Word Independent Context Pair Classification Model for Word Sense Disambiguation

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

Abstract

Traditionally, word sense disambiguation (WSD) involves a different context classification model for each individual word. This paper presents a weakly supervised learning approach to WSD based on learning a word independent context pair classification model. Statistical models are not trained for classifying the word contexts, but for classifying a pair of contexts, i.e. determining if a pair of contexts of the same ambiguous word refers to the same or different senses. Using this approach, annotated corpus of a target word *A* can be explored to disambiguate senses of a different word *B*. Hence, only a limited amount of existing annotated corpus is required in order to disambiguate the entire vocabulary. In this research, maximum entropy modeling is used to train the word independent context pair classification model. Then based on the context pair classification results, clustering is performed on word mentions extracted from a large raw corpus. The resulting context clusters are mapped onto the external thesaurus WordNet. This approach shows great flexibility to efficiently integrate heterogeneous knowledge sources, e.g. trigger words and parsing structures. Based on Senseval-3 Lexical Sample standards, this approach achieves state-of-the-art performance in the unsupervised learning category, and performs comparably with the supervised Naïve Bayes system.

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 training the context classification models for each ambiguous word. (Gale et al. 1992) uses the Naïve Bayes method for context classification which requires a manually annotated corpus for each ambiguous word. This causes a serious *Knowledge Bottleneck*. The bottleneck is particularly serious when considering the domain dependency of word senses. To overcome the *Knowledge Bottleneck*, unsupervised or weakly supervised learning approaches have been proposed. These include the bootstrapping approach (Yarowsky 1995) and the context clustering approach (Schütze 1998).

The above unsupervised or weakly supervised learning approaches are less subject to the *Knowledge Bottleneck*. For example, (Yarowsky 1995) only requires sense number and a few seeds for each sense of an ambiguous word (hereafter called *keyword*). (Schütze 1998) may only need minimal annotation to map the resulting context clusters onto external thesaurus for benchmarking and application-related purposes. Both methods are based on trigger words only.

This paper presents a novel approach based on learning word-independent context pair classification model. This idea may be traced back to (Schütze 1998) where context clusters based on generic Euclidean distance are regarded as distinct word senses. Different from (Schütze 1998), we observe that generic context clusters may not always correspond to distinct word senses. Therefore, we used supervised machine learning to model the relationships between the context distinctness and the sense distinctness.

Although supervised machine learning is used for the context pair classification model, our overall system belongs to the weakly supervised category because the learned context pair classification

Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL),
pages 33–39, Ann Arbor, June 2005.

model is independent of the keyword for disambiguation. Our system does not need human-annotated instances for each target ambiguous word. The weak supervision is performed by using a limited amount of existing annotated corpus which does not need to include the target word set.

The insight is that the correlation regularity between the sense distinction and the context distinction can be captured at Part-of-Speech category level, independent of individual words or word senses. Since context determines the sense of a word, a reasonable hypothesis is that there is some mechanism in the human comprehension process that will decide when two contexts are similar (or dissimilar) enough to trigger our interpretation of a word in the contexts as one meaning (or as two different meanings). We can model this mechanism by capturing the sense distinction regularity at category level.

In the light of this, a maximum entropy model is trained to determine if a pair of contexts of the same keyword refers to the same or different word senses. The maximum entropy modeling is based on heterogeneous context features that involve both trigger words and parsing structures. To ensure the resulting model's independency of individual words, the keywords used in training are different from the keywords used in benchmarking. For any target keyword, a collection of contexts is retrieved from a large raw document pool. Context clustering is performed to derive the optimal context clusters which globally fit the local context pair classification results. Here statistical annealing is used for its optimal performance. In benchmarking, a mapping procedure is required to correlate the context clusters with external ontology senses.

In what follows, Section 2 formulates the maximum entropy model for context pair classification. The context clustering algorithm, including the object function of the clustering and the statistical annealing-based optimization, is described in Section 3. Section 4 presents and discusses benchmarks, followed by conclusion in Section 5.

2 Maximum Entropy Modeling for Context Pair Classification

Given n mentions of a keyword, 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 $\{f_\alpha\}$.

In this section, we study the context pair classification task, i.e. given a pair of contexts C_i and C_j of the same target word, are they referring to the same sense? This task is formulated as comparing the following conditional probabilities: $\Pr(S_i = S_j | CS_{i,j})$ and $\Pr(S_i \neq S_j | CS_{i,j})$. Unlike traditional context classification for WSD where statistical model is trained for each individual word, our context pair classification model is trained for each Part-of-speech (POS) category. The reason for choosing POS as the appropriate category for learning the context similarity is that the parsing structures, hence the context representation, are different for different POS categories.

The training corpora are constructed using the Senseval-2 English Lexical Sample training corpus. To ensure the resulting model's independency of individual words, the target words used for benchmarking (which will be the ambiguous words used in Senseval-3 English Lexicon Sample task) are carefully removed in the corpus construction process. For each POS category, positive and negative instances are constructed as follows.

Positive instances are constructed using context pairs referring to the same sense of a word. Negative instances are constructed using context pairs that refer to different senses of a word.

For each POS category, we have constructed about 36,000 instances, half positive and half negative. The instances are represented as pairwise context similarities, taking the form of $\{f_\alpha\}$.

Before presenting the context similarity features we used, we first introduce the two categories of the involved context features:

i) Co-occurring trigger words within a predefined window size equal to 50 words to both sides of the keyword. The trigger words are learned from a TIPSTER document pool containing ~170 million words of AP and WSJ news articles. Following (Schütze 1998), χ^2 is used to measure the cohesion between the keyword and a co-occurring word. In our ex-

periment, all the words are first sorted based on its χ^2 with the keyword, and then the top 2,000 words are selected as trigger words.

ii) Parsing relationships associated with the keyword automatically decoded by a broad-coverage parser, with F-measure (i.e. the precision-recall combined score) at about 85% (reference temporarily omitted for the sake of blind review). The logical dependency 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-phrase-modifier

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

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

(1) VSM-based (Vector Space Model based) trigger word similarity: the trigger words around the keyword are represented as a vector, and the word *i* in context *j* is weighted as follows:

$$weight(i, j) = tf(i, j) * \log \frac{D}{df(i)}$$

where $tf(i, j)$ is the frequency of word *i* in the *j-th* context; D is the number of documents in the pool; and $df(i)$ is the number of documents containing the word *i*. D and $df(i)$ are estimated using the document pool introduced above. The cosine of the angle between two resulting vectors is used as the context similarity measure.

(2) LSA-based (Latent Semantic Analysis based) trigger word similarity: LSA (Deerwester et al. 1990) is a technique used to uncover the underlying semantics based on co-occurrence data. The first step of LSA is to construct word-vs.-document co-occurrence matrix. Then singular value decomposition (SVD) is performed on this co-occurring matrix. The key idea of LSA is to reduce noise or insignificant association patterns by filtering the insignificant components uncovered by SVD. This is done by keeping only the top *k* singular values. By using the resulting word-vs.-document co-occurrence matrix after the filtering, each word can be represented as a vector in the semantic space.

In our experiment, we constructed the original word-vs.-document co-occurring matrix as follows: 100,000 documents from the TIPSTER corpus were used to construct the co-occurring matrix. We processed these documents using our POS tagger, and selected the top *n* most frequently mentioned words from each POS category as base words:

top 20,000 common nouns
top 40,000 proper names
top 10,000 verbs
top 10,000 adjectives
top 2,000 adverbs

In performing SVD, we set *k* (i.e. the number of nonzero singular values) as 200, following the practice reported in (Deerwester et al. 1990) and (Landauer & Dumais, 1997).

Using the LSA scheme described above, each word is represented as a vector in the semantic space. The co-occurring 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 the context similarity measure.

(3) LSA-based parsing relationship similarity: each relationship is in the form of $R_\alpha(w)$. Using LSA, each word *w* is represented as a

semantic vector $V(w)$. The similarity between $R_\alpha(w_1)$ and $R_\alpha(w_2)$ is represented as the cosine of the 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.

In matching parsing relationships in a context pair, if only exact node match counts, very few cases can be covered, hence significantly reducing the effect of the parser in this task. To solve this problem, LSA is used as a type of synonym expansion in matching. For example, using LSA, the following word similarity values are generated:

similarity(good, good) 1.00
similarity(good, pretty) 0.79
similarity(good, great) 0.72
……

Given a context pair of a noun keyword, suppose the first context involves a relationship *has-adjective-modifier* whose value is *good,* and the second context involves the same relationship *has-adjective-modifier* with the value *pretty,* then the system assigns 0.79 as the similarity value for this relationship pair.

To facilitate the maximum entropy modeling in the later stage, all the three categories of the resulting similarity values are discretized into 10 integers. Now the pairwise context similarity is represented as a set of similarity features, e.g.

{VSM-Trigger-Words-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-Trigger-Word-Similairty-equal-to-2,
LSA- Trigger-Word-Similarity-equal-to-1,
LSA-Subject-Similarity-equal-to-2,
[VSM-Similairty-equal-to-2 *and*
LSA-Trigger-Word-Similarity-equal-to-1],
[VSM-Similairty-equal-to-2 *and*
LSA-Subject-Similarity-equal-to-2],
……
[VSM-Trigger-Word-Similairty-equal-to-2
and LSA-Trigger-Word-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. Combining basic features and induced features under a smoothing scheme, maximum entropy modeling may achieve optimal performance.

Using the context similarity features defined above, the training corpora for the context pair classification model is in the following format:

Instance_0 tag="positive" {VSM-Trigger-Word-Similairty-equal-to-2, …}
Instance_1 tag="negative" {VSM-Trigger-Word-Similairty-equal-to-0, …}
……
where *positive* tag denotes a context pair associated with same sense, and *negative* tag denotes a context pair associated with different senses.

The maximum entropy modeling is used to compute the conditional probabilities $\Pr(S_i = S_j | CS_{i,j})$ and $\Pr(S_i \neq S_j | CS_{i,j})$: once the context pair $CS_{i,j}$ is represented as $\{f_\alpha\}$, the conditional probability is given as

$$\Pr(t|\{f_\alpha\}) = \frac{1}{Z}\prod_{f\in\{f_\alpha\}} w_{t,f} \qquad (1)$$

where $t \in \{S_i = S_j, S_i \neq S_j\}$, Z is the normalization factor, $w_{t,f}$ is the weight associated with tag t and feature f. Using the training corpora constructed above, the weights can be computed based on Iterative Scaling algorithm (Pietra etc. 1995) The exponential prior smoothing scheme (Goodman 2003) is adopted in the training.

3 Context Clustering based on Context Pair Classification Results

Given n mentions $\{C_i\}$ of a keyword, we use the following context clustering scheme. The discovered context clusters correspond to distinct word senses.

For any given context pair, the context similarity features defined in Section 2 are computed. With n mentions of the same keyword, $\frac{n(n-1)}{2}$ context similarities $CS_{i,j}\ (i \in [1,n], j \in [1,i))$ are computed. Using the context pair classification model, each pair is associated with two scores $sc^0_{i,j} = \log(\Pr(S_i = S_j | CS_{i,j}))$ and $sc^1_{i,j} = \log(\Pr(S_i = S_j | CS_{i,j}))$ which correspond to the probabilities of two situations: the pair refers to the same or different word senses.

Now we introduce the symbol $\{K, M\}$ which refers to the final context cluster configuration, where K refers to the number of distinct sense, and M represents the many-to-one mapping (from contexts to a sense) such that $\mathrm{M(i)} = \mathrm{j}, \mathrm{i} \in [1,\mathrm{n}], \mathrm{j} \in [1,\mathrm{K}]$. Based on the pairwise scores $\{sc^0_{i,j}\}$ and $\{sc^1_{i,j}\}$, WSD is formulated as searching for $\{K, M\}$ which maximizes the following global scores:

$$sc(\{\mathrm{K},\mathrm{M}\}) = \sum_{\substack{i\in[1,n],\\ j\in[1,i)}} sc_{i,j}^{k(i,j)} \quad (2)$$

where $k(i,j) = \begin{cases} 0, if\ M(i) = M(j) \\ 1, \quad otherwise \end{cases}$

Similar clustering scheme has been used successfully for the task of co-reference in (Luo etc. 2004), (Zelenko, Aone and Tibbetts, 2004a) and (Zelenko, Aone and Tibbetts, 2004b).

In this paper, statistical annealing-based optimization (Neal 1993) is used to search for $\{K, M\}$ which maximizes Expression (2).

The optimization process consists of two steps. First, an intermediate 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. The optimization algorithm is as follows.

1. Set the initial state $\{K, M\}$ as $K = n$, and $M(i) = i,\ \mathrm{i} \in [1,\mathrm{n}]$;
2. Select a cluster pair for merging that maximally increases $sc(\{\mathrm{K},\mathrm{M}\}) = \sum_{\substack{i\in[1,n],\\ j\in[1,i)}} sc_{i,j}^{k(i,j)}$
3. If no cluster pair can be merged to increase $sc(\{\mathrm{K},\mathrm{M}\}) = \sum_{\substack{i\in[1,n],\\ j\in[1,i)}} sc_{i,j}^{k(i,j)}$, output $\{K, M\}$ as the intermediate solution; otherwise, update $\{K, M\}$ by the merge and go to step 2.

Using the intermediate solution $\{K, M\}_0$ of the greedy algorithm as the initial state, the statistical annealing is implemented using the following pseudo-code:

```
Set {K,M} = {K,M}_0 ;
for(β = β_0 ; β < β_final ; β* = 1.01)
{
  iterate pre-defined number of times
  {
    set {K,M}_1 = {K,M};
    update {K,M}_1 by randomly changing
  cluster number and cluster contents;
      set x = sc({K,M}_1) / sc({K,M})
      if(x>=1)
      {
        set {K,M} = {K,M}_1
      }
      else
      {
        set {K,M} = {K,M}_1 with probability
          x^β .
      }
      if sc({K,M}) > sc({K,M}_0)
      then set {K,M}_0 = {K,M}
  }
}
output {K,M}_0 as the optimal state.
```

4 Benchmarking

Corpus-driven context clusters need to map to a word sense standard to facilitate performance benchmark. Using Senseval-3 evaluation standards, we implemented the following procedure to map the context clusters:

i) Process TIPSTER corpus and the original unlabeled Senseval-3 corpora (including the training corpus and the testing corpus) by our parser, and save all the parsing results into a repository.

ii) For each keyword, all related contexts in Senseval-3 corpora and up-to-1,000 related contexts in TIPSTER corpus are retrieved from the repository.

iii) All the retrieved contexts are clustered based on the context clustering algorithm presented in Sect. 2 and 3.

iv) For each keyword sense, three annotated contexts from Senseval-3 training corpus are used for the sense mapping. The context cluster is mapped onto the most frequent word 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.

v) Each context in the testing corpus is tagged with the sense to which its context cluster corresponds to.

As mentioned above, Sensval-2 English lexical sample training corpora is used to train the context pair classification model. And Sensval-3 English lexical sample testing corpora is used here for benchmarking. There are several keyword occurring in both Senseval-2 and Senseval-3 corpora. The sense tags associated with these keywords are not used in the context pair classification training process.

In order to gauge the performance of this new weakly supervised learning algorithm, we have also implemented a supervised Naïve Bayes system following (Gale et al. 1992). This system is trained based on the Senseval-3 English Lexical Sample training corpus. In addition, for the purpose of quantifying the contribution from the parsing structures in WSD, we have run our new system with two configurations: (i) using only trigger words; (ii) using both trigger words and parsing relationships. All the benchmarking is performed using the Senseval-3 English Lexical Sample testing corpus and standards.

The performance benchmarks for the two systems in three runs are shown in Table 1, Table 2 and Table 3. When using only trigger words, this algorithm has 8 percentage degradation from the supervised Naïve Bayes system (see Table 1 vs. Table 2). When adding parsing structures, performance degradation is reduced, with about 5 percentage drop (see Table 3 vs. Table 2). Comparing Table 1 with Table 3, we observe about 3% enhancement due to the contribution from the parsing support in WSD. The benchmark of our algorithm using both trigger words and parsing relationships is one of the best in unsupervised category of the Senseval-3 Lexical Sample evaluation.

Table 1. New Algorithm Using Only Trigger Words

Category	Accuracy	
	Fine grain (%)	Coarse grain (%)
Adjective (5)	46.3	60.8
Noun (20)	54.6	62.8
Verb (32)	54.1	64.2
Overall	54.0	63.4

Table 2. Supervised Naïve Bayes System

Category	Accuracy	
	Fine grain (%)	Coarse grain (%)
Adjective (5)	44.7	56.6
Noun (20)	66.3	74.5
Verb (32)	58.6	70.0
Overall	61.6	71.5

Table 3. New Algorithm Using Both Trigger Words and Parsing

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

It is noted that Naïve Bayes algorithm has many variation, and its performance has been greatly enhanced during recent research. Based on Senseval-3 results, the best Naïve Bayse system outperform our version (which is implemented based on Gale et al. 1992) by 8%~10%. So the best supervised WSD systems output-perform our weakly supervised WSD system by 13%~15% in accuracy.

5 Conclusion

We have presented a weakly supervised learning approach to WSD. Statistical models are not trained for the contexts of each individual word, but for context pair classification. This approach overcomes the knowledge bottleneck that challenges supervised WSD systems which need labeled data for each individual word. It captures the correlation regularity between the sense distinction and the context distinction at Part-of-Speech category level, independent of individual words and senses. Hence, it only requires a limited amount of existing annotated corpus in order to disambiguate the full target set of ambiguous words, in particular, the target words that do not appear in the training corpus.

The weakly supervised learning scheme can combine trigger words and parsing structures in supporting WSD. Using Senseval-3 English Lexical Sample benchmarking, this new approach reaches one of the best scores in the unsupervised category of English Lexical Sample evaluation. This performance is close to the performance for the supervised Naïve Bayes system.

In the future, we will implement a new scheme to map context clusters onto WordNet senses by exploring WordNet glosses and sample sentences. Based on the new sense mapping scheme, we will benchmark our system performance using Senseval English all-words corpora.

References

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*

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

Goodman, J. 2003. Exponential Priors for Maximum Entropy Models. In *Proceedings of HLT-NAACL 2004*.

Landauer, T. K., & Dumais, S. T. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240, 1997.

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla and S. Roukos. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. In *The Proceedings of ACL 2004.*

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*.

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

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

Zelenko, D., C. Aone and J. 2004. Tibbetts. Coreference Resolution for Information Extraction. In *Proceedings of ACL 2004 Workshop on Reference Resolution and its Application*.

Zelenko, D., C. Aone and J. 2004. Tibbetts. Binary Integer Programming for Information Extraction. In *Proceedings of ACE 2004 Evaluation Workshop*.