Unsupervised Part-Of-Speech Tagging Supporting Supervised Methods

Chris Biemann
University of Leipzig, NLP Dept.
Johannisgasse 26
04103 Leipzig, Germany
biem@informatik.uni-leipzig.de

Claudio Giuliano
ITC-IRST
Via Sommarive, 18
I-38050 Povo (Trento), Italy
giuliano@itc.it

Alfio Gliozzo
ITC-IRST
Via Sommarive, 18
I-38050 Povo (Trento), Italy
gliozzo@itc.it

Abstract

This paper investigates the utility of an unsupervised part-of-speech (PoS) system in a task oriented way. We use PoS labels as features for different supervised NLP tasks: Word Sense Disambiguation, Named Entity Recognition and Chunking. Further we explore, how much supervised tagging can gain from unsupervised tagging. A comparative evaluation between variants of systems using standard PoS, unsupervised PoS and no PoS at all reveals that supervised tagging gains substantially from unsupervised tagging. Further, unsupervised PoS tagging behaves similarly to supervised PoS in Word Sense Disambiguation and Named Entity Recognition, while only chunking benefits more from supervised PoS. Overall results indicate that unsupervised PoS tagging is useful for many applications and a veritable low-cost alternative, if none or very little PoS training data is available for the target language or domain.

Keywords

Unsupervised PoS Tagging, Named Entity Recognition, Word Sense Disambiguation, Chunking

1. Introduction

Even if, in principle, supervised approaches reach the best performance in many NLP tasks, in practice it is not always easy to make them work in applicative settings. In fact, supervised systems require to be trained on a large amount of manually provided annotations. In most of the cases this scenario is quite unpractical, if not infeasible. In the NLP literature the problem of providing large amounts of manually annotated data is known as the knowledge acquisition bottleneck. A promising direction to tackle this problem is to provide unlabeled data together with labeled texts, which is called semi-supervised learning.

The underlying idea behind our approach is that syntactic similarity of words is an inherent property of corpora, and it can be exploited to help a supervised classifier to build a better categorization hypothesis, even if the amount of labeled training data provided for learning is very low.

Previous work on distributional clustering for word class induction was mostly not evaluated in an application-based way. [4] and [7] state that their clustering examples look plausible. [17], [5] and [8] evaluate their tagging by comparing it to predefined tagsets. Notable exceptions to this are [20], where distributional clustering supports a supervised PoS tagger (see Section 3.1), and the incorporation of an unsupervised tagger into a NER system in [9] (see Section 4.3).

This is, to our knowledge, the first comprehensive study on the utility of distributional word classes for a variety of NLP tasks. As the same unsupervised tagger is used for all tasks tested, we show the robustness of the system across tasks and languages.

In this work, the unsupervised PoS tagger as described in [2] is evaluated by testing performance of applications equipped with this tagger. Section 2 is devoted to a short description of the tagger; Section 3 lays out the systems the tagger has been incorporated into. In Section 4, evaluation results examine the competitiveness of the unsupervised tagger, Section 5 concludes.

2. Unsupervised PoS tagging

Unlike in standard (supervised) PoS tagging, the unsupervised variant relies neither on a set of predefined categories, nor on any labeled text. As a PoS tagger is not an application of its own right, but serves as a preprocessing step for systems building upon it, the names and the number of syntactic categories is very often not important.

The basic procedure behind our unsupervised PoS tagging is as follows: (i) (soft) clusters of contextually similar words are identified, each class is assumed being a different PoS, and (ii) words belonging to more than one class are disambiguated by considering the context in which they are located. The clustering methodology at the basis of the first step is motivated by the fact that words belonging to the same syntactic classes can be substituted in the same context producing grammatical sentences as well, leading us to adopt contextual similarity features for clustering.

For a detailed description of the unsupervised PoS tagger system, we refer to [2]. Increased lexicon size up to some 50,000 words is the main difference between this and other approaches (cf. Section 1.1), that typically operate with 5,000 clustered words. The tagsets obtained with this method are usually more fine-grained than standard tagsets and reflect syntactic as well as semantic similarity.

In [2], the tagger output was directly evaluated against supervised taggers for English, German and Finnish via information-theoretic measures. While it is possible to

relatively compare the performance of different components of a system or different systems along this scale, it does only give a poor impression on the utility of the unsupervised tagger's output. Therefore, an application-based evaluation is undertaken here.

Corpus	*BNC*	*CLEF*	*Wortschatz*
Language	English	Dutch	German
Size (Tokens)	100M	70M	755M
Nr. of Tags	344	418	511
Lexicon Size	25706	21863	74398

Table 2: Three corpora used for the induction of tagger models. BNC = British National Corpus, for CLEF see [14], Wortschatz is described in [15]

To induce tagger models, three different corpora are used in our experiments. Table 2 lists some corpus characteristics as well as quantitative data of the respective tagger model.

3. Supervised NLP Systems

In this section, the systems that are used for evaluation are described: a simple Viterbi trigram tagger as used in [2], the supervised WSD system of [10], and the simple NER and chunking systems we set up.

In the design of all of these systems, the task is perceived as a machine learning exercise: the PoS tagger component provides some of the features that are used to learn a function that assigns a label to unseen examples, characterized by the same set of features as the examples used for training.

The systems were chosen to cover a wide range of machine learning paradigms: Markov chains in the PoS tagging system, kernel methods in the WSD system and Conditional Random Fields (CRFs, see [11]) for NER and chunking.

3.1 PoS Tagger

The tagger employed in [2] is a very simple trigram tagger that does not use parameter re-estimation or smoothing techniques. It was designed to be trained from large amounts of unlabeled data, arguing that increasing training data will lead to better results than increasing model complexity, cf. [1]. For training, the frequency of tag trigrams and the number of times each word occurs with each tag are counted and directly transformed into (transition) probabilities by normalization.

The sequence of tags for a chunk of text is found by maximizing the probability of the joint occurrence of tokens $T=(t_i)$ and categories/tags $C=(c_i)$ for a sequence of length n:

$$P_{plain}(T,C) = \prod_{i=1}^{n} P(c_i \mid c_{i-1}, c_{i-2}) P(c_i \mid t_i) .$$

In the unsupervised case, the transition probabilities $P(c_i|c_{i-1},c_{i-2})$ are only estimated from trigrams where all three tags are present. In the supervised case, tags are provided for all tokens in the training corpus. The probability $P(c_i|t_i)$ [1] is obtained from the tagger's lexicon and equals 1 if t_i is not contained.

For the incorporation of unsupervised tags, another factor $P(c_i|u_i)$ is introduced that accounts for the fraction of times the supervised tag c_i was found together with the unsupervised tag u_i in the training text, which has been tagged with the unsupervised tagger before:

$$P_{unsu}(T,C) = \prod_{i=1}^{n} P(c_i \mid c_{i-1}, c_{i-2}) P(c_i \mid t_i) P(c_i \mid u_i) .$$

Notice that only the unsupervised tag at the same position influences the goal category in this simple extension. Using surrounding unsupervised tags would be possible, but was not carried out. More elaborate strategies, like morphological components as in [3] or the utilization of a more up-to-date tagger model, are not considered here. The objective is to examine the influence of unsupervised tags, not to construct a state of the art PoS tagger.

A somewhat related strategy is described in [20], where a hierarchical clustering of words was used for reducing the error rate of a decision-tree-based tagger up to 43%, achieving 87% accuracy on a fine-grained tagset. However, the improvements were reached by manually adding rules that made use of the cluster IDs yielded by a word clustering method and this approach therefore caused extra work as opposed to narrowing down the acquisition bottleneck.

3.2 Word Sense Disambiguation (WSD)

For performing WSD, we used a state of the art supervised WSD methodology based on a combination of syntagmatic and domain kernels [10] in a Support Vector Machine classification framework.

Kernel WSD basically takes two different aspects of similarity into account: domain aspects, mainly related to the topic (i.e. the global context) of the texts in which the word occurs, and syntagmatic aspects, concerning the lexical-syntactic pattern in the local contexts. Domain aspects are captured by the *domain kernel*, while syntagmatic aspects are taken into account by the *syntagmatic kernel*.

For our experiments, we substitute the sequences of PoS required by the syntagmatic kernel by using

[1] Although [6] report that using $P(t_i|c_i)$ instead leads to superior results in the supervised setting, we use the 'direct' lexicon probability, which does not require smooting and re-estimation. For the purely unsupervised setting, this does not affect results negatively, as a much larger training corpus levels out the effects measured in [6].

unsupervised PoSs, comparing the results obtained with different combinations.

3.3 Named Entity Recognition and Chunking

For performing chunking and NER, we perceived these applications as a tagging task. For both tasks, we train the MALLET tagger[2].

The tagger operates on a different set of features for our two tasks. In the NER system, the following features are accessible, time-shifted by -2, -1, 0, 1, 2: a) Word itself, b) PoS-tag, c) Orthographic predicates and d) Character bigram and trigram predicates.
In the case of chunking, features are only time-shifted by -1, 0, 1 and consist only of: a) Word itself and b) PoS-tag.

Per system, three experiments were carried out, using standard PoS features, unsupervised PoS features and no PoS features.

4. Evaluation

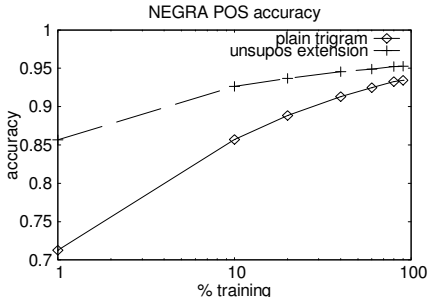
The systems are tested in a standard way on annotated resources. For supervised PoS tagging, we evaluate on the German NEGRA corpus [18]. The English lexical sample task (fine-grained scoring) of Senseval-3 [12] is chosen for WSD. For NER, the Dutch dataset of CoNLL-2002 [16] is employed, and the evaluation set for English chunking is the CoNLL-2000 dataset [19]. The supervised PoS tags for WSD, NER and chunking were provided in the respective datasets.

Supervised PoS tagging is measured in accuracy, which is obtained through dividing the number of correctly classified instances by the total number of instances. For NER and chunking, results are reported in terms of the F1[3] measure. WSD performance is measured using the scorer provided by Senseval-3. All evaluation results are compared in a pair wise fashion using the approximate randomization procedure of [13] as significance test.

4.1 Unsupervised PoS for supervised PoS

To evaluate the influence of unsupervised tags on a supervised tagger, training sets of varying sizes were selected randomly from the 20,000 sentences of NEGRA corpus, the remainder was used for evaluation. We compare the performance of the plain Viterbi tagger with the performance of the tagger using unsupervised tags (cf. formulae in section 3.1), which were obtained by tagging the NEGRA corpus with a tagger model induced on the Wortschatz corpus, which is 2,000 times larger. Results are reported in tagging accuracy, averaged over three different splits per training size each. Figure 1 shows the learning curve.

Results indicate that supervised tagging can clearly benefit from unsupervised tags: already at 20% training with unsupervised tags, the performance on 90% training without the unsupervised extension is surpassed. At 90% training, error rate reduction is 27.8%, indicating that the unsupervised tagger grasps very well the linguistically motivated syntactic categories and provides a valuable feature to either reduce the size of the required annotated training corpus or to improve overall accuracy. Despite its simplicity, the unsupervised extension does not fall too short of the performance of [3], where an accuracy of 0.967 at 90% training on the same corpus is reported.

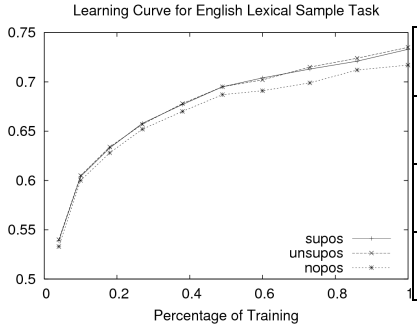


%	*1*	*10*	*20*	*40*	*60*	*80*	*90*
plain	0.713	0.857	0.888	0.913	0.925	0.933	0.934
unsu.	**0.857**	**0.926**	**0.937**	**0.946**	**0.949**	**0.952**	**0.953**

Figure 1: Learning curve for supervised PoS tagging with and without using unsupervised PoS tags (accuracy)

4.2 Unsupervised PoS for WSD

The modularity of the kernel approach makes it possible to easily compare systems with different configurations by testing various kernel combinations. To examine the influence of PoS tags, two comparative experiments were undertaken.



	only PoS	*full*
no PoS	N/A	0.717
sup. PoS	0.629	0.733
uns. PoS	**0.633**	**0.735**

Figure 2: Comparative evaluation on Senseval scores for WSD and learning curve. No differences are significant at p<0.1

The first experiment uses only the PoS kernel, i.e. the PoS labels are the only feature visible to the learning and classification algorithm. In a second experiment, the full system of [10] is tested against replacing the original PoS kernel with the unsupervised PoS kernel and omitting the

[2] http://mallet.cs.umass.edu

[3] $F1 = \frac{2PR}{P+R}$ with $P = \frac{\#correct}{\#classified}, R = \frac{\#correct}{\#total}$

PoS kernel completely. Figure 2 summarizes the results in terms of accuracy.

Results show that PoS information generally contributes to a small extent to WSD accuracy in the full system. Using the unsupervised PoS tagger results in a slight performance increase, improving over the state of the art results in this task, that have been previously achieved by [10]. However, the learning curve suggests that it does not matter whether to use supervised or unsupervised tagging.

From this, we conclude that supervised tagging can safely be exchanged in kernel WSD with the unsupervised variant. Replacing the only preprocessing step that is dependent on manual resources in the system of [10], state of the art supervised WSD is proven to not being dependent on any linguistic preprocessing at all.

4.3 NER Evaluation

To evaluate the performance on NER, we employ the methodology as proposed by the providers of the CoNLL-2002 dataset. We provide no PoS information, supervised PoS information and unsupervised PoS information to the system and measure the difference in performance in terms of F1. Table 3 summarizes the results for this experiment for selected categories using the full train set for training and evaluating on the test data.

Table 3: Comparative evaluation of NER on the Dutch CoNLL-2002 dataset in terms of F1. All differences are not significant with p<0.1

Category	*PER*	*ORG*	*LOC*	*MISC*	*ALL*
no PoS	0.8084	**0.7445**	0.8151	0.7462	0.7781
su. PoS	**0.8154**	0.7418	0.8156	**0.7660**	**0.7857**
un. PoS	0.8083	0.7357	**0.8326**	0.7527	0.7817

The figures in table 3 indicate that PoS information is hardly contributing anything to the system's performance, be it supervised or unsupervised. This indicates that the training set is large enough to compensate for the lack of generalization when using no PoS tags, in line with e.g. [1]. The situation changes when taking a closer look on the learning curve, produced by using train set fractions of differing size. Figure 3 shows the learning curves for the categories *LOCATION* and the micro average F1 evaluated over all the categories (ALL).

On the *LOCATION* category, unsupervised PoS tags provide a high generalization power for a small number of training samples. This is due to the fact that the induced tagset treats locations as a different tag; the tagger's lexicon plays the role of a gazetteer in this case, comprising 765 lexicon entries for the location tag. On the combination of ALL categories, this effect is smaller, yet the incorporation of PoS information outperforms the system without PoS for small percentages of training.

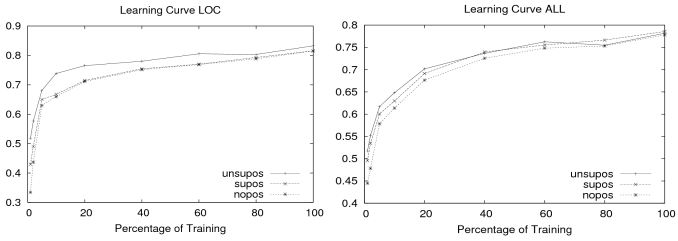


Figure 3: Learning curves in NER task in F1 for category LOC and combined category

This disagrees with the findings of [9], where features produced by distributional clustering were used in a boosting algorithm. Freitag reports improved performance on *PERSON* and *ORGANISATION*, but not on *LOCATION*, as compared to not using a tagger at all. In [9], however, a different training corpus for PoS induction and English NER data was used.

Experiments on NER reveal that PoS information is not making a difference, as long as the training set is large enough. For small training sets, usage of unsupervised PoS features result in higher performance than supervised or no PoS, which can be attributed to its more fine-grained tagset.

4.4 Chunking Evaluation

For testing performance of our simple chunking system, we used different portions of the training set as given in the CoNLL-2000 data and evaluated on the provided test set. Performance is reported in Figure 4.

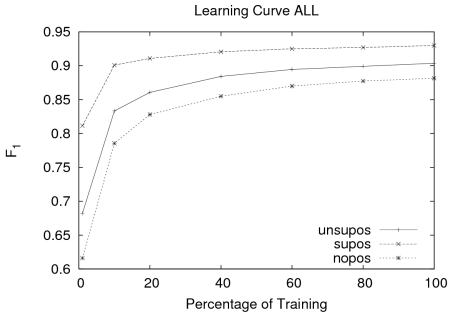


Figure 4: Learning curve for the chunking task in terms of F1. Performance at 100% training is 0.882 (no PoS), 0.904 (unsupervised PoS) and 0.930 (supervised PoS), respectively

As PoS is the only feature that is used here apart from the word tokens themselves, and chunking reflects syntactic structure, it is not surprising that providing this feature to the system results in increased performance: both kinds of PoS significantly outperform not using PoS (p<0.01).

In contrast to the previous systems tested, using the supervised PoS labels resulted in a significantly better chunking (p<0.01) than using the unsupervised labels. This can be attributed to the fact that both supervised tagging and chunking aim at reproducing the same perception of syntax, which does not necessarily fit the distributionally acquired classes of an unsupervised system. Anyhow, the use of unsupervised PoS provide very useful information to

the chunking learning process, demonstrated by the fact that the use of unsupervised PoS improves significantly the baseline provided by the system trained without PoS.

Despite the low number of features, the chunking system using supervised tags compares well with the best system in the CoNLL-2000 evaluation (F1=0.9348).

5. Conclusion

To summarize our results, we have shown that employing unsupervised PoS tags as features are useful in many NLP tasks. Improvements over the pure word level could be observed in all systems tested. We demonstrated that especially if few training data or no supervised PoS tagger is available, using this low-cost alternative leads to significantly better performance and should be used beyond doubt. In addition, unsupervised PoS tagging can be used to improve supervised PoS tagging, especially as far as the learning curve is concerned.

Comparing the two kinds of PoS tags tested, we observed that the performances achieved by the final systems are comparable in all tasks but chunking. In addition, we reported a slight improvement on WSD.

Another conclusion is that, in general, the more training data is provided, the lower the gain of using PoS tagging in supervised NLP, either if PoS tags are supervised or not. Even if this result is in itself not very interesting from our particular point of view, being in line with learnability theory, it confirms our basic motivation of adopting unsupervised PoS tagging for minority languages and, in general, for all those linguistic processing systems working with very limited manually tagged resources but huge unlabeled datasets. This situation is very common in Information Retrieval systems, and in all applications dealing with highly specialized domains (e.g. bioinformatics). In the future we plan to apply our technology to a Multilingual Knowledge Extraction scenario working on web scale corpora.

Acknowledgements

Alfio Gliozzo was supported by the FIRB-israel co-founded project N.RBIN045PXH. Claudio Giuliano was supported by the X-Media project (http://www.x-media-project.org), sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

6. References.

[1] M. Banko and E. Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In Proceedings of ACL-01, pp. 26-33, Toulouse, France

[2] C. Biemann. 2006. Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering. In Proceedings of the COLING/ACL-06 Student Research Workshop, Sydney, Australia

[3] T. Brants. 2000. TnT - a statistical part-of-speech tagger. In Proceedings of ANLP-2000, Seattle, USA

[4] P. F. Brown, V. J. Della Pietra, P. V. DeSouza, J. C. Lai and R. L. Mercer. 1992. Class-Based n-gram Models of Natural Language. Computational Linguistics 18(4), pp. 467-479

[5] A. Clark. 2003. Combining Distributional and Morphological Information for Part of Speech Induction, In Proceedings of EACL-03, Budapest, Hungary

[6] E. Charniak, C. Hendrickson, N. Jacobson and M. Perkowitz. 1993. Equations for part-of-speech tagging. In Proceedings of the 11th Natl. Conference on AI, pp. 784-789, Menlo Park

[7] S. Finch and N. Chater. 1992. Bootstrapping Syntactic Categories Using Statistical Methods. In Proc. 1st SHOE Workshop. Tilburg, The Netherlands

[8] D. Freitag. 2004a. Toward unsupervised whole-corpus tagging. In Proceedings of COLING-04, pp. 357-363, Geneva, Switzerland

[9] D. Freitag. 2004b. Trained named entity recognition using distributional clusters. In Proceedings of EMNLP 2004, pp. 262–269, Barcelona, Spain

[10] A. M. Gliozzo, C. Giuliano and C. Strapparava. 2005. Domain Kernels for Word Sense Disambiguation. In Proceedings of ACL-05, pp. 403-410, Ann Arbor, Michigan, USA

[11] J. Lafferty, A. K. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML-01, pages 282-289

[12] R. Mihalcea, T. Chklovsky and A. Kilgarriff. 2004. The Senseval-3 English lexical sample task. In Proceedings of Senseval-3, Barcelona, Spain.

[13] E. W. Noreen. 1989. Computer-Intensive Methods for testing Hypothesis. John Wiley & Sons, New York

[14] C. Peters. 2006. Working notes for the CLEF 2006 Workshop. Alicante, Spain

[15] U. Quasthoff, M. Richter and C. Biemann: Corpus Portal for Search in Monolingual Corpora. In Proceedings of LREC 2006, pp. 1799-1802, Genova, Italy

[16] D. Roth and A. van den Bosch. Editors. 2002. Proceedings of the Sixth CoNNL Workshop, Taipei, Taiwan

[17] H. Schütze. 1995. Distributional part-of-speech tagging. In Proceedings of EACL 7, pp. 141–148

[18] W. Skut, B. Krenn, T. Brants and H. Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In Proceedings of the ANLP-97. Washington, DC, USA

[19] E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal

[20] A. Ushioda. 1996. Hierarchical clustering of words and applications to NLP tasks. In Proceedings of the Fourth Workshop on Very Large Corpora, pp. 28-41, Somerset, NJ, USA