Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering

Chris Biemann
University of Leipzig, NLP Department
Augustusplatz 10/11, 04109 Leipzig, Germany
biem@informatik.uni-leipzig.de

Abstract

An unsupervised part-of-speech (POS) tagging system that relies on graph clustering methods is described. Unlike in current state-of-the-art approaches, the kind and number of different tags is generated by the method itself. We compute and merge two partitionings of word graphs: one based on context similarity of high frequency words, another on log-likelihood statistics for words of lower frequencies. Using the resulting word clusters as a lexicon, a Viterbi POS tagger is trained, which is refined by a morphological component. The approach is evaluated on three different languages by measuring agreement with existing taggers.

1 Introduction

1.1 Motivation

Assigning syntactic categories to words is an important pre-processing step for most NLP applications.

Essentially, two things are needed to construct a tagger: a lexicon that contains tags for words and a mechanism to assign tags to running words in a text. There are words whose tags depend on their use. Further, we also need to be able to tag previously unseen words. Lexical resources have to offer the possible tags, and our mechanism has to choose the appropriate tag based on the context.

Given a sufficient amount of manually tagged text, several approaches have demonstrated the ability to learn the instance of a tagging mechanism from manually labelled data and apply it successfully to unseen data. Those high-quality resources are typically unavailable for many languages and their creation is labour-intensive. We will describe an alternative needing much less human intervention.

In this work, steps are undertaken to derive a lexicon of syntactic categories from unstructured text without prior linguistic knowledge. We employ two different techniques, one for high- and medium frequency terms, one for medium- and low frequency terms. The categories will be used for the tagging of the same text where the categories were derived from. In this way, domain- or language-specific categories are automatically discovered.

1.2 Existing Approaches

There are a number of approaches to derive syntactic categories. All of them employ a syntactic version of Harris' distributional hypothesis: Words of similar parts of speech can be observed in the same syntactic contexts. Contexts in that sense are often restricted to the most frequent words. The words used to describe syntactic contexts will be called *feature words* in the remainder. *Target words*, as opposed to this, are the words that are to be grouped into syntactic clusters.

The general methodology (Finch and Chater, 1992; Schütze, 1995; inter al.) for inducing word class information can be outlined as follows:

1. Collect global context vectors for target words by counting how often feature words appear in neighbouring positions.
2. Apply a clustering algorithm on these vectors to obtain word classes

Throughout, feature words are the 150-250 words with the highest frequency. Contexts are the feature words appearing in the immediate neighbourhood of a word. The word's global context is the sum of all its contexts.

For clustering, a similarity measure has to be defined and a clustering algorithm has to be chosen. Finch and Chater (1992) use the Spearman Rank Correlation Coefficient and a hierarchical clustering, Schütze (1995) uses the cosine between vector angles and Buckshot clustering.

An extension to this generic scheme is presented in (Clark, 2003), where morphological

Proceedings of the COLING/ACL 2006 Student Research Workshop, pages 7–12,
Sydney, July 2006. ©2006 Association for Computational Linguistics

information is used for determining the word class of rare words. Freitag (2004) does not sum up the contexts of each word in a context vector, but the most frequent instances of four-word windows are used in a co-clustering algorithm.

Regarding syntactic ambiguity, most approaches do not deal with this issue while clustering, but try to resolve ambiguities at the later tagging stage.

A severe problem with most clustering algorithms is that they are parameterised by the number of clusters. As there are as many different word class schemes as tag sets, and the exact amount of word classes is not agreed upon intra- and interlingually, inputting the number of desired clusters beforehand is clearly a drawback. In that way, the clustering algorithm is forced to split coherent clusters or to join incompatible sub-clusters. In contrast, unsupervised part-of-speech induction means the induction of the tag set, which implies finding the number of classes in an unguided way.

1.3 Outline

This work constructs an unsupervised POS tagger from scratch. Input to our system is a considerable amount of unlabeled, monolingual text bar any POS information. In a first stage, we employ a clustering algorithm on distributional similarity, which groups a subset of the most frequent 10,000 words of a corpus into several hundred clusters (partitioning 1). Second, we use similarity scores on neighbouring co-occurrence profiles to obtain again several hundred clusters of medium- and low frequency words (partitioning 2). The combination of both partitionings yields a set of word forms belonging to the same derived syntactic category. To gain on text coverage, we add ambiguous high-frequency words that were discarded for partitioning 1 to the lexicon. Finally, we train a Viterbi tagger with this lexicon and augment it with an affix classifier for unknown words.

The resulting taggers are evaluated against outputs of supervised taggers for various languages.

2 Method

The method employed here follows the coarse methodology as described in the introduction, but differs from other works in several respects. Although we use 4-word context windows and the top frequency words as features (as in Schütze 1995), we transform the cosine similarity values between the vectors of our target words into a graph representation. Additionally, we provide a methdology to identify and incorporate POS-ambiguous words as well as low-frequency words into the lexicon.

2.1 The Graph-Based View

Let us consider a weighted, undirected graph *G(V,E)* ($v \in V$ vertices, $(v_i,v_j,w_{ij}) \in E$ edges with weights w_{ij}). Vertices represent entities (here: words); the weight of an edge between two vertices indicates their similarity.

As the data here is collected in feature vectors, the question arises why it should be transformed into a graph representation. The reason is, that graph-clustering algorithms such as e.g. (van Dongen, 2000; Biemann 2006), find the number of clusters automatically[1]. Further, outliers are handled naturally in that framework, as they are represented as singleton nodes (without edges) and can be excluded from the clustering. A threshold *s* on similarity serves as a parameter to influence the number of non-singleton nodes in the resulting graph.

For assigning classes, we use the Chinese Whispers (CW) graph-clustering algorithm, which has been proven useful in NLP applications as described in (Biemann 2006). It is time-linear with respect to the number of edges, making its application viable even for graphs with several million nodes and edges. Further, CW is parameter-free, operates locally and results in a partitioning of the graph, excluding singletons (i.e. nodes without edges).

2.2 Obtaining the lexicon

Partitioning 1: High and medium frequency words

Four steps are executed in order to obtain partitioning 1:

1. Determine 200 feature and 10.000 target words from frequency counts
2. construct graph from context statistics
3. Apply CW on graph.
4. Add the feature words not present in the partitioning as one-member clusters.

The graph construction in step 2 is conducted by adding an edge between two words a and b

[1] This is not an exclusive characteristic for graph clustering algorithms. However, the graph model deals with that naturally while other models usually build some meta-mechanism on top for determining the optimal number of clusters.

with weight w=1/(1-cos(a,b)), if w exceeds a similarity threshold *s*. The latter influences the number of words that actually end up in the graph and get clustered. It might be desired to cluster fewer words with higher confidence as opposed to running in the danger of joining two unrelated clusters because of too many ambiguous words that connect them.

After step 3, we already have a partition of a subset of our target words. The distinctions are normally more fine-grained than existing tag sets.

As feature words form the bulk of tokens in corpora, it is clearly desired to make sure that they appear in the final partitioning, although they might form word classes of their own[2]. This is done in step 4. We argue that assigning separate word classes for high frequency words is a more robust choice then trying to disambiguate them while tagging.

Lexicon size for partitioning 1 is limited by the computational complexity of step 2, which is time-quadratic in the number of target words. For adding words with lower frequencies, we pursue another strategy.

Partitioning 2: Medium and low frequency words

As noted in (Dunning, 1993), log-likelihood statistics are able to capture word bi-gram regularities. Given a word, its neighbouring co-occurrences as ranked by the log-likelihood reflect the typical immediate contexts of the word. Regarding the highest ranked neighbours as the profile of the word, it is possible to assign similarity scores between two words A and B according to how many neighbours they share, i.e. to what extent the profiles of A and B overlap. This directly induces a graph, which can be again clustered by CW.

This procedure is parametrised by a log-likelihood threshold and the minimum number of left and right neighbours A and B share in order to draw an edge between them in the resulting graph. For experiments, we chose a minimum log-likelihood of 3.84 (corresponding to statistical dependence on 5% level), and at least four shared neighbours of A and B on each side.

Only words with a frequency rank higher than 2,000 are taken into account. Again, we obtain several hundred clusters, mostly of open word classes. For computing partitioning 2, an efficient algorithm like CW is crucial: the graphs as used for the experiments consisted of 52,857/691,241 (English), 85,827/702,349 (Finnish) and 137,951/1,493,571 (German) nodes/edges.

The procedure to construct the graphs is faster than the method used for partitioning 1, as only words that share at least one neighbour have to be compared and therefore can handle more words with reasonable computing time.

Combination of partitionings 1 and 2

Now, we have two partitionings of two different, yet overlapping frequency bands. A large portion of these 8,000 words in the overlapping region is present in both partitionings. Again, we construct a graph, containing the clusters of both partitionings as nodes; weights of edges are the number of common elements, if at least two elements are shared. And again, CW is used to cluster this graph of clusters. This results in fewer clusters than before for the following reason: While the granularities of partitionings 1 and 2 are both high, they capture different aspects as they are obtained from different sources. Nodes of large clusters (which usually consist of open word classes) have many edges to the other partitioning's nodes, which in turn connect to yet other clusters of the same word class. Eventually, these clusters can be grouped into one.

Clusters that are not included in the graph of clusters are treated differently, depending on their origin: clusters of partition 1 are added to the result, as they are believed to contain important closed word class groups. Dropouts from partitioning 2 are left out, as they mostly consist of small, yet semantically motivated word sets. Combining both partitionings in this way, we arrive at about 200-500 clusters that will be further used as a lexicon for tagging.

Lexicon construction

A lexicon is constructed from the merged partitionings, which contains one possible tag (the cluster ID) per word. To increase text coverage, it is possible to include those words that dropped out in the distributional step for partitioning 1 into the lexicon. It is assumed that these words dropped out because of ambiguity. From a graph with a lower similarity threshold *s* (here: such that the graph contained 9,500 target words), we obtain the neighbourhoods of these words one at a time. The tags of those neighbours – if known – provide a distribution of possible tags for these words.

[2] This might even be desired, e.g. for English *not.*

2.3 Constructing the tagger

Unlike in supervised scenarios, our task is not to train a tagger model from a small corpus of hand-tagged data, but from our clusters of derived syntactic categories and a considerably large, yet unlabeled corpus.

Basic Trigram Model

We decided to use a simple trigram model without re-estimation techniques. Adopting a standard POS-tagging framework, we maximize the probability of the joint occurrence of tokens (t_i) and categories (c_i) for a sequence of length *n*:

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

The transition probability $P(c_i|c_{i-1},c_{i-2})$ is estimated from word trigrams in the corpus whose elements are all present in our lexicon.

The last term of the product, namely $P(c_i|t_i)$, is dependent on the lexicon[3]. If the lexicon does not contain (t_i), then (c_i) only depends on neighbouring categories. Words like these are called out-of-vocabulary (OOV) words.

Morphological Extension

Morphologically motivated add-ons are used e.g. in (Clark, 2003) and (Freitag 2004) to guess a more appropriate category distribution based on a word's suffix or its capitalization for OOV words. Here, we examine the effects of Compact Patricia Trie classifiers (CPT) trained on prefixes and suffixes. We use the implementation of (Witschel and Biemann, 2005). For OOV words, the category-wise product of both classifier's distributions serve as probabilities $P(c_i|t_i)$: Let $w=ab=cd$ be a word, *a* be the longest common prefix of *w* that can be found in all lexicon words, and *d* be the longest common suffix of *w* that can be found in all lexicon words. Then

$$P(c_i \mid w) = \frac{\left|\{u \mid u = ax \wedge \text{class}(u) = c_i\}\right|}{\left|\{u \mid u = ax\}\right|} \bullet \frac{\left|\{v \mid v = yd \wedge \text{class}(v) = c_i\}\right|}{\left|\{v \mid v = yd\}\right|}$$

CPTs do not only smoothly serve as a substitute lexicon component, they also realize capitalization, camel case and suffix endings naturally.

3 Evaluation methodology

We adopt the methodology of (Freitag 2004) and measure *cluster-conditional tag perplexity* PP as the average amount of uncertainty to predict the tags of a POS-tagged corpus, given the tagging with classes from the unsupervised method. Let

$$I_X = -\sum_x P(x) \ln P(x)$$

be the entropy of a random variable X and

$$M_{XY} = \sum_{xy} P(x,y) \ln \frac{P(x,y)}{P(x)P(y)}$$

be the mutual information between two random variables X and Y. Then the cluster-conditional tag perplexity for a gold-standard tagging T and a tagging resulting from clusters C is computed as

$$PP = \exp(I_{T|C}) = \exp(I_T - M_{TC}).$$

Minimum PP is 1.0, connoting a perfect congruence on gold standard tags.

In the experiment section we report PP on lexicon words and OOV words separately. The objective is to minimize the total PP.

4 Experiments

4.1 Corpora

For this study, we chose three corpora: the British National Corpus (BNC) for English, a 10 Million sentences newspaper corpus from Projekt Deutscher Wortschatz[4] for German, and 3 million sentences from a Finnish web corpus (from the same source). Table 1 summarizes some characteristics.

lang.	sent.	tok.	tagger	nr. tags	200 cov.	10K cov.
en	6M	100M	BNC[5]	84	55%	90%
fi	3M	43M	Connexor[6]	31	30%	60%
ger	10M	177M	(Schmid,1994)	54	49%	78%

Table 1: Characteristics of corpora: number of sentences, tokens, tagger and tagset size, corpus coverage of top 200 and 10,000 words.

Since a high coverage is reached with few words in English, a strategy that assigns only the most frequent words to sensible clusters will take us very far here. In the Finnish case, we can expect a high OOV rate, hampering performance

[3] Although (Charniak et al. 1993) report that using $P(t_i|c_i)$ instead leads to superior results in the supervised setting, we use the 'direct' lexicon probability. Note that our training material size is considerably larger than hand-labelled POS corpora.

[4] See http://corpora.informatik.uni-leipzig.de.

[5] Semi-automatic tags as provided by BNC.

[6] Thanks goes to www.connexor.com for an academic license; the tags do not include interpunctuation marks, which are treated seperately.

of strategies that cannot cope well with low frequency or unseen words.

4.2 Baselines

To put our results in perspective, we computed the following baselines on random samples of the same 1000 randomly chosen sentences that we used for evaluation:

- *1*: the trivial top clustering: all words are in the same cluster
- *200*: The most frequent 199 words form clusters of their own; all the rest is put into one cluster.
- *400*: same as 200, but with 399 most frequent words

Table 2 summarizes the baselines. We give PP figures as well as tag-conditional cluster perplexity PP_G (uncertainty to predict the clustering from the gold standard tags, inverse direction of PP):

lang	English			Finnish			German		
base	*1*	*200*	*400*	*1*	*200*	*400*	*1*	*200*	*400*
PP	29	3.6	3.1	20	6.1	5.3	19	3.4	2.9
PP_G	1.0	2.6	3.5	1.0	2.0	2.5	1.0	2.5	3.1

Table 2: Baselines for various tag set sizes

4.3 Results

We measured the quality of the resulting taggers for combinations of several substeps:

- **O:** Partitioning 1
- **M:** the CPT morphology extension
- **T:** merging partitioning 1 and 2
- **A:** adding ambiguous words to the lexicon

Figure 2 illustrates the influence of the similarity threshold *s* for O, OM and OMA for German – the other languages showed similar results. Varying *s* influences coverage on the 10,000 target words. When clustering very few words, tagging performance on these words reaches a PP as low as 1.25 but the high OOV rate impairs the total performance. Clustering too many words results in deterioration of results - most words end up in one big partition. In the medium ranges, higher coverage and lower known PP compensate each other, optimal total PPs were observed at target coverages 4,000-8,000. Adding ambiguous words results in a worse performance on lexicon words, yet improves overall performance, especially for high thresholds.

For all further experiments we fixed the threshold in a way that partitioning 1 consisted of 5,000 words, so only half of the top 10,000 words are considered unambiguous. At this value, we found the best performance averaged over all corpora.

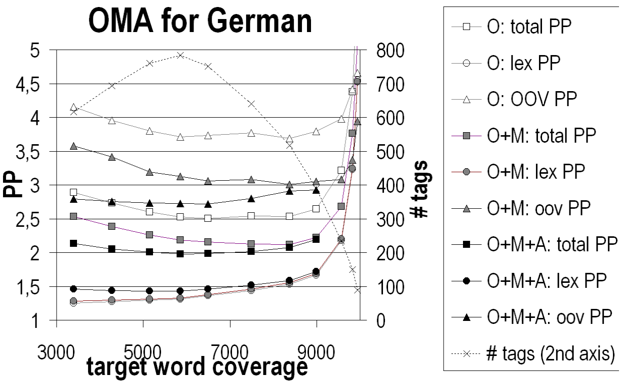

Fig 2. Influence of threshold s on tagger performance: cluster-conditional tag perplexity PP as a function of target word coverage.

lang		O	OM	OMA	TM	TMA
EN	total	2.66	2.43	2.08	2.27	2.05
	lex	1.25		1.51	1.58	1.83
	oov	6.74	6.70	5.82	9.89	7.64
	oov%	28.07		14.25	14.98	4.62
	tags	619			345	
FI	total	4.91	3.96	3.79	3.36	3.22
	lex	1.60		2.04	1.99	2.29
	oov	8.58	7.90	7.05	7.54	6.94
	oov%	47.52		36.31	32.01	23.80
	tags	625			466	
GER	total	2.53	2.18	1.98	1.84	1.79
	lex	1.32		1.43	1.51	1.57
	oov	3.71	3.12	2.73	2.97	2.57
	oov%	31.34		23.60	19.12	13.80
	tags	781			440	

Table 3: results for English, Finnish, German. oov% is the fraction of non-lexicon words.

Overall results are presented in table 3. The combined strategy TMA reaches the lowest PP for all languages. The morphology extension (M) always improves the OOV scores. Adding ambiguous words (A) hurts the lexicon performance, but largely reduces the OOV rate, which in turn leads to better overall performance. Combining both partitionings (T) does not always decrease the total PP a lot, but lowers the number of tags significantly. Finnish figures are generally worse than for the other languages, akin to higher baselines.

The high OOV perplexities for English in experiment TM and TMA can be explained as follows: The smaller the OOV rate gets, the more likely it is that the corresponding words were also OOV in the gold standard tagger. A remedy would be to evaluate on hand-tagged data. Differences between languages are most obvious when comparing OMA and TM: whereas for English it pays off much more to add ambiguous words than to merge the two partitionings, it is the other way around in the German and Finnish experiments.

To wrap up: all steps undertaken improve the performance, yet their influence's strength varies. As a flavour of our system's output, consider the example in table 4 that has been tagged by our English TMA model: as in the introductory example, "saw" is disambiguated correctly.

Word	**cluster ID**	**cluster members (size)**
I	166	I (1)
saw	2	*past tense verbs* (3818)
the	73	a, an, the (3)
man	1	*nouns* (17418)
with	13	*prepositions* (143)
a	73	a, an, the (3)
saw	1	*nouns* (17418)
.	116	. ! ? (3)

Table 4: Tagging example

We compare our results to (Freitag, 2004), as most other works use different evaluation techniques that are only indirectly measuring what we try to optimize here. Unfortunately, (Freitag 2004) does not provide a total PP score for his 200 tags. He experiments with an hand-tagged, clean English corpus we did not have access to (the Penn Treebank). Freitag reports a PP for known words of 1.57 for the top 5,000 words (91% corpus coverage, baseline 1 at 23.6), a PP for unknown words without morphological extension of 4.8. Using morphological features the unknown PP score is lowered to 4.0. When augmenting the lexicon with low frequency words via their distributional characteristics, a PP as low as 2.9 is obtained for the remaining 9% of tokens. His methodology, however, does not allow for class ambiguity in the lexicon, the low number of OOV words is handled by a Hidden Markov Model.

5 Conclusion and further work

We presented a graph-based approach to unsupervised POS tagging. To our knowledge, this is the first attempt to leave the decision on tag granularity to the tagger. We supported the claim of language-independence by validating the output of our system against supervised systems in three languages.

The system is not very sensitive to parameter changes: the number of feature words, the frequency cutoffs, the log-likelihood threshold and all other parameters did not change overall performance considerably when altered in reasonable limits. In this way it was possbile to arrive at a one-size-fits-all configuration that allows the parameter-free unsupervised tagging of large corpora.

To really judge the benefit of an unsupervised tagging system, it should be evaluated in an application-based way. Ideally, the application should tell us the granularity of our tagger: e.g. semantic class learners could greatly benefit from the high-granular word sets arising in both of our partitionings, which we endeavoured to lump into a coarser tagset here.

References

C. Biemann. 2006. *Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems*. Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-06, New York, USA

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

A. Clark. 2003. *Combining Distributional and Morphological Information for Part of Speech Induction*, Proceedings of EACL-03

T. Dunning. 1993. *Accurate Methods for the Statistics of Surprise and Coincidence*, Computational Linguistics 19(1), pp. 61-74

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

D. Freitag. 2004. *Toward unsupervised whole-corpus tagging*. Proc. of COLING-04, Geneva, 357-363.

H. Schmid. 1994. *Probabilistic Part-of-Speech Tagging Using Decision Trees*. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK, pp. 44-49

H. Schütze. 1995. *Distributional part-of-speech tagging*. In EACL 7, pages 141–148

S. van Dongen. 2000. *A cluster algorithm for graphs*. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

F. Witschel, and C. Biemann. 2005. *Rigorous dimensionality reduction through linguistically motivated feature selection for text categorisation*. Proc. of NODALIDA 2005, Joensuu, Finland