Res on Lang and Comput (2009) 7:101–135
DOI 10.1007/s11168-010-9067-9

Unsupervised Part-of-Speech Tagging in the Large

Chris Biemann

Published online: 29 June 2010

Abstract Syntactic preprocessing is a step that is widely used in NLP applications. Traditionally, rule-based or statistical Part-of-Speech (POS) taggers are employed that either need considerable rule development times or a sufficient amount of manually labeled data. To alleviate this acquisition bottleneck and to enable preprocessing for minority languages and specialized domains, a method is presented that constructs a statistical syntactic tagger model from a large amount of unlabeled text data. The method presented here is called unsupervised POS-tagging, as its application results in corpus annotation in a comparable way to what POS-taggers provide. Nevertheless, its application results in slightly different categories as opposed to what is assumed by a linguistically motivated POS-tagger. These differences hamper evaluation procedures that compare the output of the unsupervised POS-tagger to a tagging with a supervised tagger. To measure the extent to which unsupervised POS-tagging can contribute in application-based settings, the system is evaluated in supervised POS-tagging, word sense disambiguation, named entity recognition and chunking. Unsupervised POS-tagging has been explored since the beginning of the 1990s. Unlike in previous approaches, the kind and number of different tags is here generated by the method itself. Another difference to other methods is that not all words above a certain frequency rank get assigned a tag, but the method is allowed to exclude words from the clustering, if their distribution does not match closely enough with other words. The lexicon size is considerably larger than in previous approaches, resulting in a lower out-of-vocabulary (OOV) rate and in a more consistent tagging. The system presented here is available for download as open-source software along with tagger models for several languages, so the contributions of this work can be easily incorporated into other applications.

C. Biemann (✉)
Microsoft Corporation / Powerset, 475 Brannan St. #330, San Francisco, CA 94110, USA
e-mail: cbiemann@microsoft.com

Keywords Unsupervised and knowledge-free NLP · Induction of parts-of-speech · Application-based evaluation · Graph clustering

1 Introduction to Unsupervised POS-Tagging

Assigning syntactic categories to words is an important pre-processing step for most NLP applications. POS-tags are used for parsing, chunking, anaphora resolution, named entity recognition and information extraction, just to name a few.

Essentially, two things are needed to construct a tagger: a lexicon that contains tags for words and a mechanism to assign tags to tokens in a text. For some words, the tags depend on their use, e.g. in "I saw the man with a saw". It is also necessary to handle previously unseen words. Lexical resources have to offer the possible tags, and a mechanism has to choose the appropriate tag based on the context, in order to produce annotation like this: "I/PNP saw/VVD the/AT0 man/NN1 with/PRP a/AT0 saw/NN1. /PUN".[1]

Given a sufficient amount of manually tagged text, two approaches have demonstrated the ability to learn the instance of a tagging mechanism from labelled data and apply it successfully to unseen data. The first is the rule-based approach (Brill 1992), where transformation rules are constructed that operate on a tag sequence delivered by the lexicon. The second approach is statistical, for example HMM-taggers (Charniak et al. 1993, inter al.) or taggers based on conditional random fields (Lafferty et al. 2001). Both approaches employ supervised learning and therefore need manually tagged training data. Those high-quality resources are typically unavailable for many languages and their creation is labour-intensive. Even for languages with rich resources like English, tagger performance breaks down on noisy input. Texts of a different genre than the training material may also create problems, e.g. e-mails as opposed to newswire or literature. It is, in general, not viable to annotate texts for all these cases.

Here, an alternative needing much less human intervention is described. Steps are undertaken to derive a lexicon of syntactic categories from unstructured text following the Structure Discovery paradigm (Biemann 2007), which strives at finding and annotating regularities of language using unsupervised and knowledge-free procedures. Hence, it is not possible to aim at exact correspondence with linguistically motivated tagsets, but for obvious reasons: even for the same language, linguistically motivated tagsets differ considerably, as it was measured for various tagsets for English by Clark (2003).

Two different techniques are employed here, one for high-and medium frequency words, another for medium- and low frequency words. The categories will be used for the tagging of the same text the categories were derived from. In this way, domain- or language-specific categories are automatically discovered. Extracting syntactic categories for text processing from the texts to be processed fits the obtained structures neatly and directly to them, which is not possible using general-purpose resources.

[1] in this tagset (Garside et al. 1987), PNP stands for personal pronoun, VVD is full verb, AT0 is determiner is singular or plural, NN1 is singular noun, PRP is Preposition, PUN is punctuation.

With moving POS tagging to a data-driven, unsupervised step that can serve as feature-based input for subsequent steps, a major step in alleviating the acquisition bottleneck can be taken. The motivation behind this work is primarily to lower the amount of work that goes into manual annotation or the creation of rule sets; on a larger perspective, however, it can also unveil principles of language structure in such as common features and differences between languages are mirrored in the way the data arranges itself for different languages.

This article is organised as follows. After discussing related work in Sect. 2, the Chinese Whispers graph clustering algorithm is described in Sect. 3, which is used here as a means to perform necessary abstractions and generalisations for grouping words into POS-classes. Section 4 lays out the steps undertaken to arrive at an unsupervised POS-tagger in detail. The quality of the tagger output is assessed in two ways: Sect. 5 performs a comparison between standard tagsets and the unsupervised variant for three typologically different languages. Further, influence of different system components, corpus size and domain shifting are examined and the tagger is compared to another word clustering system. In Sect. 6, the tagger's annotations are used as features in Machine Learning for various NLP tasks, evaluating its contributions in an application-based way. Section 7 concludes and provides a way how to obtain the system and a number of tagger models.

2 Related Work

There are a number of approaches to derive syntactic categories. All of them employ a syntactic version of Harris' distributional hypothesis (Harris 1968): words of similar parts of speech can be observed in the same syntactic contexts. Measuring to what extent two words appear in similar contexts measures their similarity, cf. (Miller and Charles 1991). As function words form the syntactic skeleton of a language and almost exclusively contribute to the most frequent words in a corpus, contexts in that sense are often restricted to the most frequent words. The words used to describe syntactic contexts are further called *feature words*. *Target words*, as opposed to this, are the words that are to be grouped into syntactic clusters. Note that usually, the feature words form a subset of the target words.

The general methodology (Finch and Chater 1992; Schütze 1993, 1995; Gauch and Futrelle 1994; Clark 2000; Rapp 2005) for inducing word class information can be outlined as follows:

1. Collect global context vectors of 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. Some authors employ a much larger number of features and reduce the dimensions of the resulting matrix using Singular Value Decomposition (Schütze 1993; Rapp 2005). The choice of high frequency words as features is motivated by Zipf's law: these few stop words constitute the bulk of tokens in a corpus. Pruning context features to these allows efficient implementations without considerably losing on coverage. Contexts are the feature words appearing in the immediate neighbourhood of a word. The word's global

Table 1 Corpus and context vectors for 6 feature words and a context window of size 4. The feature vectors of different positions are concatenated

```
... COMMA sagte der Sprecher bei der Sitzung FULLSTOP
... COMMA rief der Vorsitzende in der Sitzung FULLSTOP
... COMMA warf in die Tasche aus der Ecke FULLSTOP
```

Features: der(1), die(2), bei(3), in(4), FULLSTOP (5), COMMA (6)

Position	−2				−1						+1						+2	
Target/feature	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2
sagte										1	1							
rief										1	1							
warf										1				1				1
Sprecher					1								1				1	
Vorsitzende					1									1			1	
Tasche		1				1											1	
Sitzung	1	1			2										2			
Ecke					1										1			

context is the sum of all its contexts. Table 1 illustrates the collection of contexts for a German toy example.

The clustering step is defined by a similarity measure and a clustering algorithm. Finch and Chater (1992) use the Spearman Rank Correlation Coefficient and a hierarchical clustering, (Schütze 1993, 1995) uses the cosine between vector angles and Buckshot clustering, (Gauch and Futrelle 1994) use cosine on Mutual Information vectors for hierarchical agglomerative clustering and (Clark 2000) applies Kullback-Leibler divergence in his CDC algorithm.

An extension to this generic scheme is presented in (Clark 2003), where morphological information is used for determining the word class of rare words. Further, clustering is driven by the likelihood of an HMM model with a fixed number of states. Freitag (2004a) does not sum up the contexts of each word in a context vector, but uses the most frequent instances of four-word windows in a co-clustering algorithm (Dhillon et al. 2003): rows and columns (here words and contexts) are clustered simultaneously. Two-step clustering is undertaken by Schütze (1993): clusters from the first step are used as features in the second step.

The number of target words in the clustering differ from 1,000 target words in a 2,00,000 token corpus (Gauch and Futrelle 1994) over 5,000 target words (Finch and Chater 1992; Freitag 2004a) to all 47,025 words in the Brown Corpus in (Schütze 1995). Clark (2000) uses 12 million tokens as input; (Finch and Chater 1992) operate on 40 million tokens.

Evaluation methodologies differ considerably amongst the papers discussed here. Finch and Chater (1992) inspect their clusters manually, Rapp (2005) performs flawlessly in sorting 50 medium frequency words into nouns, verbs and adjectives. Schütze (1995) presents precision and recall figures for a reduced tagset, excluding rare and non-English-word tags from the evaluation. More recent approaches (Clark 2000, 2003; Freitag 2004a) employ information-theoretic measures, see Sect. 5. Regarding syntactic ambiguity, most approaches do not deal with this issue while clustering, but try to resolve ambiguities at the later tagging stage, if at all.

As the virtue of unsupervised POS-tagging lies in its possible application to all natural languages or domain-specific subsets, it is surprising that in most previous works, only experiments with English are reported. An exception is (Clark 2003), who additionally uses languages of the Slavonic, Finno-Ugric and Romance families.

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 tagsets, and the exact amount of word classes is not agreed upon intra- and interlingually, having to specify the number of desired clusters a-priori 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 tagset, which implies finding the number of classes in an unguided way.

Another alternative which operates on a predefined tagset is presented by Haghighi and Klein (2006): in this semi-supervised framework, only three words per tag have to be provided to induce a POS-tagger for English with 80% accuracy. The amount of data the authors use in their experiments is rather small (8,000 sentences), but their computationally expensive methods—gradient-based search to optimise Markov Random Field parameters—does not allow for substantially more input data. This issue is also shared with Baysian approaches as conducted in Goldwater and Griffiths (2007).

3 Chinese Whispers Graph Clustering

Chinese Whispers (CW, Biemann 2006) is a very basic—yet effective—algorithm to partition the nodes of weighted, undirected graphs. It is motivated by the eponymous children's game, where children whisper words to each other. This game is also known as "telephone" in American English. While the game's goal is to arrive at some funny derivative of the original message by passing it through several noisy channels, the CW algorithm aims at finding groups of nodes that broadcast the same message to their neighbors. The algorithm is outlined as follows:

Algorithm 1 Standard Chinese Whispers CW(graph $G(V, E)$)

```
for all $v_i \in V$ do
    $class(v_i) = i$
end for
for it=1 to number-of-iterations do
    for all $v \in V$, randomised order do
        $class(v)$=predominant class in $neigh(v)$
    end for
end for
return partition $P$ induced by class labels
```

Regions of the same class stabilise during the iteration and grow until they reach the border of a stable region of another class. Notice that classes are updated continuously:

a vertex can obtain classes from the neighbourhood that were introduced there in the same iteration. The fractions of a class a in the neighbourhood $neigh(v)$ of a vertex v with $ew(v, w)$ edge weight of edge vw is computed as

$$fraction(a, v) = \frac{\sum_{w \in neigh(v), class(w) = a} ew(v, w)}{\sum_{w \in neigh(v)} ew(v, w)},$$

the predominant class a in the neighbourhood of v is given by

$$\arg\max_{a} fraction(a, v).$$

For each class label in the neighbourhood, the sum of the weights of the edges to the vertex in question is taken as score for ranking. Intuitively, the algorithm works as follows in a bottom-up fashion: First, all nodes get different classes. Then the nodes are processed for a small number of iterations and inherit the strongest class in the local neighborhood. This is the class whose sum of edge weights to the current node is maximal. In case of multiple strongest classes, one is chosen randomly. Regions of the same class stabilize during the iteration and grow until they reach the border of a stable region of another class.

The CW algorithm cannot cross component boundaries (between unconnected subgraphs), because there are no edges between nodes belonging to different components. Further, nodes that are not connected by any edge are discarded from the clustering process, which possibly leaves a portion of nodes unclustered. In all graphs tested, almost no changes were observed after 10–20 iterations.

The result of CW is a hard partitioning of the given graph into a number of partitions that emerges in the process—CW itself is parameter-free. It is possible to obtain a soft partitioning by assigning a class distribution to each node, based on the weighted distribution of (hard) classes in its neighborhood in a final step. The outcomes of CW resemble those of Min-Cut, see (Wu and Leahy 1993): Dense regions in the graph are grouped into one cluster while sparsely connected regions are separated. In contrast to Min-Cut, CW does not find an optimal hierarchical clustering but yields a non-hierarchical (flat) partition. Furthermore, it does not require any threshold as input parameter and is more efficient. Parametrization is, however, realized by the way the graph is built—this entails the similarity function between nodes and a threshold on minimal edge weight.

CW can be compared to the Potts (1952) Model in statistical mechanics, which models phase transitions of spins in chrystalline lattices. The lattice is replaced by a graph in CW, where coupling strength is given by edge weights. As opposed to the approaches in statistical mechanics, CW does not attempt to optimally solve the model, but approximates a solution by aggressively propagating only a single label at a time. This leads to nondeterministic behavior, which has been observed to lead to different clusterings for different runs. However, in the case of weighted scale-free small world graphs as present in the data we operate on, variations are miniscule and do not severely influence the outcome, cf. (Biemann 2007).

With its run-time linear in the number of edges, Chinese Whispers belongs to the class of graph partitioning algorithms at the lower bound of computational complexity: at least, the graph itself has to be taken into account when attempting to partition it, and the list of edges is the most compact form of its representation. This allows the clustering of very large graphs and is a crucial feature for large lexicon sizes in unsupervised POS-tagging.

4 Unsupervised POS-System

What follows is the description of the construction of the unsupervised POS-tagger from scratch. Input to the system is a considerable amount of unlabelled, tokenised monolingual text without any POS information. In a first stage, Chinese Whispers is applied to distributional similarity data, which groups a subset of the most frequent 10,000 words of a corpus into several hundred clusters (tagset 1). Second, similarity scores on neighbouring co-occurrence profiles are used to obtain again several hundred clusters of medium- and low frequency words (tagset 2). The combination of both partitions yields sets of word forms belonging to the same induced syntactic category. To gain on text coverage, ambiguous high-frequency words that were discarded for tagset 1 are added to the lexicon. Finally, a Viterbi trigram tagger is trained with this lexicon and augmented with an affix classifier for unknown words.

Figure 1 depicts the process of unsupervised POS-tagging from unlabelled to fully labelled text. The details of the method will be outlined in the following subsections.

The method employed here follows the coarse methodology as described in the Sect. 2, but differs from other works in several respects. Although four-word context

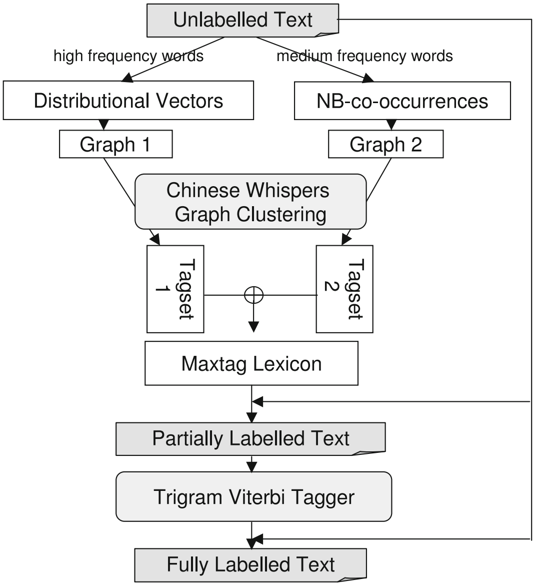


Fig. 1 Diagram of the process of unsupervised POS-tagging, from unlabelled over partially labelled to fully labelled text

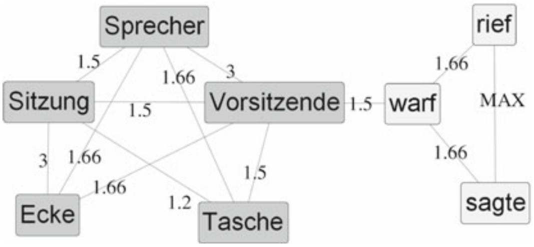


Fig. 2 Graph for the data given in Table 1 and its partition into nouns and verbs

windows and the top frequency words as features are used as in (Schütze 1995), the cosine similarity values between the vectors of target words are transformed into a graph representation in order to be able to cluster them with Chinese Whispers graph clustering, see Section 3. Additionally, a method to identify and incorporate POS-ambiguous words as well as low-frequency words into the lexicon is provided.

4.1 Tagset 1: High and Medium Frequency Words

Four steps are executed in order to obtain tagset 1 for high- and medium frequency words from a text corpus.

1. Determine 10,000 target and 200 feature words from frequency counts
2. Collect context statistics and construct graph
3. Apply Chinese Whispers on graph
4. Add the feature words not present in the partition as one-member clusters.

The graph construction in step 2 is conducted by adding an edge between two words a and b with weight[2] $w = 1/(1 - cos(\vec{a}, \vec{b}))$, computed using the feature vectors $\vec{a}$ and $\vec{b}$ (cf. Table 1) of words a and b. The edge is only drawn 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 the risk of joining two unrelated clusters because of too many ambiguous words that connect them. After step 3, there is already a partition of a subset of target words that can be perceived as tagset. Figure 2 shows the weighted graph and its CW-partition for the example given in Table 1. The number of target words is limited by computational considerations: since the feature vectors have to be compared in a pair-wise fashion, a considerably higher number of target words results in long run times. The number of feature words was examined in preliminary experiments, showing only minor differences with respect to cluster quality in the range of 100–300.

As noted e.g. in (Schütze 1995), the clusters are motivated syntactically as well as semantically and several clusters per open word class can be observed. The distinctions are normally finer-grained than existing tagsets, as Fig. 3 illustrates.

[2] Cosine similarity is a standard measure for POS induction, however, other measures would be possible.

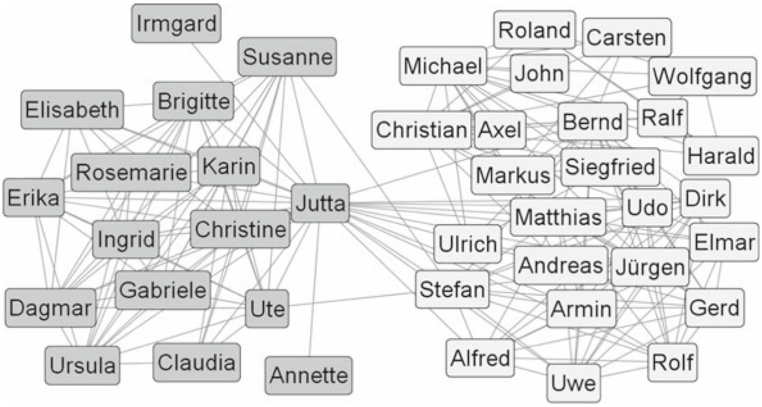


Fig. 3 Fine-grained distinctions: female and male first names from German corpus. Note that German lexicalizes gender in titles and professions, which makes it possible to learn this distinction. The figure shows only a local neighbourhood of the graph for tagset 1

Since the feature words form the bulk of tokens in the corpus, it is clearly desired to make sure that they appear in the tagset, although they might end up in clusters with only one element. This might even be desired, e.g. for English *'not'*, which usually has its own POS-tag in linguistic tagsets. This is done in step 4, where assigning separate word classes for high frequency words is considered to be a more robust choice than trying to disambiguate them while tagging. Starting from this, it is possible to map all words contained in a cluster onto one feature and iterate this process, replacing feature words by the clusters obtained, cf. (Schütze 1993). In that way, higher counts in the feature vectors are obtained, which could provide a better basis for similarity statistics. Experiments conducting this showed, however, that only marginal improvements could be reached, as text coverage is not substantially increased.

Table 2 shows as illustration a selection of clusters for the British National Corpus (BNC,[3] Burnard 1995). Several clusters for nouns can be observed. Evaluating lexical clusters against a gold standard may lead to inconclusive results, because the granularities of the gold standard and the clusters usually differ, e.g. English singular and plural nouns end up in one cluster, but first and last names are distinguished. The evaluation scores are largely depending on the tagset used for gold standard. Here, an information-theoretic measure is employed that allows an intuitive interpretation: Entropy precision (EP) measures the extent to which the gold standard classification is reconstructable from the clustering result. EP directly relates to the precision measure in information retrieval. Its counterpart, recall as the number of retrieved vs. the total number of instances relates to the coverage on target words as reached by the clustering algorithm. For the gold standard, each word gets assigned its most frequent tag, ignoring POS-ambiguities. Despite all these disadvantages, EP provides a means to relatively compare the quality of partitions for varying thresholds s.

[3] http://.natcorp.ox.ac.uk/ [April 1st, 2007].

Table 2 Selected clusters from the BNC clustering for setting *s* such that the partition contains 5,000 words. In total, 464 clusters are obtained. EP for this partition is 0.8276. Gold standard tags have been gathered from the BNC, sample words are presented in decreasing order of frequency

Rank	Size	Gold standard tags (count)	Description	Sample words
1	662	NN1(588), NN2(44)	Singular nouns	day, government, world, system, company, house, family
2	401	NN1(311), NN2(86)	Singular nouns	part, end, state, development, members, question, policy, ...
3	292	NN2(284), NN1(7)	Plural nouns	men, services, groups, companies, systems, schools, ...
4	254	NP0(252), NN1(2)	First names	John, David, Peter, Paul, George, James, Michael, ...
5	247	AJ0(233), NN1(9)	Adjectives	social, political, real, economic, national, human, private, ...
6	220	NN1(148), NN2(63)	Singular and plural nouns	business, water, service, staff, land, training, management, ...
7	209	VVI(209)	Verbs	get, make, take, give, keep, provide, play, move, leave, ...
8	195	AJ0(118), NN1(25)	Adjectives (country)	British, police, New, European, individual, National, ...
9	110	NP0(109), NN1(1)	Last names	Smith, Jones, Brown, Wilson, Lewis, Taylor, Williams, ...
10	92	AJ0(89), CRD(1)	Adjectives (size/quality)	new, good, little, few, small, great, large, major, big, special
11	73	AJ0(73)	Adjectives (animate)	heavy, beautiful, quiet, soft, bright, charming, cruel, ...
12	67	NN2(67)	Plural nouns	problems, conditions, costs, issues, activities, lines, ...
12	67	NP0(66), NN1(1)	Countries	England, Scotland, France, Germany, America, Ireland, ...
16	57	NP0(57)	Cities	Oxford, Edinburgh, Liverpool, Manchester, Leeds, Glasgow, ...
22	39	AV0(39)	Sentence beginning	Well, However, Thus, Indeed, Also, Finally, Nevertheless, ...
25	30	NN2(30)	Plural professions	teachers, managers, farmers, governments, employers, ...
34	17	CRD(17)	Numbers	three, four, five, six, ten, eight, seven, nine, twelve, fifteen, ...
65	6	NP0(6)	Titles	Mr, Mrs, Dr, Miss, Aunt, Ms
217	2	AT0(2)	Indefinite determiner	a, an
217	2	NP0(2)	Location 1st	Saudi, Sri
217	2	VVZ, VVD	To wear	wore, wears
217	2	VVZ, VVD	To insist	insisted, insists

Definition Entropy Precision (EP): Let $G = G_1, \ldots G_m$ be the gold standard classification and $C = C_1, \ldots C_p$ be the clustering result. Then, EP is computed as follows:

$$EP(C, G) = \frac{M_{CG}}{I_G}$$

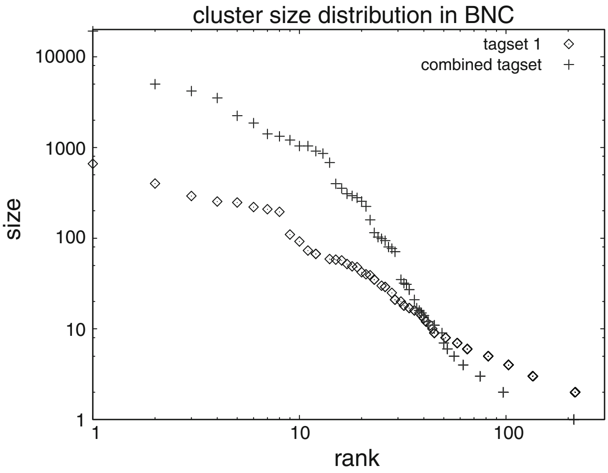


Fig. 4 Cluster size distribution for tagset 1 and combined tagset (see Sect. 4.3) in the BNC, ordered decreasingly by cluster size (rank). Note that while the distribution of cluster sizes in tagset 1 looks like a power law distribution, cluster sizes in the combined tagset exhibit much larger clusters (due to adding more words) and a steeper slope for mid-range sized clusters

with mutual information M_{XY} between X and Y

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

and I_X entropy of X.

$$I_X = -\sum_{x} P(x) ln P(x)$$

A maximal EP of 1 is reached by a trivial clustering of singleton clusters. This does not impose a severe problem, considering the typical cluster size distribution as depicted in Fig. 4: there is a substantial amount of words to be found in large clusters. Nevertheless, optimising EP promotes a large number of small clusters, which is why the number of clusters has to be provided along with the EP figures to give an impression of the result's quality. A minimal EP of 0 indicates statistical independence of C and G.

For evaluation of tagset 1, three corpora of different languages were chosen: 10 million sentences of German tagged with 52 tags using TreeTagger (Schmid 1994), the 6 million sentences of BNC for English, pretagged semi-automatically with the CLAWS tagset of 84 tags (Garside et al. 1987) and 1 million sentences from a Norwegian web corpus tagged with the Oslo-Bergen tagger (Hagen et al. 2000), using a simplified tagset of 66 tags. These corpora are automatically tagged, yet we use them as gold standards for optimizing parameters, arguing that automatic high precision taggers serve well for this purpose. Note that we use the result from this evaluation not to tune the unsupervised tagger to a particular dataset or language, but rather

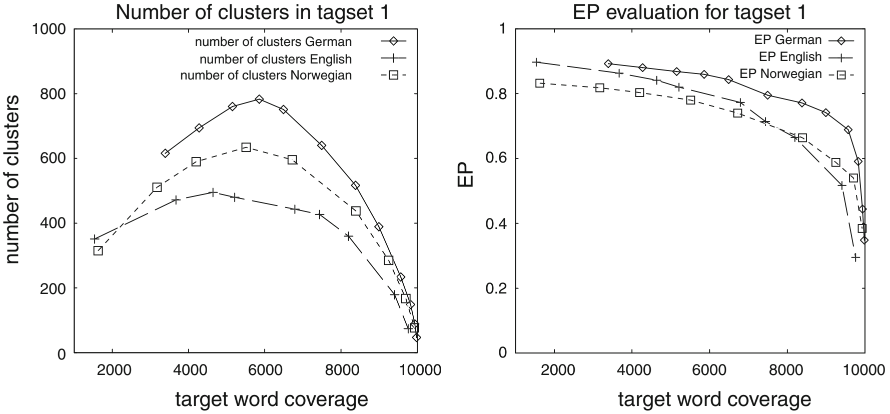


Fig. 5 Tagset size and Entropy precision dependent on number of included target words for tagset 1

derive general settings for its parametriation. Figure 5 gives the EP results for varying numbers of target words included in the partition and the number of clusters.

From Fig. 5 it is possible to observe that EP remains stable for a wide range of target word coverage between about 2,000–9,000 words. The number of parts is maximal for the medium range of coverage: at higher coverage, POS-ambiguous words that are related to several clusters serve as bridges. If too many links are established between two clusters, CW will collapse both into one cluster, possibly at cost of EP. At lower coverage, many classes are left out. This evaluation indicates the language-independence of the method, as results are qualitatively similar for all languages tested.

As indicated above, lexicon size for tagset 1 is limited by the computational complexity of step 2, which is time-quadratic in the number of target words. Due to the non-sparseness of context vectors of high-frequency words there is not much room for optimisation. In order to add words with lower frequencies, another strategy is pursued.

4.2 Tagset 2: Medium and Low Frequency Words

As noted in (Dunning 1993), log likelihood statistics capture word bigram regularities. Given a word, its neighbouring co-occurrences as ranked by their log likelihood ratio are 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. The hypothesis here is that words sharing many neighbours should usually be observed with the same part-of-speech. For the acquisition of word classes in tagset 2, the second-order graph on neighbouring co-occurrences is used. To set up the graph, a co-occurrence calculation is performed which yields word pairs based on their significant co-occurrence as immediate neighbours. Here, all word pairs exceeding a log likelihood threshold of 1.00 (corresponding to a positive correlation, yet the outcome is robust in a wide threshold range) enter this bipartite graph.

Note that if similar words occur in both parts, they form two distinct vertices. Only words with a frequency rank higher than 2,000 are taken into account: as preliminary experiments revealed, high-frequency words of closed word classes spread over the clusters, resulting in deteriorated tagging performance later, so they are excluded in this step.

This graph is transformed into a second-order graph by comparing the number of common right and left neighbours for two words. The similarity (edge weight) between two words is the sum the number of common neighbours on both sides. Figure 6 depicts the significant neighbouring graph, the second-order graph derived from it, and its CW-partition. The word-class-ambiguous word ’drink’ (to drink the drink) is responsible for all inter-cluster edges. In the example provided in Figure 6, three clusters are obtained that correspond to different parts-of-speech. For computing the similarities based on the significant neighbour-based word co-occurrence graphs for both directions, at maximum the 150 most significant co-occurrences per word are considered, which regulates the density of the graph and leads to improvements in run-time.

To test this on a large scale, the second-order similarity graph for the BNC was computed, excluding the most frequent 2,000 words and drawing edges between words only if they shared at least two left and four right common neighbours. The clusters are checked against a lexicon that contains the most frequent tag for each word in the BNC. The largest clusters are presented in Table 3, together with the predominant tags in the BNC.

In total, CW produced 282 clusters, of which 26 exceed a size of 100. The weighted average of cluster purity (i.e. the number of predominant tags divided by cluster size) was measured at 88.8%, which exceeds significantly the precision of 53% on word type as reported by Schütze (1995).

Again, several hundred clusters, mostly of open word classes are obtained. For computing tagset 2, an efficient algorithm like CW is crucial: the graphs as used for the experiments consist typically of 10,000 to 100,000 vertices and about 100,000 to 1 million edges. Note that for the construction of graph 2, it is not necessary to do a pairwise comparison between all nodes. Rather, the neighbourhood co-occurrence

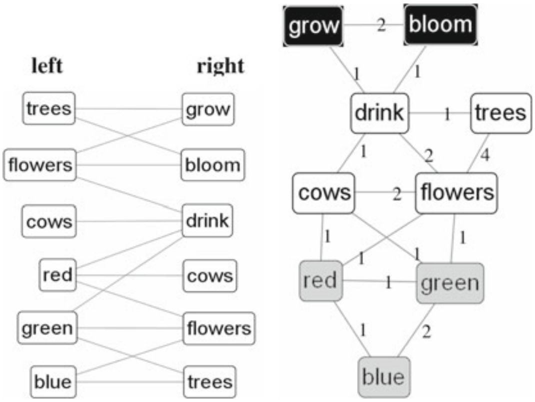


Fig. 6 *Left* Bi-partite neighbouring co-occurrence graph. *Right* second-order graph on neighbouring co-occurrences clustered with CW, as used for graph 2

Table 3 The largest clusters of tagset 2 for the BNC

Size	Tags (count)	Sample words
18432	NN(17120), AJ(631)	secret, officials, transport, unemployment, farm, county, wood, procedure, grounds, ...
4916	AJ(4208), V(343)	busy, grey, tiny, thin, sufficient, attractive, vital, ...
4192	V(3784), AJ(286)	filled, revealed, experienced, learned, pushed, occurred, ...
3515	NP(3198), NN(255)	White, Green, Jones, Hill, Brown, Lee, Lewis, Young, ...
2211	NP(1980), NN(174)	Ian, Alan, Martin, Tony, Prince, Chris, Brian, Harry, Andrew, Christ, Steve, ...
1855	NP(1670), NN(148)	Central, Leeds, Manchester, Australia, Yorkshire, Belfast, Glasgow, Middlesbrough, ...

statstics select only a relatively small subset of possible pairs, keeping the construction time of the graph feasible and the complexity sub-quadratic.

4.3 Combination of Tagsets 1 and 2

Now, there are two tagsets of two different, yet overlapping frequency bands. A large portion of these 8,000 words in the overlapping region is present in both tagsets. Again, a graph is constructed, containing the clusters of both tagsets as vertices; weights of edges represent the number of common elements, if at least two elements are shared. Notice that the graph is bipartite.

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 tagsets 1 and 2 are both high, they capture different aspects as they are obtained from different sources. Vertices of large clusters (which usually consist of open word classes) have many edges to the other partition's vertices, 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 tagset 1 are added to the result, as they are believed to contain important closed word class groups. Dropouts from tagset 2 are simply left out, as they mostly consist of small, yet semantically motivated word sets. The total loss of words by disregarding these many, but small clusters did never exceed 10% in any experiment. Figure 7 illustrates this combination process.

Conducting this combination yields about 300–600 clusters that will be further used as a lexicon for tagging. As opposed to the observations made in Schütze (1995), only a handful of clusters are found per open word class, of which most distinctions are syntactically motivated, e.g. adjectives with different case markers. For unsupervised POS-tagging, the aim is to arrive at a low number of clusters to mimic the supervised counterparts. A more rigid method to arrive at yet less clusters would be to leave out classes of low corpus frequency.

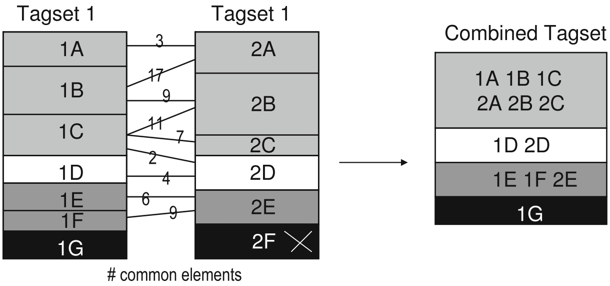
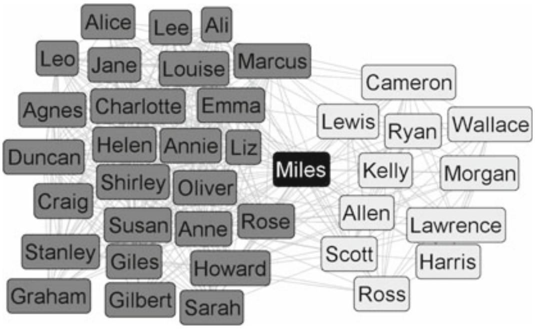


Fig. 7 Combination process: tagsets 1 and 2 are related via the number of common elements in their respective clusters. Shades symbolise the outcome of Chinese Whispers on this graph of clusters. Clusters marked with x are not included in the resulting graph of clusters

Fig. 8 POS-disambiguation in the BNC for 'Miles' as first and last name. Note that most of the last names are ambiguous themselves, causing 'Miles' to be similar to them



4.4 Setting up the Tagger

4.4.1 Lexicon Construction

From the merged tagsets, a lexicon is constructed that 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 tagset 1 into the lexicon. It is assumed that some of these words could not be assigned to any cluster because of ambiguity. From a graph with a lower similarity threshold s (here: such that the graph contains 9,500 target words), neighbourhoods of these words are obtained one at a time. This is comparable to the methodology in (Ertöz, et al. 2002), where only some vertices are used for clustering and the rest is assimilated. Here, the added target words are not assigned to only one cluster: the tags of their neighbours—if known—provide a distribution of possible tags for these words. Figure 8 gives an example: the name 'Miles' (frequency rank 8,297 in the BNC) is rated 65% as belonging to a first name cluster and 35% as last name.

4.4.2 Constructing the Tagger

Unlike in supervised scenarios, the task is not to train a tagger model from a small corpus of hand-tagged data, but from the clusters of derived syntactic categories and a large, yet unlabelled corpus. This realises a class-based N-gram model (Brown et al. 1992).

Here, a simple trigram Viterbi model without re-estimation techniques (such as Baum-Welch) is employed in order not to blur the quality of lexicon construction and to speed up processing. Adapting a previous standard POS-tagging framework (Charniak et al. 1993), the probability of the joint occurrence of tokens t_i and categories c_i for a sequence of length n is maximised. At this, the tagger does not only guess the categories of OOV tokens, but also assigns the most appropriate category for words that are recorded with several possibilities in the lexicon, cf. Sect. 4.4.1. We emphasize that the resulting tagger can be applied to unseen text (with unseen types) and has the capability to disambiguate word classes based on context.

4.4.3 Morphological Extension

A main performance flaw of supervised POS-taggers originates from OOV words. Morphologically motivated add-ons are used e.g. in (Clark 2003; Freitag 2004a) to guess a more appropriate category distribution based on a word's suffix or its capitalisation. Here, Compact Patricia Trie classifiers (CPT, see Knuth 1998) trained on prefixes and suffixes are employed. 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 and any lexicon word, and d be the longest common suffix of w and any lexicon words. Then

$$P(c_i|w) = \frac{|\{u|u = ax \wedge class(u) = c_i\}|}{|\{u|u = ax\}|} \cdot \frac{|\{v|v = yd \wedge class(v) = c_i\}|}{|\{v|v = yd\}|}.$$

CPTs do not only serve as a substitute lexicon component, they also handle capitalisation, camelCase and suffix endings without having to define features explicitly or setting length or maximal number thresholds (as in Freitag 2004a for suffixes). A similar technique is employed by Cucerzan and Yarowsky (1999) in the context of named entity recognition. The CPT implementation is further used in supervised settings by Witschel and Biemann (2005) for compound splitting and in (Eiken et al. 2006) for base form reduction, where it is described in more detail.

5 Direct Evaluation of Tagging

For directly measuring the quality of our unsupervised POS-tagger, we adopt the methodology of Freitag (2004a). We measure the 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. For the same corpus tagged with two methods, the measure indicates how well one tagging can be reproduced from the other. Let

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

Table 4 Characteristics of corpora for POS induction evaluation: number of sentences, number of tokens, tagger and tagset size, corpus coverage of top 200 and 10,000 words

Language	Sent.	Tokens	Tagger	nr.tags	200 cov. (%)	10K cov. (%)
English	6M	100M	BNC	84	55	90
Finnish	3M	43M	Connexor[a]	31	30	60
German	10M	177M	Schmid (1994)	54	49	78

[a] Thanks goes to Connexor Oy, Helsinki, for an academic licence of their Finnish MBT tagger

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 with gold standard tags. Below, PP on lexicon words and OOV words is reported separately. The objective is to minimise the total PP.

Unsupervised POS-tagging is meant for yet untagged text, so a system should be robustly performing on a variety of typologically different languages. For evaluating tagging performance, three corpora are chosen: the BNC for English, a 10 million sentences newspaper corpus from the Leipzig Corpora Collection (LCC,[4] Quasthoff et al. 2006) for German, and 3 million sentences from LCC's Finnish web corpus. Table 4 summarises some characteristics.

Since a high text coverage is reached with only a few words in English, a strategy that assigns only the most frequent words to sensible clusters already ensures satisfactory performance. In the Finnish case, a high OOV rate can be expected, hampering performance of strategies that cannot cope well with low frequency or unseen words.

To put the results in perspective, the following baselines on random samples of the same 1,000 randomly chosen sentences used for evaluation were computed:

- 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 5 summarises the baselines in terms of PP. Additional baselines were included for comparison with system results.

[4] http://corpora.uni-leipzig.de/ [December 1st, 2007].

Table 5 Baselines for various tagset sizes

English					
Baseline	1	200	345	400	619
PP	29.3	3.69	3.17	3.03	2.53
Finnish					
Baseline	1	200	400	466	625
PP	20.2	6.14	5.58	5.46	5.23
German					
Baseline	1	200	400	440	781
PP	18.24	3.32	2.79	2.73	2.46

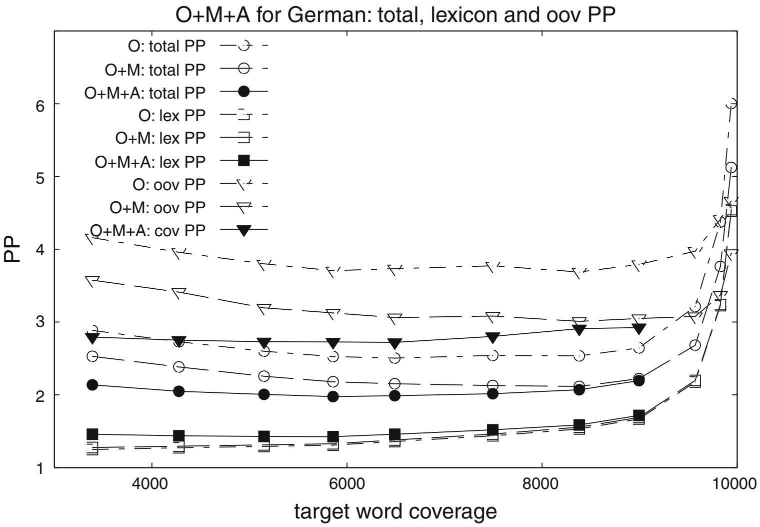


Fig. 9 Influence of threshold s on tagger performance: cluster-conditional tag perplexity PP as a function of target word coverage for tagset 1

5.1 Influence of System Components

The quality of the resulting taggers for combinations of several sub-steps is measured using:

- O: tagset 1
- M: the CPT morphology extension
- T: merged tagsets 1 and 2
- A: adding ambiguous words to the lexicon

Figure 9 illustrates the influence of the similarity threshold s for O, O+M and O+M+A for German—for other languages, results look qualitatively similar. Varying s influences coverage on the 10,000 target words. When clustering on 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 part. In the medium ranges, higher coverage

Table 6 Results in PP for English, Finnish, German. OOV% is the fraction of non-lexicon words in terms of tokens

Lang	Words	O	O+M	O+M+A	T+M	T+M+A
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	

and lower known PP compensate each other, optimal total PPs were observed at target word coverages of 4,000–8,000. The system's performance is stable with respect to changing thresholds, as long as it is set in reasonable ranges. Adding ambiguous words results in a worse performance on lexicon words, yet improves overall performance, especially for high thresholds.

For all further experiments, the threshold s was fixed in a way that tagset 1 consisted of 5,000 words, so only half of the top 10,000 words are considered unambiguous. At this value, the best performance throughout all corpora tested was achieved.

Overall results are presented in Table 6. The combined strategy T+M+A 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 partitions (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, consistent with higher baselines. Differences between languages are most obvious when comparing O+M+A and T+M: whereas for English it pays off much more to add ambiguous words than to merge the two partitions, it is the other way around in the German and Finnish experiments.

To sum up the discussion of results: all introduced steps improve the performance, yet their influence's strength varies. As a sample of the system's output, consider the example in Table 7 that has been tagged by the English T+M+A model: as in the example above, 'saw' is disambiguated correctly. Further, the determiner cluster is complete; unfortunately, the pronoun 'I' constitutes a singleton cluster.

Table 7 Tagging example

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)

The results can be compared to (Freitag 2004a); most other work uses different evaluation techniques that are only indirectly measuring what is tried to optimise here. Unfortunately, (Freitag 2004a) does not provide a total PP score for his 200 tags. He experiments with a hand-tagged, clean English corpus that is not free (the Penn Treebank) and is therefore not an option here. Freitag reports a PP for known words of 1.57 for the top 5,000 words (91% corpus coverage, baseline 1 at 23.6: his corpus seems 'easier' for both the baseline and with respect to OOV rates), 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 trained with Baum-Welch-Reestimation. Due to different evaluation tagsets and a different baseline, it is hard to assess whether Freitag's method performs better or worse on information-theoretic measures on English. For most other languages with flatter frequency distributions, Freitag's method can be expected to perform worse because of higher OOV rates resulting from the 5,000 word limit.

5.2 Influence of Parameters

A number of parameters for the process of unsupervised POS-tagging were introduced at the points where they arised. Now, all parameters are listed for recapitulating the possibilities to fine-tune the method. Table 8 gives the parameters, a short explanation, and the default setting used in all experiments.

Their influence and interplay is outlined as follows. $FEAT$ did not show to have a large influence in ranges 100–300. It might be adviseable to use higher values for languages with low Zipfian exponents (such as Finnish) to gain higher text coverage for building tagset 1. When processing small corpora, $TARG$ should not be too high, because a low corpus frequency for target words results in unreliable context statistics. The parameter $CWTARG$ must be set smaller than $TARG$, Fig. 9 indicates that 40–80% of $TARG$ is a sensible range. Higher settings result in more words that can overlap for combining the two tagsets.

NB_SIG and NB_THRESH go hand in hand to regulate the density of the graph for tagset 2: Lower significance thresholds lead to more edges in the neighbouring

Table 8 Parameters, default settings and explanation

Parameter	Default	Explanation
$FEAT$	200	Number of feature words for tagset 1 similarities
$TARG$	10,000	Number of target words for tagset 1 similarities
$CWTARG$	5,000	Number of words that are clustered for tagset 1 amongst $TARG$ words, by applying an appropriate similarity threshold s on the graph
$TOPADD$	9,500	Number of words that are considered for adding ambiguous words amonst $TARG$ words, by applying an appropriate similarity threshold s on the graph
NB_SIG	1.00	Significance threshold for neighbour-based co-occurrences
NB_THRESH	2	Minimum number of common neighbour-based co-occurrences per side for constituting an edge in the graph for tagset 2
NB_MAX	150	Maximum neighbouring co-occurrences per word to consider for the second-order graph of tagset 2
$CONF\_OVERLAP$	2	Minimum number of common words for connecting partitions in the graph of clusters to merge tagset 1 and 2
$BEHEAD$	2,000	Minimum rank of words to enter the graph for tagset 2
$SING\_ADD$	200	Maximum frequency rank of words to add as singletons, if not already contained in the combined tagset

co-occurrence graph, higher values for NB_THRESH prune edges in the graph for tagset 2. The maximum number of neighbouring co-occurrences per word NB_MAX influences the density of the graph for tagset 2, lower settings result in less edges per word. All experiments were carried out with the default value, however, higher values lead to more coarse-grained tagsets that e.g. join common and proper nouns. Different settings could prove advantageous for different applications, but no experiments were conducted to measure to what extent.

$BEHEAD$ should be set in a way that stop words are excluded from tagset 2, but considerably lower than $TARG$, to enable sufficient overlap between the two tagsets. Less than a value of 2 for $CONF\_OVERLAP$ can result in spurious combinations in the graph of clusters, higher values reduce the lexicon size since clusters from tagset 2 are more likely to be excluded.

Adding more singletons amongst the $SING\_ADD$ most frequent words increases the number of tags, but also the number of trigrams available for training the Viterbi tagger.

A sensible extension would be to limit the total number of tags by excluding these clusters from the combined tagset that have the lowest corpus frequency, i.e. the sum of frequencies of the lexicon words constituting this tag.

5.3 Influence of Corpus Size

Having determined a generic setting for the interplay of the different system components (M+T+A), the influence of corpus size on the cluster-conditional tag perplexity

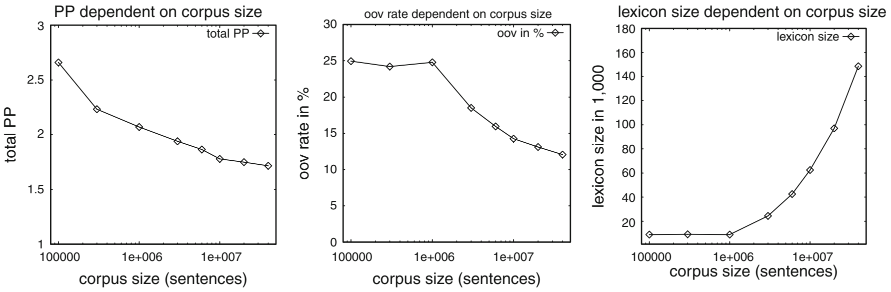


Fig. 10 PP, OOV rate and lexicon size versus corpus size for German

PP shall be examined now. For this purpose, taggers were induced from German corpora of varying size from 1,00,000 sentences up to 40 million sentences, taken from LCC. Evaluation was carried out by measuring PP between the resulting taggers and the hand-tagged German NEGRA corpus (Brants et al. 1997), testing on all 20,000 sentences of NEGRA. The evaluation corpus was not part of the corpora used for tagger induction. Figure 10 provides total PP, the OOV rate and the lexicon size, dependent on corpus size.

Not surprisingly, the more corpus is provided for tagger induction, the better performance levels are reached in terms of PP. The more data provided, the more reliable the statistics for tagset 1, which is reflected in tremendous PP improvement from using 1,00,000 to 10,00,000 sentences. In this range, tagset 2 is almost empty and does not contribute to the lexicon size, which is mirrored in a constant OOV rate for this range. Above 1 million sentences, the size of tagset 2 increases, resulting in lower PP and OOV rates. The lexicon size explodes to some 1,00,000 entries for a corpus size of 40 million sentences. Summarizing the results obtained by training the unsupervised POS-tagger on corpora of various sizes, there always seems to be room for improvements by simply adding more data. However, improvements beyond 10 million sentences are small in terms of PP.

The interpretation of the PP measure is difficult, as it largely depends on the gold standard. While it is possible to relatively compare the performance of different components of a system or different systems along these lines, it only gives a poor impression on the utility of the unsupervised tagger's output. Therefore, several application-based evaluations are undertaken in Sect. 6. But before that, we discuss domain shifting and make an attempt to compare this method with the clustering described in Clark (2003).

5.4 Domain Shifting

Now we shed some light in the advantage of using unsupervised POS tags for domain adaptation. When cultivating an NLP system for one domain or genre and then applying it to a different domain, a major drop in performance can be expected due to different vocabulary and different constructions, cf. (Hal and Marcu 2006).

Table 9 OOV rates for unsupervised POS models for BNC and MEDLINE, both in-domain and cross-domain. For reference, also OOV rates with respect to the most frequent 10 K words per corpus are given

	BNC model (%)	BNC top 10 K (%)	MEDLINE model (%)	MEDLINE top 10 K (%)
BNC OOV	7.1	8.6	12.0	20.7
MEDLINE OOV	18.8	21.9	5.0	9.5

It would be desirable to train on one domain and test performance on the other, using various ways to incorporate our unsupervised POS tags, e.g. inducing separate models or a single model on a mixed corpus and measure the contribution to a task. Apart from corpora from different domains, this would require gold standard data for the same task in different domains, which we unfortunately did not have available. This is why we revert to quantitative observations, accompanied by exemplifying data, from contrasting a general-domain corpus of British English (the BNC as used above) with a specialized medical domain corpus of mixed-spelling English (the 2004 MeSH abstracts,[5] henceforth called MEDLINE).

In Table 9, OOV rates for two unsupervised POS models trained on our corpora are given. It is clearly observable that changing domain results in higher OOV rates. Applying a model trained on the specialized domain MEDLINE corpus to the general-domain BNC leads to almost twice the OOV rate. Applying the general domain BNC model to the specialized domain—a more common scenario in domain adaptation—results in a more than 3-fold increase of the OOV rate from 5 to 18.8%.

While OOV rates only reveal how much vocabulary is covered when shifting domain, it does not reveal the utility of the tagset. Tagsets for specialized corpora often reflect their domain: in addition to the core word classes, additional word classes are discovered that could help for certain applications. In Table 10, domain-specific clusters for MEDLINE are given. For example the tag for units can help to project from generally used units like *kg* or *gallons* to specialized units like *kcal/mol*. The tag for cell lines or viruses might facilitate Information Extraction tasks.

5.5 Comparison with Clark 2003

This section aims at comparing the unsupervised tagger to the unsupervised word clustering described in (Clark 2003).[6]

Clark's system as used here for a comparison consists of a clustering that is inspired by the well-known k-means algorithm. All words above a certain frequency threshold t are clustered into k clusters by iteratively improving the likelihood of a given clustering. The likelihood is measured with respect to a combination of a

[5] http://www.nlm.nih.gov/mesh/filelist.html [Jan 2010].

[6] Thanks goes to Alexander Clark for making his clustering software available for download http://www.cs.rhul.ac.uk/home/alexc/pos2.tar.gz [Jan 2010].

Table 10 Selected clusters from the MEDLINE clustering (final tagset) with randomly selected words. In total, 479 clusters were obtained, which were ordered by decreasing size to rank them. These clusters reflect domain specific word classes that are usually not found in general-domain corpora

Rank	Size	Description	Sample words
10	1707	Cell lines	F98, ANA-1, HUC, NTERA2, Caco-2BBe, AT5BIVA, YH, TIG-1, EG2+, LNCaP-FGC, IEC-6, Raw264.7, spleen, RKO, H292, HT29, BCE, SRG, MLE-15, S16, Mer+, SP-ir, C-21, SW1990, Caki-2, HT-1080, HT29-Cl.16E, SK-N-SH, MH-S, Haller, MES-SA, CA46, NFS-60, MN9D, MCTC, ...
19	705	Viruses	APEC, SIVsm, salmonella, herpes, IKC, virus-2, dengue, Hib, Adv, Pnc, pneumococcal, TMEV, anthrax, THO, BVDV-1, NDV, dengue-2, TCLA, HGV, SV-40, toxoplasma, heartworm, WNV, YF, diphtheria, Ara-, IFV, MLV, cryptococcus, ...
27	474	Units	kg, g/l, Hz, hm2, mo, CFU, mm2/s, mm2, dL, hrs, U/ml, min)-1, microV, IU/ml, Pounds, kcal/mol, cm3/ min, g/ min, microg/ml, metre, PDLs, ml/g, centimetres, gallons, euros, mumol/l, pg, months, nanometres, pmoles, MPa, cm2, MJ/d, bp, francs, IU/L, U/m2/day, g/d, g/kg/ min, mol, cfu/cm2, ...
39	213	Time relative to treatment	postsurgery, post-transplant, postload, postfracture, gestation, ago, 12-months, post-release, post-test, postop, postcoitus, postchallenge, posttrauma, postdose, post-insult, postadmission, poststroke, postburn, postresuscitation, post-stress, postpartal, post-challenge, regraft, postnatal, EGA, postinsemination, postaxotomy, post-hatching, posthemorrhage, postmenstrual, ...

bigram class model (Ney et al. 1994) for distributional evidence, a letter Hidden Markov Model for modeling morphological similarities and a frequency prior. The parameter k can be used to set the desired granularity of the clustering.

In contrast to the method described in this paper, all words of a given corpus are clustered: words with frequency of t or less are simply clustered together in one large cluster. This means that there are no OOV words in the corpus w.r.t. the clustering. The method as it stands does not allow assigning tags for unknown words (in new text) from the context alone. Also, the same word always gets assigned the same tag, there is no mechanism that accounts for ambiguity.

Due to the computational cost of the clustering, which increases for larger k, we could not successfully run the methods on corpora larger than the BNC, which took up to a week in CPU time for large k. In contrast, the method presented here induces a model for the BNC in a few hours. We used the default settings ($t = 5$, 10 iterations) to produce clusterings for varying k for the BNC and a German corpus of 5M sentences, assigning cluster IDs as tags for the same evaluation corpus used in Sect. 5 above.

The V-measure is the harmonic mean of two measures: Homogeneity h and Completeness c. h quantifies how well the system assignment can be mapped to the gold standard, not penalising too fine-grained system clusters. This measure is very closely related to EP as defined above. The symmetrically defined c quantifies how well

Table 11 V-measure evaluation for German and English: Baselines, Clark's system (with number of clusters) and this system for the German 40M sentence corpus and the BNC

System	Completeness	Homogeneity	V-measure
German			
Base-1	1	0	0
Base-200	0.6096	0.58626	0.5977
Base-440	0.5798	0.6536	0.6145
Base-781	0.5539	0.6878	0.6136
Clark-128	0.5481	0.8407	0.6636
Clark-256	0.5221	0.8941	0.6592
Clark-440	0.4982	0.9079	0.6434
Unsupos-440	0.5670	0.8604	**0.6835**
English BNC			
Base-1	1	0	0
Base-200	0.6988	0.6172	0.6555
Base-345	0.6697	0.6612	0.6654
Base-400	0.6633	0.6739	0.6686
Clark-128	0.6228	0.8204	**0.7081**
Clark-256	0.5841	0.8554	0.6942
Clark-345	0.5727	0.8708	0.6910
Unsupos-345	0.6349	0.766	0.6943

Best V-measure per corpus marked in bold

the gold standard can be mapped to the system assignment, at this penalising fine-grained system distinctions. Both h and c are normalised and take on values between 0 (undesirable) and 1 (desirable), serving as analogies to Precision and Recall.

Table 11 shows the V-measure for the German and English data for different k and different baselines in comparison with the M+T+A models described above.

From Table 11, it becomes clear that the V-measure is relatively independent for a wide range of the number of clusters. Baseline scores demonstrate the tradeoff between h and c. Clark's system generally produces higher h and lower c than the system presented here for the same number of clusters. This points at a different cluster size distribution: Clark's cluster size distribution on the token level is flatter. For German, the system presented here outperforms Clark's system for all ks tested on the V-measure. For English, Clark's system with 128 clusters shows a higher performance. Overall, the systems exhibit a similar performance, with the system presented here being more expressive (with regard to ambiguous words) and more flexible (with respect to unseen types).

6 Application-Based Evaluation

POS-taggers are a standard component in any applied NLP system. In this section, a number of NLP tasks are viewed as machine learning problems: the POS-tagger component provides some of the features that are used to learn a function that assigns a label to unseen examples, characterised by the same set of features as the examples used for training. In this setting, it is straightforward to evaluate the contribution of

POS-taggers—be they supervised or unsupervised—by providing the different POS-tagging annotations to the learning algorithm or not.

Having advanced machine learning algorithms at hand that automatically perform feature weighting and selection, the standard approach to NLP systems is to provide all possible features and to leave the choice to the learning algorithm.

The task-performing systems for application-based evaluation were chosen to cover two different machine learning paradigms: kernel methods in a word sense disambiguation (WSD) system and Conditional Random Fields (CRFs, see Lafferty et al. 2001) for supervised POS, named entity recognition (NER) and chunking. Some results of this section have been previously published in (Biemann et al. 2007).

All evaluation results are compared in a pair-wise fashion using the approximate randomisation procedure of Noreen (1989) as significance test, for which p-values as error probabilities are given, i.e. a significant difference with $p < 0.01$ means that the test is more than 99% sure that the difference has not been caused by chance.

6.1 Unsupervised POS for Supervised POS

It might seem contradictory to evaluate an unsupervised POS tagger in terms of the contribution it can make to supervised POS tagging. While there exist high precision supervised POS taggers and elaborate feature sets have been worked out (see Toutanova et al. 2003 for state-of-the art POS tagging on the Penn Treebank), it does not seem necessary to create an unsupervised tagger in presence of training data. This, however, changes if one looks at different domains or languages. In these settings, any method that can help reduce the amount of training data is a contribution to development speed and cost of natural language processing systems.

In this section, we examine the contribution of the unsupervised tagger as a feature in supervised POS tagging. We show that the unsupervised tags capture structural regularities beyond standard features such as capitalization and affixes.

As a machine learning algorithm we use first-order Conditional Random Fields, leveraging the CRF++ implementation.[7]

In order to test the contribution of unsupervised POS features, we set up 4 different systems and compare their performance:

- System A: Only lexical features, time-shifted by −2, −1, 0, 1, 2
- System B: Like System A. Additionally, features time-shifted by −1, 0, 1 for capitalization, number, 2-letter prefix and 2-letter suffix
- System C: Like System A. Additionally, unsupervised labels time-shifted by −2, −1, 0, 1, 2 as assigned by the tagger model induced on 40 million sentences from the Wortschatz project as evaluated in Fig. 10
- System D: Combination of all features present in systems A, B and C.

Training sets of varying sizes are selected randomly from the 20,000 sentences of the hand-tagged NEGRA corpus for German, the respective remainders are used for evaluation. Results are reported in tagging accuracy (number of correctly assigned tags

[7] Available at http://crfpp.sourceforge.net/ [version 0.53].

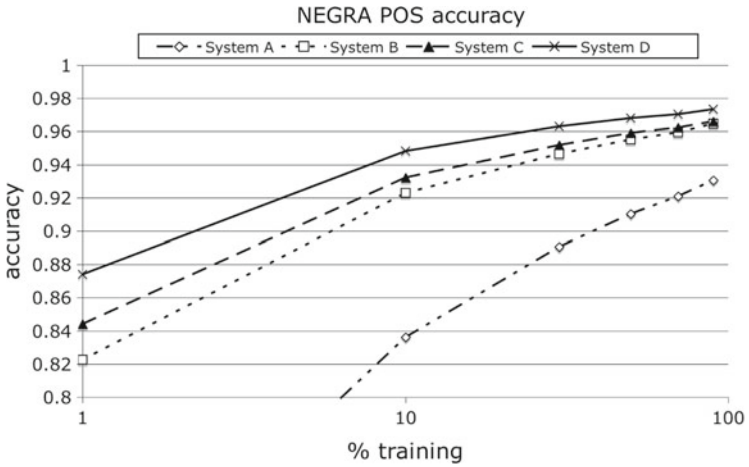


Fig. 11 Learning curve for supervised POS-tagging with and without using unsupervised POS features (accuracy)

%training	1	10	30	50	70	90
System A	0.6527	0.8362	0.8907	0.9103	0.9210	0.9305
System B	0.8227	0.9229	0.9464	0.9550	0.9595	0.9647
System C	0.8440	0.9323	0.9517	0.9590	0.9626	0.9660
System D	0.8739	0.9481	0.9630	0.9680	0.9706	0.9733

divided by total number of tokens), averaged over three different splits per training size each. Figure 11 shows the learning curve.

Results indicate that supervised tagging can clearly benefit from unsupervised tags: between 30% and 50% training with unsupervised tags, the performance on 90% training without the unsupervised extension is surpassed comparing systems D and B. At 90% training, error rate reduction of system D over B is 24.3%, 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. Comparing the gains of systems B, C and D over system A, we conclude that the unsupervised features provide more than simple capitalization, number of affix features, since their combination D significantly outperforms systems B and C.

Probably also due to a more advanced machine learning paradigm, system D with its 0.9733 accuracy compares favourably to the performance of (Brants 2000), who reports an accuracy of 0.967 at 90% training on the same data set—equal to the performance of system C. To our knowledge, system D presented here constitutes state-of-the art for German POS tagging.

When swapping the unsupervised POS features in system D with the 128 clusters from Clark's method—the *k* for which the best V-measure were obtained in Sect. 5.5—we measured equal performance of 0.9733 on the same splits for 90% training. Combining unsupervised features and Clark's features, precision is further improved to 0.9743. Since the improvements of the single unsupervised features do not wipe out each other, it can be concluded that the two clustering methods capture somewhat different aspects of syntactic similarity.

6.2 Unsupervised POS for Word Sense Disambiguation

The task in word sense disambiguation (WSD) is to assign the correct word sense to ambiguous words in a text based on the context. The senses are provided by a sense inventory (usually WordNet, Miller et al. 1990). Supervised WSD is trained on examples where the correct sense is provided manually, and tested by comparing the system's outcome on held-out examples.

In the WSD literature, many algorithms have been proposed, characterised by different feature sets and classification algorithms. The state of the art supervised WSD methodology, reporting the best results in most of the Senseval-3 lexical sample tasks (Mihalcea et al. 2004) in different languages, is based on a combination of syntagmatic and domain kernels (Gliozzo 2005) in a Support Vector Machine classification framework.

A great advantage of this methodology is that all its pre-processing steps are also unsupervised and knowledge-free and therefore comply to the SD paradigm. It is shown here that the only language-dependent component in the system of (Gliozzo et al. 2005)—a supervised POS-tagger—can safely be replaced by the unsupervised POS-tagger.

Kernel WSD basically encompasses two different aspects of similarity: 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.

The domain kernel handles domain aspects of similarity among two texts based on the Domain Model as introduced in (Gliozzo 2005), which is a soft clustering of terms reflecting semantic domains. On the other hand, syntagmatic aspects are probably the most important evidence while recognising sense similarity. In general, the strategy adapted to model syntagmatic relations in WSD is to provide bigrams and trigrams of collocated words as features to describe local contexts (Yarowsky 1994). The main drawback of this approach is that non-contiguous or shifted collocations cannot be identified, decreasing the generalisation power of the learning algorithm.

The syntagmatic kernel allows estimating the number of common non-continuous subsequences of lemmas (i.e. collocations) between two examples, in order to capture syntagmatic similarity. Analogously, the POS kernel is defined to operate on sequences of parts-of-speech. The syntagmatic kernel is given by a linear combination of the collocation kernel and the POS kernel.

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. 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 as in (Gliozzo et al. 2005) is tested against replacing the original POS kernel with the

Table 12 Comparative evaluation on Senseval scores for WSD. All differences are not significant at $p < 0.1$

System	Only POS	Full
No POS	N/A	0.717
Supervised POS	0.629	0.733
Unsupervised POS	0.633	0.735

unsupervised POS kernel and omitting the POS kernel completely. Table 12 summarises the results in terms of Senseval scores for WSD, tested on the lexical sample task for English. The unsupervised POS annotation was created using the BNC tagger model, see Sect. 5.

Results show that POS information is generally contributing to a very small extent to WSD accuracy in the full WSD 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 with the same system using supervised POS-tags. In conclusion, supervised tagging can safely be exchanged in kernel WSD with the unsupervised variant. Replacing the only pre-processing step that is dependent on manual resources in the system of (Gliozzo et al. 2005), state of the art supervised WSD is proven to not being dependent on any linguistic pre-processing at all.

Gains in using an unsupervised tagger for WSD can probably be attributed to the finer distinctions the unsupervised tagger makes. E.g. a separate tag for professions can help to generalize over this category. While it is arguable whether this distinction should be part of a standard POS tagset, since this is rather a semantic than a syntactic restriction, it is desirable from the point of view of this application.

6.3 Unsupervised POS for NER and Chunking

Named entity recognition (NER) is the task of finding and classifying named entities, such as persons, organisations and locations. Chunking is concerned with shallow syntactic annotation; here, words in a text are labelled as being syntactically correlated, e.g. in noun phrases, verb phrases and prepositional phrases. For performing NER and chunking, these applications are perceived as a tagging task: in each case, labels from a training set are learned and applied to unseen examples. In the NER task, these labels mark named entities and non-named entities, in the chunking task, the respective phrases or chunks are labelled.

For both tasks, the MALLET tagger (McCallum 2002) is trained. It is based on first-order Conditional Random Fields (CRFs), which define a conditional probability distribution over label sequences given a particular observation sequence. The flexibility of CRFs to include arbitrary, non-independent features makes it easy to supply either standard POS-tags, unsupervised POS-tags or no POS-tags to the system without changing its overall architecture.

The tagger operates on a different set of features for the two tasks. In the NER system, the following features are accessible, time-shifted by $-2, -1, 0, 1, 2$:

- the word itself
- its POS-tag

Table 13 Comparative evaluation of NER on the Dutch CoNLL-2002 dataset in terms of F1 for PERson, ORGanisation, LOCation, MISCellaneous, ALL. No differences are significant with $p < 0.1$

Category	PER	ORG	LOC	MISC	ALL
No POS	0.8084	0.7445	0.8151	0.7462	0.7781
Supervised POS	0.8154	0.7418	0.8156	0.7660	0.7857
Unsupervised POS	0.8083	0.7357	0.8326	0.7527	0.7817

- Orthographic predicates
- Character bigram and trigram predicates

In the case of chunking, features are only time-shifted by -1, 0, 1 and consist only of:

- Word itself
- POS-tag

This simple feature set for chunking was chosen to obtain a means of almost direct comparison of the different POS schemes without blurring results by other features or system components. Per system, three experiments were carried out, using standard POS features, unsupervised POS features and no POS features.

To evaluate the performance on NER, the methodology as proposed by the providers of the CoNLL-2002 (Roth and van den Bosch 2002) dataset is adopted: for all settings, the difference in performance in terms of the F1[8] measure is reported. Here, the Dutch dataset is employed, the unsupervised POS-tagger is induced on the 70 million token Dutch CLEF corpus, see (Peters 2006). Table 13 summarises the results of this experiment for selected categories using the full training set for training and evaluating on the test data.

The scores in Table 13 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 generalisation when using no POS-tags, in line with e.g. (Banko and Brill 2001) and (van den Bosch and Buchholz 2001). The situation changes when taking a closer look on the learning curve, produced by using train set fractions of differing size. Figure 12 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 generalisation 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.

This disagrees with the findings of (Freitag 2004b), where features produced by distributional clustering were used in a boosting algorithm. Freitag reports improved

[8] F1 is the harmonic mean of precision P (number of correct divided by number of assigned labels) and recall R (number of correct divided by number of all labels), $F1 = \frac{2PR}{P+R}$ cf. (Van Rijsbergen 1979).

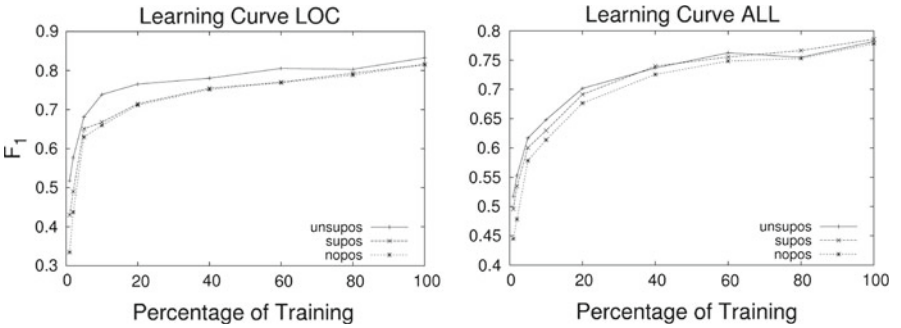


Fig. 12 Learning curves in NER task for category LOC and combined category

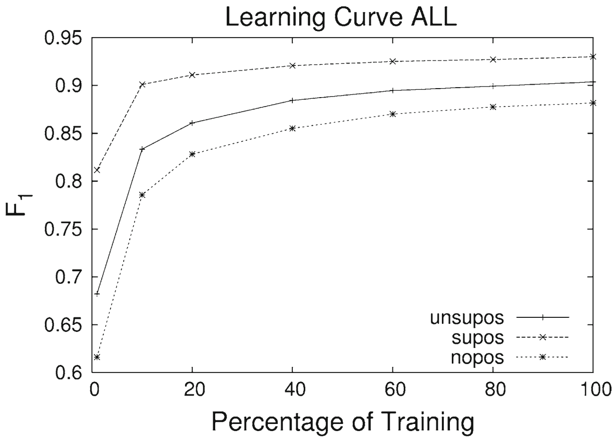


Fig. 13 Learning curve for the chunking task in terms of F1. Performance at 100% training is 0.882 (no POS), 0.904 (unsupervised POS) and 0.930 (supervised POS), respectively

performance on PERSON and ORGANISATION, but not on LOCATION, as compared to not using a tagger at all.

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 results in higher performance than supervised or no POS, which can be attributed to its finer-grained tagset that directly indicates types of named entities.

Performance of the simple chunking system was tested using different portions of the training set as provided in the English CoNLL-2000 data (Tjong kim Sang and Buchholz 2000) for training, evaluation was carried out on the provided test set. Performance is reported in Fig. 13.

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 significantly better chunking ($p < 0.01$) than using the unsupervised labels. This can be attributed to a smaller tagset for supervised POS, providing more reliable statistics because of less sparseness. Further, 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. 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).

7 Conclusion

An unsupervised POS-tagging system was described in detail and evaluated directly and indirectly on various languages and tasks. In difference to previous approaches to unsupervised POS-tagging, this method allows for a larger lexicon, where also POS ambiguities are handled. Further, the discovery of the number of POS categories is part of the method, rather than chosen beforehand.

Comparison with another unsupervised word clustering method shows that the model presented here differs from the system presented in (Clark 2003). Both systems yield competitive scores on evaluations, while the system described here is more expressive and faster to induce. The takeaway, however, is that combining several different unsupervised systems as features empowers supervised systems to reach higher performance levels.

Table 14 Available taggermodels to date for 14 languages, with corpus size (million sentences), lexicon size (thousand words) and number of tags

Language	Source	Sentences	Lexicon size	Tagset size
Catalan	LCC	3M	50K	369
Czech	LCC	4M	71K	538
Danish	LCC	3M	43K	376
Dutch	LCC	18M	140K	332
English	BNC	6M	26K	344
English	MEDLINE	34M	118K	479
Finnish	LCC	11M	130K	444
French	LCC	3M	42K	358
German	LCC	40M	258K	395
Hungarian	LCC	18M	180K	332
Icelandic	LCC	14M	132K	326
Italian	LCC	9M	85K	381
Norwegian	LCC	16M	135K	393
Spanish (Mexico)	LCC	4M	34K	414
Swedish	LCC	3M	43K	370

Evaluation on typologically different languages demonstrated the language-independence and robustness of the method. In indirect evaluation it was shown that for many tasks that use POS as a pre-processing step, there is no significant difference in results between using a trained POS-tagger or the unsupervised tagger presented here.

As far as performance in applications is concerned, the manual efforts necessary to construct a POS-tagger should rather be invested in collecting a large basis of text of the target domain or language, which can be also used for other purposes besides training the unsupervised POS-tagger.

An implementation of the unsupervised POS-tagger system by Andreas Klaus is available for download.[9] This implementation uses the parameter names as given in Table 8. On the same page, the tagger models listed in Table 14 can be obtained.

References

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of ACL-01* (pp. 26–33).

Biemann, C. (2006). Chinese whispers—an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of textGraphs: The second workshop on graph based methods for natural language processing* (pp. 73–80). New York City, June. Association for Computational Linguistics.

Biemann, C. (2007). *Unsupervised and knoweldge-free natural language processing in the structure discovery paradigm*. Ph.D. thesis, University of Leipzig.

Biemann, C., Giuliano, C., & Gliozzo A. (2007). Unsupervised part-of-speech tagging supporting supervised methods. In *Proceedings of recent advances in natural language processing (RANLP-07)*, Borovets, Bulgaria.

Brants, T. (2000). TnT: A statistical part-of-speech tagger. In *Proceedings of the sixth conference on applied natural language processing (ANLP-00)* (pp. 224–231). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Brants, T., Hendriks, R., Kramp, S., Krenn, B., Preis, C., Skut, W., et al. (1997). Das NEGRA-Annotationsschema. Negra project report, Universität des Saarlandes, Saarbrücken.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on applied natural language processing (ANLP-92)* (pp. 152–155). Morristown, NJ, USA: Association for Computational Linguistics.

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics, 18*(4), 467–479.

Burnard, L. (1995). *Users reference guide for the british national corpus*. Oxford, U.K.: Oxford University Computing Service.

Charniak, E., Hendrickson, C., Jacobson N., & Perkowitz, M. (1993). Equations for part-of-speech tagging. In *National Conference on Artificial Intelligence* (pp. 784–789).

Clark, A. (2000). Inducing syntactic categories by context distribution clustering. In Cardie, C., Daelemans, W., Nédellec, C., & Tjong Kim Sang, E., (Eds.), In *Proceedings of the fourth conference on computational natural language learning and of the second learning language in logic workshop, Lisbon, 2000* (pp. 91–94). Somerset, New Jersey: Association for Computational Linguistics.

Clark, A. (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 (EACL-03)* (pp. 59–66). Morristown, NJ, USA: Association for Computational Linguistics.

[9] http://wortschatz.uni-leipzig.de/~cbiemann/software/unsupos.html [July 7th, 2007].

Cucerzan, S., & Yarowsky, D. (1999). Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of 1999 joint SIGDAT conference on EMNLP and VLC* (pp. 132–138). College Park.

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic co-clustering. In *Proceedings of The ninth ACM SIGKDD international conference on knowledge discovery and data mining(KDD-2003)* (pp. 89–98).

Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics, 19*(1), 61–74.

Eiken, U. C., Liseth, A. T., Witschel, H. F., Richter, M., & Biemann, C. (2006). Ord i dag: Mining Norwegian daily newswire. In *Proceedings of the FinTAL*. Turku, Finland.

Ertöz, L., Steinbach, M., & Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Proceedings of workshop on clustering high dimensional data and its applications* (pp. 105–115).

Finch, S., & Chater, N. (1992). Bootstrapping syntactic categories using statistical methods. In *Background and experiments in machine learning of natural language: Proceedings of the 1st SHOE Workshop* (pp. 229–235). Brabant, Holland: Katholieke Universiteit.

Freitag, D. (2004a). Toward unsupervised whole-corpus tagging. In *Proceedings of the 20th international conference on Computational Linguistics (COLING-04)* (p. 357). Morristown, NJ, USA: Association for Computational Linguistics.

Freitag, D. (2004b). Trained named entity recognition using distributional clusters. In *Proceedings of EMNLP-04*, (pp. 262–269).

Garside, R., Leech, G., & Sampson, G. (1987). *The computational analysis of English: A corpus-based approach*. Harlow, UK: Longman.

Gauch, S., & Futrelle, R. (1994). Experiments in automatic word class and word sense identification for information retrieval. In *Proceedings of the 3rd annual symposium on document analysis and information retrieval* (pp. 425–434). Las Vegas, NV, April.

Gliozzo, A. M. (2005). *Semantic domains in computational linguistics*. Ph.D. thesis, University of Trento, Italy.

Gliozzo, A. M, Giuliano, C., & Strapparava, C. (2005). Domain kernels for word sense disambiguation. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL-05)* (pp. 403–410). Ann Arbor, Michigan, USA.

Goldwater, S., & Griffiths, T. (2007). A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 744–751). Prague, Czech Republic, June. Association for Computational Linguistics.

Hagen, K., Johannessen J. B., & Nøklestad, A. (2000). A constraint-based tagger for Norwegian. In Lindberg, C.-E. og S. Nordahl Lund (red.) *Proceedings of 17th scandinavian conference of linguistics, vol. I. Odense: Odense Working Papers in Language and Communication*, I(19).

Haghighi, A., & Klein, D. (2006). Prototype-driven learning for sequence models. In *Proceedings of the human language technology conference of the North American chapter of the association of computational linguistics (HLT-NAACL-06)*. New York, NY, USA.

Harris, Z. S. (1968). *Mathematical structures of language*. New York: Interscience Publishers.

Hal, D., III, & Daniel, M. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR), 26*, 101–126.

Knuth, D. E. (1998). *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th international conference on machine learning (ICML-01)* (pp 282–289). San Francisco, CA: Morgan Kaufmann

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Mihalcea, R., Chklovsky, T., & Kilgarriff, A. (2004). The SENSEVAL-3 english lexical sample task. In *Proceedings of SENSEVAL-3: Third international workshop on the evaluation of systems for the semantic analysis of text* (pp. 25–28). New Brunswick, NJ, USA.

Miller, G. A., Beckwith, R. T., Fellbaum, C. D., Gross, D., & Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography, 3*(4), 235–244.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1–28.

Ney, H., Essen, U., & Knese, R. (1994). On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language, 8*(1), 1–38.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses : An introduction*. NY: Wiley-Interscience.

Peters, C. (Ed.). (2006). *Working notes for the CLEF 2006 Workshop*. Alicante, Spain.

Potts, R. B. (1952). Some generalized order-disorder transformations. *Proceedings of the Cambridge Philosophical Society, 48*, 106–109.

Quasthoff, U., Richter, M., & Biemann, C. (2006). Corpus portal for search in monolingual corpora. In *Proceedings of the fifth international conference on language resources and evaluation (LREC-06)* (pp. 1799–1802).

Rapp, R. (2005). A practical solution to the problem of automatic part-of-speech induction from text. In *Conference companion volume of the 43rd annual meeting of the association for computational linguistics (ACL-05)*. Ann Arbor, Michigan, USA.

Roth, D., & van den Bosch, A. (Eds.). (2002). *Proceedings of the sixth workshop on computational language learning (CoNLL-02)*. Taipei, Taiwan.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International conference on new methods in language processing*. Manchester, UK.

Schütze, H. (1993). Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting on association for computational linguistics (ACL-93)* (pp. 251–258). Morristown, NJ, USA: Association for Computational Linguistics.

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of the 7th conference on European chapter of the association for Computational Linguistics (EACL-95)* (pp. 141–148), San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Tjong Kim Sang, E., & Buchholz, S. (2000). Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000*, Lisbon, Portugal.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003* (pp. 252–259).

van den Bosch, A., & Buchholz, S. (2001) Shallow parsing on the basis of words only: A case study. In *ACL '02: Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 433–440). Morristown, NJ, USA: Association for Computational Linguistics.

Van Rijsbergen C. J. (1979). *Information retrieval*, 2nd edition. Department of Computer Science, University of Glasgow.

Witschel, H. F., & Biemann, C. (2005). Rigorous dimensionality reduction through linguistically motivated feature selection for text categorization. In *Proceedings of NODALIDA'05*, Joensuu, Finland.

Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(11), 1101–1113.

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: application to accent restoration in spanish and french. In *Proceedings of the 32nd annual meeting on association for computational linguistics (ACL-94)* (pp. 88–95). Las Cruces, New Mexico.