SENSEVAL-3: Third International Workshop on the Evaluation of Systems
for the Semantic Analysis of Text, Barcelona, Spain, July 2004
Association for Computational Linguistics

Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities

Cheng Niu, Wei Li, Rohini K. Srihari, Huifeng Li, Laurie Crist
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221. USA.
{cniu, wei, rohini, hli, lcrist}@cymfony.com

Abstract

Traditionally, word sense disambiguation (WSD) involves a different context model for each individual word. This paper presents a new approach to WSD using weakly supervised learning. Statistical models are not trained for the contexts of each individual word, but for the similarities between context pairs at category level. The insight is that the correlation regularity between the sense distinction and the context distinction can be captured at category level, independent of individual words. This approach only requires a limited amount of existing annotated training corpus in order to disambiguate the entire vocabulary. A context clustering scheme is developed within the Bayesian framework. A maximum entropy model is then trained to represent the generative probability distribution of context similarities based on heterogeneous features, including trigger words and parsing structures. Statistical annealing is applied to derive the final context clusters by globally fitting the pairwise context similarity distribution. Benchmarking shows that this new approach significantly outperforms the existing WSD systems in the unsupervised category, and rivals supervised WSD systems.

1 Introduction

Word Sense Disambiguation (WSD) is one of the central problems in Natural Language Processing. The difficulty of this task lies in the fact that context features and the corresponding statistical distribution are different for each individual word. Traditionally, WSD involves modeling the contexts for each word. [Gale et al. 1992] uses the Naïve Bayes method for context modeling which requires a manually truthed corpus for each ambiguous word. This causes a serious *Knowledge Bottleneck*. The situation is worse when considering the domain dependency of word senses. To avoid the Knowledge Bottleneck, unsupervised or weakly supervised learning approaches have been proposed. These include the bootstrapping approach [Yarowsky 1995] and the context clustering approach [Schutze 1998].

Although the above unsupervised or weakly supervised learning approaches are less subject to the Knowledge Bottleneck, some weakness exists: i) for each individual keyword, the sense number has to be provided and in the bootstrapping case, seeds for each sense are also required; ii) the modeling usually assumes some form of evidence independency, e.g. the vector space model used in [Schutze 1998] and [Niu et al. 2003]: this limits the performance and its potential enhancement; iii) most WSD systems either use selectional restriction in parsing relations, and/or trigger words which co-occur within a window size of the ambiguous word. We previously at-tempted combining both types of evidence but only achieved limited improvement due to the lack of a proper modeling of information over-lapping [Niu et al. 2003].

This paper presents a new algorithm that addresses these problems. A novel context clustering scheme based on modeling the similarities between pairwise contexts at category level is presented in the Bayesian framework. A generative maximum entropy model is then trained to represent the generative probability distribution of pairwise context similarities based on heterogeneous features that cover both co-occurring words and parsing structures. Statistical annealing is used to derive the final context clusters by globally fitting the pairwise context similarities.

This new algorithm only requires a limited amount of existing annotated corpus to train the generative maximum entropy model for the entire vocabulary. This capability is based on the observation that a system does not necessarily require training data for word *A* in order to disambiguate *A*. The insight is that the correlation regularity between the sense distinction and the context distinction can be captured at category level, independent of individual words.

In what follows, Section 2 formulates WSD as a context clustering task based on the pairwise

context similarity model. The context clustering algorithm is described in Sections 3 and 4, corresponding to the two key aspects of the algorithm, i.e. the generative maximum entropy modeling and the annealing-based optimization. Section 5 describes benchmarks and conclusion.

2 Task Definition and Algorithm Design

Given n mentions of a key word, we first introduce the following symbols. C_i refers to the i-th context. S_i refers to the sense of the i-th context. $CS_{i,j}$ refers to the context similarity between the i-th context and the j-th context, which is a subset of the predefined context similarity features. f_α refers to the α-th predefined context similarity feature. So $CS_{i,j}$ takes the form of $\{f_\alpha\}$.

The WSD task is defined as the hard clustering of multiple contexts of the key word. Its final solution is represented as $\{K, M\}$ where K refers to the number of distinct senses, and M represents the many-to-one mapping (from contexts to a cluster) such that $\mathrm{M(i)} = \mathrm{j}, \mathrm{i} \in [1, \mathrm{n}], \mathrm{j} \in [1, \mathrm{K}]$.

For any given context pair, a set of context similarity features are defined. With n mentions of the same key word, $\frac{n(n-1)}{2}$ context similarities $CS_{i,j}$ $(i \in [1,n],\ j \in [1,i))$ are computed. The WSD task is formulated as searching for $\{K, M\}$ which maximizes the following conditional probability:

$$\Pr(\{K, M\} | \{CS_{i,j}\}) \quad (i \in [1,n],\ j \in [1,i))$$

Based on Bayesian Equity, this is equivalent to maximizing the joint probability in Eq. (1), which contains a prior probability distribution of WSD, $\Pr(\{K, M\})$.

$$\begin{aligned} &\Pr(\{K, M\}, \{CS_{i,j}\}) \quad (i \in [1,n],\ j \in [1,i)) \\ &= \Pr(\{CS_{i,j}\} | \{K, M\}) \Pr(\{K, M\}) \qquad (1) \\ &= \prod_{\substack{i=1,N \\ j=1,i-1}} \Pr(CS_{i,j} | \{K, M\}) \Pr(\{K, M\}) \end{aligned}$$

Because there is no prior knowledge available about what solution is preferred, it is reasonable to take an equal distribution as the prior probability distribution. So WSD is equivalent to searching for $\{K, M\}$ which maximizes Expression (2).

$$\prod_{\substack{i=1,N \\ j=1,i-1}} \Pr(CS_{i,j} | \{K, M\}) \qquad (2)$$

where

$$\Pr(CS_{i,j} | \{K, M\}) = \begin{cases} \Pr(CS_{i,j} | S_i = S_j), \text{ if } \mathrm{M(i)} = \mathrm{M(j)} \\ \Pr(CS_{i,j} | S_i \neq S_j), \text{ otherwise} \end{cases} \qquad (3)$$

To learn the conditional probabilities $\Pr(CS_{i,j} | S_i = S_j)$ and $\Pr(CS_{i,j} | S_i \neq S_j)$ in Eq. (3), a maximum entropy model is trained. There are two major advantages of this maximum entropy model: i) the model is independent of individual words; ii) the model takes no *information independence* assumption about the data, and hence is powerful enough to utilize heterogeneous features. With the learned conditional probabilities in Eq. (3), for a given $\{K, M\}$ candidate, we can compute the conditional probability of Expression (2). In the final step, optimization is performed to search for $\{K, M\}$ that maximizes the value of Expression (2).

3 Maximum Entropy Modeling

This section presents the definition of context similarity features, and how to estimate the generative probabilities of context similarity $\Pr(CS_{i,j} | S_i = S_j)$ and $\Pr(CS_{i,j} | S_i \neq S_j)$ using maximum entropy modeling.

Using the Senseval-2 training corpus,[1] we have constructed Corpus I and Corpus II for each Part-of-speech (POS) tag. Corpus I is constructed using context pairs involving the same sense of a word. Corpus II is constructed using context pairs that refer to different senses of a word. Each corpus contains about 18,000 context pairs. The instances in the corpora are represented as pairwise context similarities, taking the form of $\{f_\alpha\}$. The two conditional probabilities $\Pr(CS_{i,j} | S_i = S_j)$ and $\Pr(CS_{i,j} | S_i \neq S_j)$ can be represented as $\Pr_{\mathrm{I}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ and $\Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ which are generative probabilities by maximum entropy for Corpus I and Corpus II.

We now present how to compute the context similarities. Each context contains the following two categories of features:

i) Trigger words centering around the key word within a predefined window size equal to 50 tokens to both sides of the key word. Trigger words are learned using the same technique as in [Niu et al. 2003].

ii) Parsing relationships associated with the key word automatically decoded by our parser

[1] Note that the words that appear in the Senseval-3 lexical sample evaluation are removed in the corpus construction process.

InfoXtract [Srihari et al. 2003]. The relationships being utilized are listed below.

Noun: *subject-of, object-of, complement-of, has-adjective-modifier, has-noun-modifier, modifier-of, possess, possessed-by, appositive-of*

Verb: *has-subject, has-object, has-complement, has-adverb-modifier, has-prepositional-modifier*

Adjective: *modifier-of, has-adverb-modifier*

Based on the above context features, the following three categories of context similarity features are defined:

(1) Context similarity based on a vector space model using co-occurring trigger words: the trigger words centering around the key word are represented as a vector, and the *tf*idf* scheme is used to weigh each trigger word. The cosine of the angle between two resulting vectors is used as a context similarity measure.

(2) Context similarity based on Latent semantic analysis (LSA) using trigger words: LSA [Deerwester et al. 1990] is a technique used to uncover the underlying semantics based on co-occurrence data. Using LSA, each word is represented as a vector in the semantic space. The trigger words are represented as a vector summation. Then the cosine of the angle between the two resulting vector summations is computed, and used as a context similarity measure.

(3) LSA-based Parsing Structure Similarity: each relationship is in the form of $R_\alpha(w)$. Using LSA, each word w is represented as semantic vector $V(w)$. Then, the similarity between $R_\alpha(w_1)$ and $R_\alpha(w_2)$ is represented as the cosine of angle between $V(w_1)$ and $V(w_2)$. Two special values are assigned to two exceptional cases: i) when no relationship R_α is decoded in both contexts; ii) when the relationship R_α is decoded only for one context.

To facilitate the maximum entropy modeling in the later stage, the resulting similarity measure is discretized into 10 integer values. Now the pairwise context similarity is a set of similarity features, e.g.

{VSM-Similairty-equal-to-2, LSA-Trigger-Words-Similarity-equal-to-1, LSA-Subject-Similarity-equal-to-2}.

In addition to the three categories of basic context similarity features defined above, we also define induced context similarity features by combining basic context similarity features using the logical *AND* operator. With induced features, the context similarity vector in the previous example is represented as

{VSM-Similairty-equal-to-2, LSA- Trigger-Words-Similarity-equal-to-1, LSA-Subject-Similarity-equal-to-2,
[VSM-Similairty-equal-to-2 *and* LSA-Trigger - Words-Similarity-equal-to-1], [VSM-Similairty-equal-to-2 *and* LSA-Subject-Similarity-equal-to-2],
……,
[VSM-Similairty-equal-to-2 *and* LSA-Trigger - Words-Similarity-equal-to-1 *and* LSA-Subject-Similarity-equal-to-2]}.

The induced features provide direct and fine-grained information, but suffer from less sampling space. To make the computation feasible, we regulate 3 as the maximum number of logical *AND* in the induced features. Combining basic features and induced features under a smoothing scheme, maximum entropy modeling may achieve optimal performance.

Now the maximum entropy modeling can be formulated as follows: given a pairwise context similarity $\{f_\alpha\}$, the generative probability of $\{f_\alpha\}$ in Corpus I or Corpus II is given as

$$\Pr^{\text{maxEnt}}(\{f_\alpha\}) = \frac{1}{Z}\prod_{f\in\{f_\alpha\}} w_f \qquad (4)$$

where Z is the normalization factor, w_f is the weight associated with feature f. The Iterative Scaling algorithm combined with Monte Carlo simulation [Pietra, Pietra, & Lafferty 1995] is used to train the weights in this generative model. Unlike the commonly used conditional maximum entropy modeling which approximates the feature configuration space as the training corpus [Ratnaparkhi 1998], Monte Carlo techniques are required in the generative modeling to simulate the possible feature configurations. The exponential prior smoothing scheme [Goodman 2003] is adopted. The same training procedure is performed using Corpus I and Corpus II to estimate $\Pr_{\text{I}}^{\text{maxEnt}}(\{f_i\})$ and $\Pr_{\text{II}}^{\text{maxEnt}}(\{f_i\})$ respectively.

4 Statistical Annealing

With the maximum entropy modeling presented above, the WSD task is performed as follows: i) for a given set of contexts, the pairwise context similarity measures are computed; ii) for each context similarity $\{f_i\}$, the two generative probabilities $\Pr_{\mathrm{I}}^{\max\mathrm{Ent}}(\{f_i\})$ and $\Pr_{\mathrm{II}}^{\max\mathrm{Ent}}(\{f_i\})$ are computed; iii) for a given WSD candidate solution $\{K, M\}$, the conditional probability (2) can be computed. Optimization based on statistical annealing (Neal 1993) is used to search for $\{K, M\}$ which maximizes Expression (2).

The optimization process consists of two steps. First, a local optimal solution $\{K, M\}_0$ is computed by a greedy algorithm. Then by setting $\{K, M\}_0$ as the initial state, statistical annealing is applied to search for the global optimal solution. To reduce the search time, we set the maximum value of K to 5.

5 Benchmarking and Conclusion

To enter the Senseval-3 evaluation, we implemented the following procedure to map the context clusters to Senseval-3 standards: i) process the Senseval-3 training corpus and testing corpus using our parser; ii) for each word to be benchmarked, retrieve the related contexts from the corpora and cluster them; iii) Based on 10% of the sense tags in the Senseval-3 training corpus (10% data correspond roughly to an average of 2-3 instances for each sense), the context cluster is mapped onto the most frequent WSD sense associated with the cluster members. By design, the context clusters correspond to distinct senses, therefore, we do not allow multiple context clusters to be mapped onto one sense. In case multiple clusters correspond to one sense, only the largest cluster is retained; iv), each instance in the testing corpus is tagged with the same sense as the one to which its context cluster corresponds.

We are not able to compare our performance with other systems in Senseval-3 because at the time of writing, the Senseval-3 evaluation results are not publicly available. As a note, compared with the Senseval-2 English Lexical Sample evaluation, the benchmarks of our new algorithm (Table 1) are significantly above the performance of the WSD systems in the unsupervised category, and rival the performance of the supervised WSD systems.

Table 1. Senseval-3 Lexical Sample Evaluation

Category	Accuracy	
	Fine grain (%)	Coarse grain (%)
Adjective (5)	49.1	64.8
Noun (20)	57.9	66.6
Verb (32)	55.3	66.3
Average	56.3%	66.4%

6 Acknowledgements

This work was supported by the Navy SBIR program under contract N00178-03-C-1047.

References

Gale, W., K. Church, and D. Yarowsky. 1992. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of ACL 1995.*

Schutze, H. 1998. Automatic Word Sense Disambiguation. *Computational Linguistics*, 23.

C. Niu, Zhaohui Zheng, R. Srihari, H. Li, and W. Li 2003. Unsupervised Learning for Verb Sense Disambiguation Using Both trigger Words and Parsing Relations. In Proceeding of PACLING 2003, Halifax, Canada.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. In *Journal of the American Society of Information Science*

Goodman, J. 2003. Exponential Priors for Maximum Entropy Models.

Neal, R.M. 1993. Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical Report, Univ. of Toronto.

Pietra, S. D., V. D. Pietra, and J. Lafferty. 1995. Inducing Features Of Random Fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Adwait Ratnaparkhi. (1998). Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. Dissertation. University of Pennsylvania.

Srihari, R., W. Li, C. Niu and T. Cornell. 2003. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. In *Proceedings of HLT/NAACL 2003 Workshop on SEALTS*. Edmonton, Canada.