Naive Bayes Word Sense Induction

Do Kook Choe
Brown University
Providence, RI
dc65@cs.brown.edu

Eugene Charniak
Brown University
Providence, RI
ec@cs.brown.edu

Abstract

We introduce an extended naive Bayes model for word sense induction (WSI) and apply it to a WSI task. The extended model incorporates the idea the words closer to the target word are more relevant in predicting its sense. The proposed model is very simple yet effective when evaluated on SemEval-2010 WSI data.

1 Introduction

The task of word sense induction (WSI) is to find clusters of tokens of an ambiguous word in an unlabeled corpus that have the same sense. For instance, given a target word "crane," a good WSI system should find a cluster of tokens referring to avian cranes and another referring to mechanical cranes. We believe that neighboring words contain enough information that these clusters can be found from plain texts.

WSI is related to word sense disambiguation (WSD). In a WSD task, a system learns a sense classifier in a supervised manner from a sense-labeled corpus. The performance of the learned classifier is measured on some unseen data. WSD systems perform better than WSI systems, but building labeled data can be prohibitively expensive. In addition, WSD systems are not suitable for newly created words, new senses of existing words, or domain-specific words. On the other hand, WSI systems can learn new senses of words directly from texts because these programs do not rely on a predefined set of senses.

In Section 2 we describe relevant previous work. In Section 3 and 4 we introduce the naive Bayes model for WSI and inference schemes for the model. In Section 5 we evaluate the model on SemEval-2010 data. In Section 6 we conclude.

2 Related Work

Yarowsky (1995) introduces a semi-supervised bootstrapping algorithm with two assumptions that rivals supervised algorithms: one-sense-per-collocation and one-sense-per-discourse. But this algorithm cannot easily be scaled up because for any new ambiguous word humans need to pick a few seed words, which initialize the algorithm. In order to automate the semi-supervised system, Eisner and Karakos (2005) propose an unsupervised bootstrapping algorithm. Their system tries many different seeds for bootstrapping and chooses the "best" classifier at the end. Eisner and Karakos's algorithm is limited in that their system is designed for disambiguating words that have only 2 senses.

Bayesian WSI systems have been developed by several authors. Brody and Lapata (2009) apply Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to WSI. They run a topic modeling algorithm on texts with some fixed number of topics that correspond to senses and induce a cluster by finding target words assigned to the same topic. Their system is evaluated on SemEval-2007 noun data (Agirre and Soroa, 2007). Lau et al. (2012) apply a nonparametric model, Hierarchical Dirichlet Processes (HDP), to SemEval-2010 data (Manandhar et al., 2010).

3 Model

Following Yarowsky (1995), we assume that a word in a document has one sense. Multiple occurrences of a word in a document refer to the same object or concept. The naive Bayes model is well suited for this one-sense-per-document assumption. Each document has one topic corresponding to the sense of the target word that needs disambiguation. Context words in a document are drawn from the conditional distribution of words given the sense. Context words are assumed to be independent from each other given

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1433–1437,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

the sense, which is far from being true yet effective.

3.1 Naive Bayes

The naive Bayes model assumes that every word in a document is generated independently from the conditional distribution of words given a sense, $p(w|s)$. The mathematical definition of the naive Bayes model is as follows:

$$\begin{aligned} p(\boldsymbol{w}) &= \sum_s p(s, \boldsymbol{w}) = \sum_s p(s)p(\boldsymbol{w}|s) \\ &= \sum_s p(s) \prod_w p(w|s), \end{aligned} \tag{1}$$

where $\boldsymbol{w}$ is a vector of words in the document. With the model, a new document can be easily labeled using the following classifier:

$$s' = \operatorname*{argmax}_s p(s) \prod_w p(w|s), \tag{2}$$

where s' is the label of the new document. In contrast to LDA-like models, it is easy to construct the closed form classifier from the model. The parameters of the model, $p(s)$ and $p(w|s)$, can be learned by maximizing the probability of the corpus, $p(\boldsymbol{d}) = \prod_d p(d) = \prod_{\boldsymbol{w}} p(\boldsymbol{w})$ where $\boldsymbol{d}$ is a vector of documents and $d = \boldsymbol{w}$.

3.2 Distance Incorporated Naive Bayes

Intuitively, context words near a target word are more indicative of its sense than ones that are farther away. To account for this intuition, we propose a more sophisticated model that uses the distance between a context word and a target word. Before introducing the new model, we define a probability distribution, $f(w|s)$, that incorporates distances as follows:

$$f(w|s) = \frac{p(w|s)^{l(w)}}{\sum_{w' \in \mathcal{W}} p(w'|s)^{l(w)}}, \tag{3}$$

where $l(w) = \frac{1}{dist(w)^x}$. $\mathcal{W}$ is a set of types in the corpus. x is a tunable parameter that takes nonnegative real values. With the new probability distribution, the model and the classifier become:

$$p(\boldsymbol{w}) = \sum_s p(s) \prod_w f(w|s) \tag{4}$$

$$s' = \operatorname*{argmax}_s p(s) \prod_w f(w|s), \tag{5}$$

where $f(w|s)$ replaces $p(w|s)$. The naive Bayes model is a special case; set $x = 0$. The new model puts more weight on context words that are close to the target word. The distribution of words that are farther away approaches the uniform distribution. $l(w)$ smoothes the distribution more as x becomes larger.

4 Inference

Given the generative model, we employ two inference algorithms to learn the sense distribution and word distributions given a sense. Expectation Maximization (EM) is a natural choice for the naive Bayes (Dempster et al., 1977). When initialized with random parameters, EM gets stuck at local maxima. To avoid local maxima, we use a Gibbs sampler for the plain naive Bayes to learn parameters that initialize EM.

5 Experiments

5.1 Data

We evaluate the model on SemEval-2010 WSI task data (Manandhar et al., 2010). The task has 100 target words, 50 nouns and 50 verbs. For each target word, there are training and test documents. Table 1 have details. The training and test data are plain texts without sense tags. For evaluation, the inferred sense labels are compared with human annotations. To tune some parameters we use the trial data of SemEval-2010. The trial data consists of training and test portions of 4 verbs. On average there are 137 documents for each target word in the training part of the trial data.

	Training	Testing	Senses (#)
All	879807	8915	3.79
Nouns	716945	5285	4.46
Verbs	162862	3630	3.12

Table 1: Details of SemEval-2010 data

5.2 Task

Participants induce clusters from the training data and use them to label the test data. Resources other than NLP tools for morphology and syntax such as lemmatizer, POS-tagger, and parser are not allowed. Tuning parameters and inducing clusters are only allowed during the training phase. After training, participants submit their sense-labeled test data to organizers.

LDA models are not compatible with the scoring rules for the SemEval-2010 competition, and that is the work against which we most want to compare. These rules require that training be done strictly before the testing is done. Note however that LDA requires learning the mixture weights of topics for each individual document $p(\text{topic} \mid \text{document})$. These are, of course, learned during training. But the documents in the testing corpus have never been seen before, so clearly their topic mixture weights are not learned during training, and thus not learned at all. The way to overcome this is by training on both train and test documents, but this is exactly what SemEval-2010 forbids.

5.3 Implementation Details

The documents are tokenized and stemmed by Stanford tokenizer and stemmer. Stop words and punctuation in the training and test data are discarded. Words that occur at most 10 times are discarded from the training data. Context words within a window of 50 about a target word are used to construct a bag-of-words.

When a target word appears more than once in a document, the distance between that target word and a context word is ambiguous. We define this distance to be minimum distance between a context word and an instance of the target word. For example, the word "chip" appears 3 times. For a context word, e.g., "shining" there are three possible distances: 8 away from the first "chip," 4 away from the second "chip" and 11 away from the last "chip." We set the distance of "shining" from the target to 4.

> $\cdots$ of memory **chip**s . Currently , **chip**s are produced by shining light through a mask to produce an image on the **chip** , much as $\cdots$

Example 1: an excerpt from "chip" test data

We model each target word individually. We set α, a Dirichlet prior for senses, to 0.02 and β, a Dirichlet prior for contextual words, to 0.1 for the Gibbs sampler as in Brody and Lapata (2009). We initialize EM with parameters learned from the sampler. We run EM until the likehood changes less than 1%. We run the sampler 2000 iterations including 1000 iterations of burn-in: 10 samples at an interval of 100 are averaged. For comparison, we also evaluate EM with random initialization. All reported scores (described in Section 5.4) are averaged over ten different runs of the program.[1]

5.3.1 Tuning Parameters

Two parameters, the number of senses and x of the function $l(w)$, need to be determined before running the program. To find a good setting we do grid search on the trial data with the number of senses ranging from 2 to 5 and x ranging from 0 to 1.1 with an interval 0.1. Due to the small size of the training portion of the trial data, words that occur once are thrown out in the training portion. All the other parameters are as described in Section 5.3. We choose (4, 0.4), which achieves the highest supervised recall. See Table 2 for the performance of the model with various parameter settings. With a fixed value of x, a column is nearly unimodal in the number of senses and vice versa. $x = 0$ is not optimal and there is some noticeable difference between scores with optimal x and scores with $x = 0$.

5.4 Evaluation

We compare our system to other WSI systems and discuss two metrics for unsupervised evaluation (V-Measure, paired F-Score) and one metric for supervised evaluation (supervised recall). We refer to the true group of tokens as a gold class and to an induced group of tokens as a cluster. We refer to the model learned with the sampler and EM as NB, and to the model learned with EM only as NB0.

5.4.1 Short Descriptions of Other WSI Systems Evaluated on SemEval-2010

The baseline assigns every instance of a target word with the most frequent sense (MFS). UoY runs a clustering algorithm on a graph with words as nodes and co-occurrences between words as edges (Korkontzelos and Manandhar, 2010). Hermit approximates co-occurrence space with Random Indexing and applies a hybrid of k-means and Hierarchical Agglomerate Clustering to co-occurrence space (Jurgens and Stevens, 2010). NMF$_{lib}$ factors a matrix using nonnegative matrix factorization and runs a clustering algorithm on test instances represented by factors (Van de Cruys et al., 2011).

5.4.2 V-Measure

V-Measure computes the quality of induced clusters as the harmonic mean of two values, homogeneity and completeness. Homogeneity measures whether instances of a cluster belong to a single gold class. Completeness measures whether instances of a gold class belong to a cluster. V-Measure is between 0 and 1; higher is better. See Table 3 for details of V-Measure evaluation (#cl is the number of induced clusters).

With respect to V-Measure, NB performs much better than NB0. This holds for paired F-Score and supervised recall evaluations. The sampler improves the log-likelihood of NB by 3.8% on average (4.8% on nouns and 2.9% on verbs).

Pedersen (2010) points out that it is possible to increase the V-Measure of bad models by increasing

[1] Code used for experiments is available for download at `http://cs.brown.edu/~dc65/`.

#s \ x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
2	74.73	**74.76**	74.41	74.57	74.06	74.07	74.18	74.33	74.14	74.22	74.15	74.52
3	74.60	74.71	75.21	75.46	75.21	75.57	**75.61**	75.32	75.53	75.56	74.98	74.79
4	74.52	75.06	74.97	75.14	**76.02**	75.51	75.74	75.51	75.59	75.51	75.37	75.35
5	73.40	73.88	74.93	75.13	74.79	74.68	74.71	74.49	75.11	74.94	74.86	**75.25**

Table 2: Performance of the model with various parameters: supervised recall on the trial data. The best value from each row is bold-faced. The scores are averaged over 100 runs.

VM(%)	all	nouns	verbs	#cl
NB	**18.0**	**23.7**	9.9	3.42
NB0	14.9	19.0	9.0	3.77
Hermit	16.2	16.7	**15.6**	10.78
UoY	15.7	20.6	8.5	11.54
NMF$_{lib}$	11.8	13.5	9.4	4.80
MFS	0.0	0.0	0.0	1.00

Table 3: Unsupervised evaluation: V-Measure

the number of clusters. But increasing the number of clusters harms paired F-Score, which results in bad supervised recalls. NB attains a very high V-Measure with few induced clusters, which indicates that those clusters are high quality. Other systems use more induced clusters but fail to attain the V-Measure of NB.

5.4.3 Paired F-Score

Paired F-Score is the harmonic mean of paired recall and paired precision. Paired recall is fraction of pairs belonging to the same gold class that belong to the same cluster. Paired precision is fraction of pairs belonging to the same cluster that belong to the same class. See Table 4 for details of paired F-Score evaluation.

As with V-Measure, it is possible to attain a high paired F-Score by producing only one cluster. The baseline, MFS, attains 100% paired recall, which together with the poor performance of WSI systems makes its paired F-Score difficult to beat. V-Measure and paired F-Score are meaningful when systems produce about the same numbers of clusters as the numbers of classes and attain high scores on these metrics.

FS(%)	all	nouns	verbs	#cl
MFS	**63.5**	**57.0**	**72.7**	1.00
NB	52.9	52.5	53.5	3.42
NB0	46.8	47.4	46.0	3.77
UoY	49.8	38.2	66.6	11.54
NMF$_{lib}$	45.3	42.2	49.8	4.80
Hermit	26.7	24.4	30.1	10.78

Table 4: Unsupervised evaluation: paired F-Score

5.4.4 Supervised Recall

For the supervised task, the test data is split into two groups: one for mapping clusters to classes and the other for standard WSD evaluation. 2 different split schemes (80% mapping, 20% evaluation and 60% mapping, 40% evaluation) are evaluated. 5 random splits are averaged for each split scheme. Mapping is induced automatically by the program provided by organizers. See Table 5 for details of supervised recall evaluation (#s is the average number of classes mapped from clusters).[2]

SR(%)	all	nouns	verbs	#s
NB	**65.4**	**62.6**	69.5	1.72
NB0	63.5	59.8	69.0	1.76
NMF$_{lib}$	62.6	57.3	**70.2**	1.82
UoY	62.4	59.4	66.8	1.51
MFS	58.7	53.2	66.6	1.00
Hermit	58.3	53.6	65.3	2.06

Table 5: Supervised evaluation: supervised recall, 80% mapping and 20% evaluation

Overall our system performs better than other systems with respect to supervised recall. When a system has higher V-Measure and paired F-Score on nouns than another system, it achieves a higher supervised recall on nouns too. However, this behavior is not observed on verbs. For example, NB has higher V-Measure and paired F-Score on verbs than NMF$_{lib}$ but NB attains a lower supervised recall on verbs than NMF$_{lib}$. It is difficult to see which verbs clusters are better than some other clusters.

6 Conclusion

Of the four SemEval-2010 evaluation metrics, and restricting ourselves to systems obeying the evaluation conditions for that competition, our new model achieves new best results on three. The exception is paired F-Score. As we note earlier, this metric tends to assign very high scores when every word receives only one sense, and our model is bested by the baseline system that does exactly that.

[2] 60-40 split is omitted here due to almost identical result.

If we loosen possible comparison systems, the LDA/HDP model of Lau et al. (2012) achieves superior numbers to ours for the two supervised metrics, but at the expense of requiring LDA type processing on the test data, something that the SemEval organizers ruled out, presumably with the reasonable idea that such processing would not be feasible in the real world. More generally, their system assigns many senses (about 10) to each word, and thus nodoubt does poorly on the paired F-Score (they do not report results on V-Measure and paired F-Score).

References

Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 7–12. Association for Computational Linguistics.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 103–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Jason Eisner and Damianos Karakos. 2005. Bootstrapping without the boot. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 395–402, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Jurgens and Keith Stevens. 2010. Hermit: Flexible clustering for the semeval-2 wsi task. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 359–362. Association for Computational Linguistics.

Ioannis Korkontzelos and Suresh Manandhar. 2010. Uoy: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 355–358. Association for Computational Linguistics.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601. Association for Computational Linguistics.

Suresh Manandhar, Ioannis P Klapaftis, Dmitriy Dligach, and Sameer S Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68. Association for Computational Linguistics.

Ted Pedersen. 2010. Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 363–366. Association for Computational Linguistics.

Tim Van de Cruys, Marianna Apidianaki, et al. 2011. Latent semantic word sense induction and disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT)*, pages 1476–1485.

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.