Proceedings of the 4th International Workshop on Semantic Evaluations. Prague, Czech Republic, Association for Computational Linguistics**:** 314-317.

Lieberman, H., A. Faaborg, et al. 2005. How to wreck a nice beach you sing calm incense. Proceedings of the 10th international conference on Intelligent user interfaces. San Diego, California, USA, ACM**:** 278-280.

Magnini, B. and G. Cavagli à 2000. Integrating subject field codes into WordNet In Proceedings of the second International Conference on Language Resources and Evaluation 1413--1418.

Mihalcea, R. 2007. Using Wikipedia for automatic Word Sense Disambiguation. NAACLHLT-07**:** 196-203.

Mihalcea, R. and D. I. Moldovan 2001. eXtended WordNet: progress report in Proceedings of NAACL Workshop on WordNet and Other Lexical Resources 95--100.

Navigli, R. 2009. Using cycles and quasi-cycles to disambiguate dictionary glosses. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Athens, Greece, Association for Computational Linguistics**:** 594-602.

Navigli, R. and M. Lapata 2010. "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation." IEEE Trans. Pattern Anal. Mach. Intell. **32**(4): 678-692.

Navigli, R., K. C. Litkowski, et al. 2007. SemEval-2007 task 07: coarse-grained English all-words task. Proceedings of the 4th International Workshop on Semantic Evaluations. Prague, Czech Republic, Association for Computational Linguistics**:** 30-35.

Ponzetto, S. P. and R. Navigli 2010. Knowledge-rich Word Sense Disambiguation rivaling supervised systems. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, Association for Computational Linguistics**:** 1522-1531.

Speer, R., J. Krishnamurthy, et al. 2009. An interface for targeted collection of common sense knowledge using a mixture model. Proceedings of the 14th international conference on Intelligent user interfaces. Sanibel Island, Florida, USA, ACM**:** 137-146.