Addressing Ambiguity in Unsupervised Part-of-Speech Induction with Substitute Vectors

Volkan Cirik
Artificial Intelligence Laboratory
Koc University, Istanbul, Turkey
vcirik@ku.edu.tr

Abstract

We study substitute vectors to solve the part-of-speech ambiguity problem in an unsupervised setting. Part-of-speech tagging is a crucial preliminary process in many natural language processing applications. Because many words in natural languages have more than one part-of-speech tag, resolving part-of-speech ambiguity is an important task. We claim that part-of-speech ambiguity can be solved using substitute vectors. A substitute vector is constructed with possible substitutes of a target word. This study is built on previous work which has proven that word substitutes are very fruitful for part-of-speech induction. Experiments show that our methodology works for words with high ambiguity.

1 Introduction

Learning syntactic categories of words (i.e. part-of-speech or POS tagging) is an important preprocessing step for many natural language processing applications because grammatical rules are not functions of individual words, instead, they are functions of word categories. Unlike supervised POS tagging systems, POS induction systems make use of unsupervised methods. They categorize the words without any help of annotated data.

POS induction is a popular topic and several studies (Christodoulopoulos et al., 2010) have been performed. Token based methods (Berg-Kirkpatrick and Klein, 2010; Goldwater and Griffiths, 2007) categorize word occurrences into syntactic groups. Type based methods (Clark, 2003; Blunsom and Cohn, 2011) on the other hand, categorize word types and yield the ambiguity problem unlike the token based methods.

Type based methods suffer from POS ambiguity because one POS tag is assigned to each word type. However, occurrences of many words may have different POS tags. Two examples below are drawn from the dataset we worked on. They illustrate a situation where two occurrences of the "offers" have different POS tags. In the first sentence "offers" is a noun, whereas, in the second sentence it is a verb.

> (1) "Two rival bidders for Connaught BioSciences extended their **offers** to acquire the Toronto-based vaccine manufacturer Friday."
>
> (2) "The company currently **offers** a word-processing package for personal computers called Legend."

In this study, we try to extend the state-of-the-art unsupervised POS tagger (Yatbaz et al., 2012) by solving the ambiguity problem it suffers because it has a type based approach. The clustering based studies (Schütze, 1995) (Mintz, 2003) represent the context of a word with a vector using neighbour words. Similarly, (Yatbaz et al., 2012) proposes to use word context. They claim that the substitutes of a word have similar syntactic categories and they are determined by the context of the word.

In addition, we suggest that the occurrences with different part-of-speech categories of a word should be seen in different contexts. In other words, if we categorize the contexts of a word type we can determine different POS tags of the word. We represent the context of a word by constructing substitute vectors using possible substitutes of the word as (Yatbaz et al., 2012) suggests.

Table 1 illustrates the substitute vector of the occurrence of "offers" in (1). There is a row for each word in the vocabulary. For instance, probability of occurring "agreement" in the position of "offers" is 80% in this context. To resolve ambiguity

Proceedings of the ACL Student Research Workshop, pages 117–122,
Sofia, Bulgaria, August 4-9 2013.

Probability	Substitute Word
0.80	agreement
0.03	offer
0.01	proposal
0.01	bid
0.01	attempt
0.01	bids
.	.
.	.
.	.

Table 1: Substitute Vector for "offers" in above sentence.

of a target word, we separate occurrences of the word into different groups depending on the context information represented by substitute vectors.

We conduct two experiments. In the first experiment, for each word type we investigated, we separate all occurences into two categories using substitute vectors. In the second one we guess the number of the categories we should separate for each word type. Both experiments achieve better than (Yatbaz et al., 2012) for highly ambiguous words. The level of ambiguity can be measured with perplexity of word's gold tag distribution. For instance,the gold tag perplexity of word "offers" in the Penn Treebank Wall Street Journal corpus we worked on equals to 1.966. Accordingly, the number of different gold tags of "offers" is 2. Whereas, perplexity of "board" equals to 1.019. Although the number of different tags for "board" is equal to 2, only a small fraction of the tags of board differs from each other. We can conclude that "offers" is more ambiguous than "board".

In this paper we present a method to solve POS ambiguity for a type based POS induction approach. For the rest of the paper, we explain our algorithm and the setup of our experiments. Lastly we present the results and a conclusion.

2 Algorithm

We claim that if we categorize contexts a word type occurs in, we can address ambiguity by separating its occurrences before POS induction. In order to do that, we represent contexts of word occurrences with substitute vectors. A substitute vector is formed by the whole vocabulary of words and their corresponding probabilities of occurring in the position of the target word. To calculate these probabilities, as described in (Yatbaz et al., 2012), a 4-gram language model is built with SRILM (Stolcke, 2002) on approximately 126 million tokens of Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff et al., 1995).

We generate substitute vectors for all tokens in our dataset. We want to cluster occurrences of our target words using them. In each substitute vector, there is a row for every word in the vocabulary. As a result, the dimension of substitute vectors is equal to 49,206. Thus, in order not to suffer from the curse of dimensionality, we reduce dimensions of substitute vectors.

Before reducing the dimensions of these vectors, distance matrices are created using Jensen distance metric for each word type in step (a) of Figure 1. We should note that these matrices are created with substitute vectors of each word type, not with all of the substitute vectors.

In step (b) of Figure 1, to reduce dimensionality, the ISOMAP algorithm (Tenenbaum et al., 2000) is used. The output vectors of the ISOMAP algorithm are in 64 dimensions. We repeated our experiments for different numbers of dimensions and the best results are achieved when vectors are in 64 dimensions.

In step (c) of Figure 1, after creating vectors in lower dimension, using a modified k-means algorithm (Arthur and Vassilvitskii, 2007) 64-dimensional vectors are clustered for each word type. The number of clusters given as an input to k-means varies with experiments. We induce number of POS tags of a word type at this step.

Previous work (Yatbaz et al., 2012) demonstrates that clustering substitute vectors of all word types alone has limited success in predicting part-of-speech tag of a word. To make use of both word identity and context information of a given type, we use S-CODE co-occurrence modeling (Maron et al., 2010) as (Yatbaz et al., 2012) does.

Given a pair of categorical variables, the S-CODE model represents each of their values on a unit sphere such that frequently co-occurring values are located closely. We construct the pairs to feed S-CODE as follows.

In step (d) of Figure 1, the first part of the pair is the word identity concatenated with cluster ids we got from the previous step. The cluster ids separate word occurrences seen in different context groups. By doing that, we make sure that the occurrences

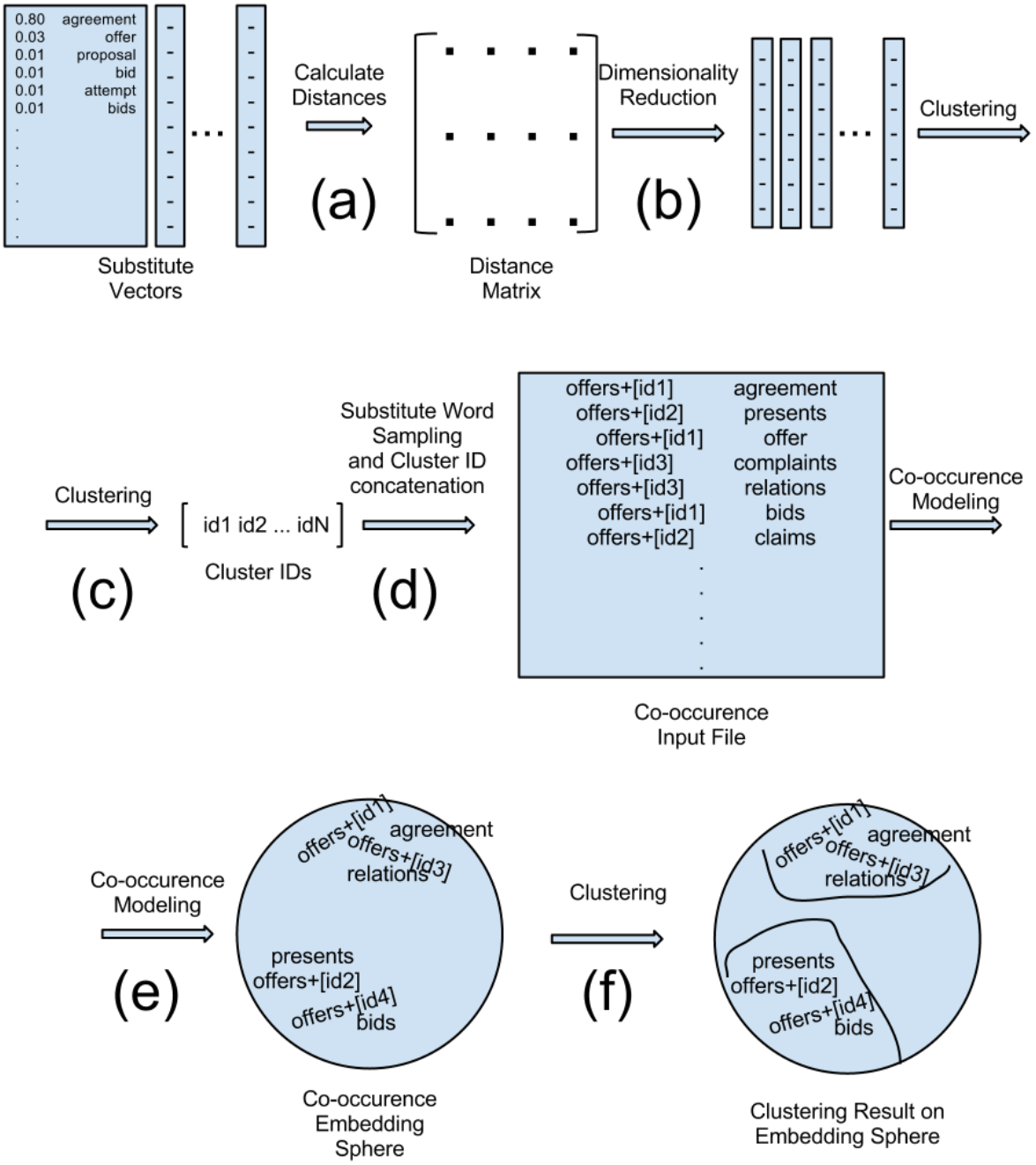

Figure 1: General Flow of The Algorithm

of a same word can be separated on the unit sphere if they are seen in different context groups.

The second part of the pair is a substitute word. For an instance of a target word, we sample a substitute word according to the target word's substitute vector probabilities. If occurrences of two different or the same word types have the same substitutes, they should be seen in the similar contexts. As a result, words occurring in the similar contexts will be close to each other on the unit sphere. Furthermore, they will have the same POS tags. We should note that the co-occurrence input file contains all word types.

In step (e) of Figure 1, on the output of the S-CODE sphere, the words occurring in the similar contexts and having the same word-identity are closely located. Thus, we observe clusters on the unit sphere. For instance, verb occurrences of "offers" are close to each other on the unit sphere. They are also close to other verbs. Furthermore,

they are separated with occurrences of "offers" which are nouns.

Lastly, in step (f) of Figure 1, we run k-means clustering method on the S-CODE sphere and split word-substitute word pairs into 45 clusters because the treebank we worked on uses 45 part-of-speech tags. The output of clustering induces part-of-speech categories of words tokens.

3 Experiments

In this section, the setup of each experiment will be presented. The experiments are conducted on Penn Treebank Wall Street Journal corpus. There are 1,173,766 tokens and, 49,206 types. Out of 49,206 word types, 1183 of them are chosen as target words. They are fed to the algorithm described above. Occurrences of these target words correspond to 37.55% of the whole data. These target words are seen in the dataset more than 100 times and less than 4000 times. This subset is chosen as such because word types occurring more than 4000 times are all with low gold tag perplexity. They also increase computation time dramatically. We exclude word types occurring less than 100 times, because the clustering algorithm running on 64-dimension vectors does not work accurately. To avoid providing noisy results, the experiments are repeated 10 times. We report many-to-one scores of the experiments. The many-to-one evaluation assigns each cluster to its most frequent gold-tag. Overall result demonstrates the percentage of correctly assigned instances and standard deviation in paranthesis.

3.1 Baseline

Because we are trying to improve (Yatbaz et al., 2012), we select the experiment on Penn Treebank Wall Street Journal corpus in that work as our baseline and replicate it. In that experiment, POS induction is done by using word identities and context information represented by substitute words. Strictly one tag is assigned to each word type. As a result, this method inaccurately induces POS tags for the occurrences of word types with high gold tag perplexity. The many-to-one accuracy of this experiment is 64%.

3.2 Upperbound

In this experiment, for each word occurence, we concatenate the gold tag for the first part of the pairs in the co-occurence input file. Thus, we skipped steps (a), (b), (c). The purpose of this experiment is to set an upperbound for all experiments since we cannot cluster the word tokens any better than the gold tags. The many-to-one accuracy of this experiment is 67.2%.

3.3 Experiment 1

In the algorithm section, we mention that after dimensionality reduction step, we cluster the vectors to separate tokens of a target word seen in the similar contexts. In this experiment, we set the number of clusters for each type to 2. In other words, we assume that the number of different POS tags of each word type is equal to 2. Nevertheless, separating all the words into 2 clusters results in some inaccuracy in POS induction. That is because not all words have POS ambiguity and some have more than 2 different POS tags However, the main purpose of this experiment is to observe whether we can increase the POS induction accuracy for ambiguous types with our approach. The many-to-one accuracy of this experiment is 63.8%.

3.4 Experiment 2

In the previous experiment, we set the number of clusters for each word type to 2. However, the number of different POS tags differs for each word type. More importantly, around 41% of our target tokens belongs to unambiguous word types. Also, around 36% of our target tokens comes from word types whose gold perplexity is below 1.5. That means, the Experiment 1 splits most of our word types that should not be separated.

In this experiment, instead of splitting all types, we guess which types should be splitted. Also, we guess the number of clusters for each type. We use gap statistic (Tibshirani et al., 2001) on 64-dimensional vectors. The Gap statistic is a statistical method to guess the number of clusters formed in given data points. We expect that substitute vectors occurring in the similar context should be closely located in 64-dimensional space. Thus, gap statistic can provide us the number of groups formed by vectors in 64-dimensional space. That number is possibly equal to the number of the number of different POS tags of the word types. The many-to-one accuracy of this experiment is 63.4%.

3.5 Experiment 3

In this experiment, we set the number of clusters for each type to gold number of tags of each type. The purpose of this experiment is to observe how the accuracy of number of tags given, which is used at step (c), affects the system. The many-to-one accuracy of this experiment is 63.9%.

3.6 Overall Results

In this section we present overall results of the experiments. We present our results in 3 separated tables because the accuracy of these methods varies with the ambiguity level of word types.

In Table 2, many-to-one scores of three experiments are presented. Since we exclude some of the word types, our results correspond to 37.55% of the data. In Table 3, results for the word types whose gold tag perplexity is lower than 1.5 are presented. They correspond to 29.11% of the data. Lastly, in Table 4, we present the results for word types whose gold tag perplexity is greater than 1.5.

Experiment	Many-to-One Score
Baseline	.64 (.01)
Experiment 1	.638 (.01)
Experiment 2	.634 (.01)
Experiment 3	.639 (.02)

Table 2: Results for the target words corresponding to 37.55% of the data.

Experiment	Many-to-One Score
Baseline	.693 (.02)
Experiment 1	.682 (.01)
Experiment 2	.68 (.01)
Experiment 3	.684 (.02)

Table 3: Results for Target Words with gold tag perplexity $\leq$1.5 which corresponds to 29.11% of the data.

Experiment	Many-to-One Score
Baseline	.458 (.01)
Experiment 1	.484 (.01)
Experiment 2	.474 (.02)
Experiment 3	.483 (.02)

Table 4: Results for Target Words with gold tag perplexity $\geq$1.5 which corresponds to 8.44% of the data..

4 Conclusion

Table 2 shows that the baseline experiment is slightly better than our experiments. That is because our experiments inaccurately induce more than one tag to unambiguous types. Additionally, most of our target words have low gold tag perplexity. Table 3 supports this claim. In Table 4, we observe that our methods outscore the baseline significantly. That is because, when ambiguity increases, the baseline method inaccurately assigns one POS tag to word types. On the other hand, the gap statistic method is not fully efficient in guessing the number of clusters. It sometimes separates unambiguous types or it does not separate highly ambiguous word types. As a result, there is a slight difference between the results of our experiments.

Additionally, the results of our experiments show that, accurately guessing number of clusters plays a crucial role in this approach. Even using the gold number of different tags in Experiment 3 does not result in a significantly accurate system. That is because, the number of different tags does not reflect the perplexity of a word type.

The results show that, POS ambiguity can be addressed by using substitute vectors for word types with high ambiguity. The accuracy of this approach correlates with the level of ambiguity of word types. Thus, the detection of the level of ambiguity for word types should be the future direction of this research. We again propose that substitute vector distributions could be useful to extract perplexity information for a word type.

Acknowledgments

I would like to thank the members of the Koc University Artificial Intelligence Laboratory for their help and support. Additionally, I would like to thank two anonymous reviewers and Murat Seyhan for their comments and suggestions.

References

D. Arthur and S. Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297, Uppsala,

Sweden, July. Association for Computational Linguistics.

Phil Blunsom and Trevor Cohn. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June. Association for Computational Linguistics.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: how far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 575–584, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 59–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June. Association for Computational Linguistics.

David Graff, Roni Rosenfeld, and Doug Paul. 1995. Csr-iii text. Linguistic Data Consortium, Philadelphia.

Yariv Maron, Michael Lamar, and Elie Bienenstock. 2010. Sphere embedding: An application to part-of-speech induction. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1567–1575.

T.H. Mintz. 2003. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90(1):91–117.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, EACL '95, pages 141–148, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November.

J.B. Tenenbaum, V. Silva, and J.C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319.

R. Tibshirani, G. Walther, and T. Hastie. 2001. Estimating the number of data clusters via the gap statistic. *Journal of the Royal Statistical Society B*, 63:411–423.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July. Association for Computational Linguistics.