# Hybrid Syntactic Category Induction

Bryan Jurish

Universität Potsdam, Institut für Linguistik, PF 601553, 14415 Potsdam, Germany

jurish@ling.uni-potsdam.de

July, 2005

### Abstract

Much research has been devoted to the task of learning lexical classes from unannotated input text. Among the chief difficulties facing any approach to the unsupervised induction of lexical classes are that of token-level ambiguity and the classification of rare and unknown words. Following the work of previous authors, the initial stage of syntactic category induction is treated in the current approach as a clustering problem over a small number of highly frequent word types. An iterative procedure making use of Zipf's law to generate the clustering schedule classifies less frequent words based on the monotonic Bernoulli entropy of expected co-occurrence probability with respect to the clusters output by the previous stage, employing a fuzzy cluster membership heuristic to approximate type-level ambiguity and reduce error propagation in a simulated melting procedure. In a second processing phase, cluster membership probabilities output by the final clustering stage are used in a procedure for the recovery of context-dependent token-level ambiguity resolution. The induced classifications are evaluated with a meta-modelling strategy intended to capture their expected linguistic utility.

## Contents

# 1 Introduction

Much research in computational language learning has been devoted to the task of learning lexical classes from unannotated input text. In particular, the unsupervised induction of syntactic categories by reference to the distributional similarity among their realizations (words) has been of great interest both for language modelling and for linguistically motivated approaches. Among the chief difficulties facing these and any other approach to unsupervised syntactic classification are that of *token-level ambiguity* – a given word type may properly belong to multiple syntactic categories, although each individual token realizes only one category – and that of *rare* and *unknown words*, the available data for which do not sufficiently motivate any classification at all. This paper presents a hybrid method for the induction of a syntactic classifier which respects the token/type distinction and which is further capable of classifying unknown text. The desired system output can be formally described as a function[1] $\tau : \mathcal{A}^* \to C^*$ which assigns to every element of an input sequence exactly one category.

Following the work of previous authors, the first phase of syntactic category induction is treated in the current approach as a clustering problem over word types, considering only a small number of highly frequent word types in the initial bootstrapping stage. An iterative procedure making use of Zipf's law to generate the clustering schedule classifies less frequent words based on the *monotonic Bernoulli entropy* of expected co-occurrence probability with respect to the clusters output by the previous stage, employing a fuzzy cluster membership heuristic to approximate type-level ambiguity and reduce error propagation.

The remainder of this paper is organized as follows: Section 2 describes selected previous

[1] Here and henceforth, $\mathcal{A}$ denotes a finite word alphabet, $C$ denotes a finite set of target categories, and $S^*$ denotes the Kleene closure of the set $S$. Variables $w$ and $c$ range over words and categories, respectively, and may be subscripted where necessary.

work on lexical category induction, Section 3 describes the method employed by the current approach in more detail, Section 4 presents a discussion of results for German and English, and Section 5 contains a brief summary and perspectives on some remaining questions.

# 2 Related Work

A detailed review of previous work on clustering techniques is beyond the scope of this document. The interested reader is referred to Jain et al. (1999) for a good survey of the clustering literature. Below, some of the most central characteristics of general clustering approaches are outlined, and some previous approaches to syntactic category induction are briefly summarized.

In general, the task of word-type clustering can be described intuitively as the automated discovery for each word type of its syntactic category on the basis of raw text input alone. To this end, word types must first be identified with some salient subset of their *distributional features*, and a *distance measure* over these features must be selected.

Output of a clustering procedure can be characterized as either *hard* or *soft* with respect to cluster membership criteria: *hard* clustering methods return a partitioning $\Pi : \mathcal{A} \rightarrow C$ of word-types into clusters, while *soft* methods return a cluster-membership probability distribution $p(c|w) : \mathcal{A} \times C \rightarrow [0, 1]$ conditioned on word types.

The degree to which output clusters themselves exhibit internal structure is often captured in terms of a distinction between *hierarchical* and *flat* clustering techniques. Hierarchical clustering algorithms return subsumption trees, the leaves of which represent the data to be clustered, while iterative algorithms usually return a set of opaque clusters often characterized in terms of a prototypical (pseudo-) element. *Iterative* clustering procedures are those which successively add new data elements to be clustered (usually one element per iteration), until all data have been assigned to some cluster.

Previous approaches to word-type clustering may be coarsely divided into *language modelling* and *linguistically motivated* approaches. Sections 2.1 and 2.2 are devoted to the respective approaches.

## 2.1 Language Modelling Approaches

Language modelling approaches to word-type clustering are those which can be reduced to a formal description of the task to be accomplished as one of *information maximization*, or equivalently of *code optimization*: the top-level goal of word-type clustering here is usually the construction of a maximally predictive language model with a minimal number of parameters.

Such approaches are characterized by (formal) notions of *minimum code length*, *information loss*, *entropy*, and *perplexity*. Typical distance functions include *mutual information*

(MI) and *Kullback-Leibler divergence* (KLD). Commonly used evaluation methods include test set probability, test-set entropy or perplexity, and KL-divergence with respect to a baseline model over a test set.

Brown et al. (1992) presents a language-modelling approach using a greedy bottom-up hierarchical clustering algorithm designed to maximize the average mutual information between adjacent clusters given empirical word-type bigram distributions. No provision is made for the word-type ambiguity. The algorithm is quite complex (on the order of $O(\mathcal{A}^3)$ even after severe optimization), leading the authors to consider an alternative iterative approach in which the selection of new clustering targets is guided by absolute word-type frequency, which is less complex than the hierarchical approach but not as successful at reducing perplexity.

An intriguing approach presented by Pereira et al. (1993) makes use of an analogy to the *free energy function* of statistical mechanics to cluster joint noun-verb distributions representative of verb subcategorization frames. The authors use the KL-divergence between distributions for cluster centroids and individual words as a similarity measure, together with a maximum entropy technique to update cluster membership probabilities in a flat centroid-based clustering procedure which is extended to produce hierarchical clusters in Lee (1997). Maximal ambiguity – every word belonging to every cluster with nonzero probability – is required in order to avoid singularities in the KL-divergence. No direct provision for context-dependent ambiguity of clusters is made, but since a joint distribution over word-type pairs mediated by clusters is learned, this is perhaps not a fatal flaw.

Clark (2000, 2001) describes an iterative agglomerative clustering procedure guided by word-type unigram frequency which uses the KL-divergence between smoothed empirical joint cluster-bigram distributions as a similarity measure. Hard clusters are produced by an initial clustering phase for the bulk of the vocabulary, while cluster membership probabilities for ambiguous words are estimated by expectation maximization (Dempster et al., 1977) for mixture models using KL-divergence between smoothed empirical context distributions – distributions over ⟨*left-neighbor, right-neighbor*⟩ pairs conditioned on word-types – and cluster centroids. Rare words are handled by an additional smoothing mechanism. The resulting classification has particular difficulties with ambiguous words, possibly resulting from their late incorporation into the learning procedure, and conceivably related to the fact that context is a poorer predictor of syntactic categories than word-type alone.

## 2.2 Linguistically Motivated Approaches

Linguistically motivated approaches to word-type clustering explicitly seek to induce classifications corresponding to conventional linguistic categories such as nouns, verbs, *etc.* The primary goal is formulated as a *cognitive modelling problem*, namely, the induction of *linguistically salient* categories.

Such approaches are characterized by notions such as *part-of-speech* or *syntactic category*, *semantic class*, and in particular of *type-level ambiguity*. Common evaluation methods include manual inspection by an expert (i.e. by a linguist), Hughes' (1994) benchmark with respect to a gold-standard corpus annotated by hand with a linguistically motivated tagset, and Schütze's (1995) method for estimating precision and recall with respect to a gold-standard. The latter two methods have been subject to some criticism in the literature, mostly motivated by language modelling concerns. These issues are addressed in more detail in Section 4.

Finch and Chater (1993) use Spearman's rank correlation coefficient as a similarity measure over empirical distributions of first- and second-order neighbors given target words to cluster the most frequent 1,000 word types in a 40 million word corpus by hierarchical agglomerative clustering. The authors report a success rate (compliance of learned clusters with a small number of traditional syntactic categories) near 95% as determined by manual inspection. Since the system produces hard clusters, several clusters are formed which correspond to ambiguous word types, which the authors count as successes. A more systematic investigation of standard agglomerative hierarchical clustering methods and distance measures is presented by Roberts (2002), who clusters the most frequent content words based on a fixed-width window of relative co-occurrence frequencies with respect to a small number of hand-selected function words.

Schütze (1993, 1995) uses a singular value decomposition on left- and right-context vectors of co-occurrence frequencies to select salient distributional features of the input language for the most frequent word types, and classifies the reduced vectors with the Buckshot clustering algorithm (Cutting et al., 1992b), an efficient clustering algorithm which combines hierarchical and iterative clustering techniques to produce semistructured hard clusters. The vector cosine is used as a similarity measure. Less frequent words are attached directly to the nearest cluster centroid output by the initial classification, by comparison of second-order context vectors based on co-occurrences with the *word classes* learned in the initial clustering stage. In Schütze (1995), trigram types are clustered by reference to the context vectors their component words.

Korkmaz and Üçoluk (1997) describe an approach making use of the Minkowski $L1$ norm as a distance metric over empirical joint bigram distributions, together with the soft cluster membership heuristic described by Gath and Geva (1989) to approximate type-level ambiguity in a postprocessing phase. A specialized distance metric for the procedure is proposed in Korkmaz and Üçoluk (1998). Another linguistically motivated approach is that described by Elghamry (2004), who uses relative mutual information between a target word and its immediate left- and right-neighbors to produce a coarse-grained unambiguous classification of word types which are then used in a procedure for bootstrapping subcategorization frames.

Of the previous approaches to syntactic word-type clustering described above, only those presented by Schütze (1993, 1995) make any provision for context-dependent token-level resolution of type-level syntactic ambiguity. Hidden Markov Model (HMM) taggers (Church, 1988; DeRose, 1988; Cutting et al., 1992a) also provide a mechanism by means of which word types may be represented as ambiguous, but in which a token is always assigned an unambiguous category, although the usual unsupervised learning method for HMMs – the *Baum-Welch* algorithm (Baum et al., 1970) – is known to perform poorly unless initialized with good estimates of the model's parameters (Elworthy, 1994).

# 3 Method

The method for syntactic word category induction described here proceeds in two main phases: first, target word-types are clustered in a multi-stage procedure, whose output is a category membership probability distribution $\hat{p}(C|W)$ conditioned on word types. In the second phase, membership probabilities are used to induce a mechanism for context-dependent token-level ambiguity resolution. Each phase of the procedure is described in more detail below.

## 3.1 Clustering Phase

The clustering procedure used here can be considered an extension of the efficient Buckshot clustering algorithm described by Cutting et al. (1992b) as employed for word-type clustering by Schütze (1993, 1995). An initial prototyping phase applies a traditional agglomerative hierarchical clustering algorithm to a small initial set of target words represented by *context vectors*. In subsequent stages, new target words are attached to an existing cluster. A *fuzzy membership heuristic* is used to incorporate the output of previous clustering solutions, and to mitigate the effects of the sparse data spaces common in natural language.

### 3.1.1 Iterative Target Selection

Clustering proceeds in $K$ stages, where $K$ is some fixed natural number,[2] At each stage of the procedure, a set $T_k \subset \mathcal{A}$ of *targets* are selected for clustering, as well as a finite set $B_k$ of *boundaries* (also referred to here as "bounds"). In the experiments described below, initial targets and bounds were selected on the basis of their global frequency ranks, and subsequent targets were selected on the basis of their co-occurrence frequency ranks with respect to previous targets. The initial target and boundary sets were identical, $T_1 = B_1$.[3]

Zipf's law (Zipf, 1949), which states that there is an inverse relation between a word's frequency and its rank in a list of words sorted in ascending order by frequency, was used to generate a *clustering schedule*. As mentioned above, targets are selected by virtue of frequency rank, and the maximum ranks of words chosen as targets for successive clustering

[2] In the experiments described here, the maximum number of iterations $K$ was fixed at 10.

[3] One additional boundary element was used at each stage to mark beginning- and end-of-sentence for left- and right-bigrams, respectively.

stages constitute a power series, which by Zipf's law approximates a linear decline in expected log target frequency.[4] The formula used to recursively compute the maximum target rank $r_{k+1}$ for stage $k+1$ was:[5]

$$\log r_{k+1} = \log r_k + \left( \frac{\log(|\mathcal{A}|) - \log(|r_1|)}{K-1} \right) \tag{1}$$

In the interest of reducing computational complexity, each target word is subject to direct inspection in only one clustering stage: $j \neq k$ implies $T_j \cap T_k = \emptyset$ for $1 \leq j, k \leq K$. Further, after the initial stage, the boundary set was identified with the set of clusters discovered by the previous iteration: $B_{k+1} = C_k$ for $1 \leq k < K$, where $C_k$ is the set of clusters output by the clustering stage $k$, $C_k \cap \mathcal{A} = \emptyset$.

### 3.1.2 Prototyping Stage, $k = 1$

Left- and right-bigram frequencies $f_{\ell,k}, f_{r,k} : T_k \times B_k \rightarrow \mathbb{R}$ from an untagged input corpus were collected for the target and bound sets. Let $f_0 : \mathcal{A}^2 \rightarrow \mathbb{N}$ be the raw corpus frequency function, and define:

$$\begin{aligned} f_{\ell,1}(w,b) &= f_0(b,w) & (2) \\ f_{r,1}(w,b) &= f_0(w,b) & (3) \end{aligned}$$

Here and henceforth, let $z \in \{\ell, r\}$ range over directions, and define:

$$\begin{aligned} f_{z,k}(w) &= \sum_{b \in B_k} f_{z,k}(w,b) & (4) \\ f_{z,k}(b) &= \sum_{w \in T_k} f_{z,k}(w,b) & (5) \\ N_{z,k} &= \sum_{w \in T_k} f_{z,k}(w) & (6) \end{aligned}$$

Bigram frequencies are used to compute the empirical (maximum likelihood) probability distributions $P_{\ell,k}$ and $P_{r,k}$ in the usual manner:[6]

$$\begin{aligned} \mathrm{P}_{z,k}(w,b) &= \frac{f_{z,k}(w,b)}{N_{z,k}} & (7) \\ \mathrm{P}_{z,k}(w) &= \frac{f_{z,k}(w)}{N_{z,k}} & (8) \end{aligned}$$

[4] Hardware limitations led to the implementation of an additional parameter which places a strict upper bound on the number of new targets incorporated at any given stage. In the experiments described here, the growth bound was arbitrarily fixed to 32768, which corresponds to a memory footprint of 25 Mb for the stage data matrix.

[5] The maximum rank of the initial target and boundary sets $r_1$ was given as an additional parameter. In the experiments described below, $r_1$ was fixed at 200.

[6] Here and elsewhere, it is assumed that $\frac{0}{0} = 0$.

The collected frequency data is used to populate a real-valued feature vector $\vec{w}_k \in \mathbb{R}^{2|B_k|}$ for each target word $w \in T_k$. Target vectors were constructed by concatenating left- and right- subvectors $\vec{w}_{\ell,k}, \vec{w}_{r,k} \in \mathbb{R}^{|B_k|}$

$$\begin{aligned} \vec{w}_{z,k} &= [\vec{w}_{z,k}(1), \ldots, \vec{w}_{z,k}(|B_k|)] && (9) \\ \vec{w}_k &= \vec{w}_{\ell,k} \circ \vec{w}_{r,k} = [\vec{w}_{\ell,k}(1), \ldots, \vec{w}_{\ell,k}(|B_k|), \vec{w}_{r,k}(1), \ldots, \vec{w}_{r,k}(|B_k|)] && (10) \end{aligned}$$

where $\vec{x}(i)$ denotes the $i^{th}$ component of the vector $\vec{x}$, and the subvectors $\vec{w}_{\ell,k}$ and $\vec{w}_{r,k}$ are defined with respect to an enumeration $b_1, \ldots, b_{|B_k|}$ of the current boundary set $B_k$.

**3.1.2.1 Target Features** Three methods for vector population or *feature selection* were considered here: raw frequency, conditional empirical probability, and monotonic Bernoulli entropy, each of which is presented in detail below.

**Raw Frequency Features** Raw frequency vectors use the directed corpus frequency directly to initialize the components of the feature vector, for $1 \leq i \leq |B_k|$:

$$\vec{w}_{z,k}(i) = f_{z,k}(w, b_i) \tag{11}$$

**Conditional Probability Features** Conditional probability target vectors use the empirical conditional distributions to populate the target features:

$$\vec{w}_{z,k}(i) \quad = \quad \mathrm{P}_{z,k}(b_i|w) = \frac{\mathrm{P}_{z,k}(w, b_i)}{\mathrm{P}_{z,k}(w)} \tag{12}$$

**Monotonic Bernoulli Entropy Features** The *Shannon entropy* of a distribution $p$ is the average length of a message that an event from $p$ has occurred under an optimal encoding (Shannon and Weaver, 1949), and can be understood as a measure of the unpredictability of the events in $p$, with 0 as a lower bound:[7]

$$\mathrm{H}(p) = -\sum_{x \in \mathrm{dom}(p)} p(x) \log_2 p(x) \tag{13}$$

A variant of the Shannon entropy is used here to populate context vector components for target words. Although it is possible to compute entropy contributions for individual events $x$ (corresponding to the joint occurrence of a given *target, bound* pair), the resulting function $h_p(x) = -p(x) \log p(x)$ is not symmetric, since entropy is properly defined over

[7] Code lengths are conventionally expressed in *bits*, thus we use base 2 logarithms here and elsewhere, omitting the subscript. An additional convention required for the computation of entropy is that $0 \log 0 = 0$.

distributions rather than events. To accommodate this fact, a Bernoulli distribution is assumed for each boundary element given a target word:

$$\begin{aligned} \mathrm{H}(X \sim \mathrm{b}(1; p_x)) &= -\sum_{x \in \{0,1\}} p(x) \log p(x) \\ &= -p_x \log p_x - (1 - p_x) \log(1 - p_x) \end{aligned} \tag{14}$$

The Shannon entropy of Bernoulli distributions is indeed symmetric, and can be computed from a single parameter $p_x$ corresponding here to the empirical conditional probability of a boundary element given a target element. The resulting function is however non-monotonic: no difference is drawn between high- and low-probability events: $\mathrm{H}(\mathrm{b}(1; p_x)) = \mathrm{H}(\mathrm{b}(1; 1 - p_x))$ for all $p \in [0..1]$. This property is desirable from an information-theoretic standpoint, as it properly ascribes identical entropies to equivalent distributions. For the current purpose of word-type context vector population however, it is unintuitive to suppose that a high-frequency predictor (boundary element) and a low-frequency predictor should be considered equivalent for the same target element.[8] Therefore, a modified function $\hat{\mathrm{H}}$ based on the Shannon entropy of Bernoulli distributions is used to populate target vector components for each boundary element:

$$\hat{\mathrm{H}}(p_x) = \begin{cases} \mathrm{H}(\mathrm{b}(1; p_x)) & \text{if } p_x \leq \frac{1}{2} \\ 2 - \mathrm{H}(\mathrm{b}(1; p_x)) & \text{otherwise} \end{cases} \tag{15}$$

The function $\hat{\mathrm{H}}$ – referred to henceforth as the "monotonic Bernoulli entropy" – applied to pointwise boundary probabilities conditioned on targets can be intuitively understood as a heuristic for estimating the mnemonic utility of "chunking" the boundary event into the target event. Monotonic Bernoulli entropy is symmetric, restricted to the range [0..2], and grows monotonically with its only parameter $p_x$. However, it is not in general the case that the values of $\hat{\mathrm{H}}(\mathrm{P}_{z,k}(\cdot|w))$ result in meaningful comparison criteria over target words $w$, due to the case differentiation in Equation 15. For this reason, an absolute mass of 1 is allotted to each target word $w \in T_k$, allowing the population of target vectors with unit-normalized conditional monotonic Bernoulli entropies of boundary elements given target words:

$$\vec{w}_{z,k}(i) \;=\; \tilde{\mathrm{H}}_{z,k}(b_i|w) = \frac{\hat{\mathrm{H}}(\mathrm{P}_{z,k}(b_i|w))}{\sum_{b \in B_k} \hat{\mathrm{H}}(\mathrm{P}_{z,k}(b|w))} \tag{16}$$

**3.1.2.2 Prototype Clustering** The characteristic feature vectors $\vec{w}$ for each target $w \in T_k$ were assembled into a $2|B_k| \times |T_k|$ clustering matrix $M_k$, the rows of which correspond to current target words, and the columns of which correspond to (monotonic Bernoulli entropies of) bounds. In the initial prototyping stage, the targets (rows of the data matrix) were clustered with a standard agglomerative hierarchical clustering algorithm from the C Clustering Library package (de Hoon et al., 2004). Several distance measures

[8] Preliminary experiments supported this intuition.

and link methods were used, which are formally presented in Sections 3.1.2.3 and 3.1.2.4, respectively.

The resulting clustering tree was cut to produce a user-specified number of clusters.[9] As an alternative to specifying the number of clusters as a system parameter, the user might instead specify a node linkage distance threshold, thus allowing the procedure to discover an "optimal" number of clusters.

**3.1.2.3 Distance Functions** Three distance functions were considered in the experiments described here: the $L1$ distance metric[10] $d_{L1}$ as used by Korkmaz and Üçoluk (1997), Spearman's rank correlation coefficient $d_S$ as used by Finch and Chater (1993), and the vector cosine as employed by Schütze (1993, 1995). Of these, the simplest is the $L1$ distance, which is a true metric defined by the Minkowski $1-$norm, $\|\cdot\|_1$. Let $n_k = 2|B_k|$ be the length of target vectors at clustering stage $k$, then:

$$d_{L1}(\vec{w}_k, \vec{v}_k) \quad = \quad \|\vec{w}_k - \vec{v}_k\|_1 \quad = \quad \sum_{i=1}^{n_k} |\vec{w}_k(i) - \vec{v}_k(i)| \tag{17}$$

Both the vector cosine and Spearman's rank correlation coefficient can be reduced to Pearson's correlation coefficient $r_P$, itself defined in terms of the sample means $\mu$ and standard deviations $\sigma$:

$$d_P(\vec{w}_k, \vec{v}_k) \quad = \quad 1 - r_P(\vec{w}_k, \vec{v}_k) \tag{18}$$

where:

$$\begin{aligned} r_P(\vec{w}_k, \vec{v}_k) &= \frac{1}{n_k}\sum_{i=1}^{n_k}\left(\frac{\vec{w}_k(i) - \mu_{\vec{w}_k}}{\sigma_{\vec{w}_k}}\right)\left(\frac{\vec{v}_k(i) - \mu_{\vec{v}_k}}{\sigma_{\vec{v}_k}}\right) \\ \mu_{\vec{x}_k} &= \frac{1}{n_k}\sum_{i=1}^{n_k}\vec{x}_k(i) \\ \sigma_{\vec{x}_k} &= \sqrt{\frac{1}{n_k}\left(\sum_{i=1}^{n_k}\vec{x}_k(i) - \mu_{\vec{x}_k}\right)^2} \end{aligned}$$

The vector cosine can be reduced to Pearson's coefficient by assuming that the vector means $\mu_{\vec{w}_k}$ and $\mu_{\vec{v}_k}$ are zero:

$$\begin{aligned} d_{cos}(\vec{w}_k, \vec{v}_k) &= 1 - r_{cos}(\vec{w}_k, \vec{v}_k) \qquad (19) \\ r_{cos}(\vec{w}_k, \vec{v}_k) &= \frac{1}{n_k}\sum_{i=1}^{n_k}\left(\frac{\vec{w}_k(i)}{\sigma^{(0)}_{\vec{w}_k}}\right)\left(\frac{\vec{v}_k(i)}{\sigma^{(0)}_{\vec{v}_k}}\right) \\ \sigma^{(0)}_{\vec{x}_k} &= \sqrt{\frac{1}{n_k}\sum_{i=1}^{n_k}\vec{x}_k(i)^2} \end{aligned}$$

[9] In the experiments described, the number of output clusters was fixed at 50.

[10] $L1$ distance is also sometimes referred to as "taxi-cab", "city-block", or "Manhattan" distance.

Spearman's rank correlation coefficient is simply Pearson's coefficient applied to vectors of features' ranks rather than to actual feature values. It is a non-parametric similarity measure which is typically more robust with respect to outliers than the other distance functions considered here.

$$d_S(\vec{w}_k, \vec{v}_k) \;=\; 1 - r_P(\text{ranks}(\vec{w}_k), \text{ranks}(\vec{v}_k)) \tag{20}$$

where:

$$\begin{aligned}
\text{ranks}(\vec{x}_k) &= [\text{rank}(\vec{x}_k, 1), \ldots, \text{rank}(\vec{x}_k, n_k)] \\
\text{rank}(\vec{x}_k, i) &= \min\left(\pi_\text{r}(\vec{x}_k, i)\right) + \frac{|\pi_\text{r}(\vec{x}_k, i)|}{2} \\
\pi_\text{r}(\vec{x}_k, i) &= \{j \mid 1 \leq j \leq n_k \;\&\; \vec{x}_k(j) = \vec{x}_k(i)\}
\end{aligned}$$

**3.1.2.4 Link Methods** For agglomerative hierarchical clustering, the link method defines the manner in which the distance function $d : \mathbb{R}^{n_k} \times \mathbb{R}^{n_k} \to \mathbb{R}$ over individual target vectors $\vec{w}, \vec{v} \in \mathbb{R}^{n_k}$ is extended to a function $\hat{d} : \mathcal{P}(\mathbb{R}^{n_k}) \times \mathcal{P}(\mathbb{R}^{n_k}) \to \mathbb{R}$ over sets of vectors $W, V \in \mathcal{P}(\mathbb{R}^{n_k})$. Without loss of generality, $\hat{d}(W, V)$ will be written $d(W, V)$, and $\hat{d}(W, \{\vec{v}\})$ will be abbreviated $d(W, \vec{v})$.

Maximum-link defines the distance between clusters as the maximum pairwise distance between their respective elements:

$$\hat{d}_{max}(W, V) \;=\; \max_{\vec{w} \in W, \vec{v} \in V} d(\vec{w}, \vec{v}) \tag{21}$$

Average-link defines the distance between clusters as the arithmetic average of the pairwise distances between their respective elements:

$$\hat{d}_{avg}(W, V) \;=\; \operatorname*{avg}_{\vec{w} \in W, \vec{v} \in V} d(\vec{w}, \vec{v}) \;=\; \frac{1}{|W| \times |V|} \sum_{\vec{w} \in W, \vec{v} \in V} d(\vec{w}, \vec{v}) \tag{22}$$

**3.1.2.5 Fuzzy Cluster Membership** The primary output[11] of each clustering stage $k$ is a cluster-membership probability distribution $\hat{p}_k(\cdot|\cdot) : T_k \times C_k \to [0, 1]$ conditioned on target words: $\sum_{c \in C_k} \hat{p}_k(c|w) = 1$ for all $w \in T_k$. Standard hierarchical clustering techniques provide only a univocal ("hard") classification however, thus predicting for each target $w$ that $\hat{p}_k(c|w) = 0$ for all but one cluster $c$. In an attempt to approximate the potential ambiguity of natural language word types, and to reduce the impact of propagating erroneous clustering decisions from a stage $k$ into later stages $k + j$, a fuzzy cluster membership heuristic is employed.

[11] In the context of the full multi-stage procedure, the final output of each clustering stage must incorporate targets from all preceding stages into a membership probability distribution $\hat{p}_{\leq k}$; for $k = 1$, $\hat{p}_{\leq k} = \hat{p}_{\leq 1} = \hat{p}_1$. The ordering notation is extended in the obvious manner for all $k > 1$: $\hat{p}_{<k} = \hat{p}_{\leq k-1}$.

The distance measure and link-method used in the clustering algorithm itself provide the most natural source of data on the basis of which to estimate $\hat{p}_k$: namely, the distances between each target vector and each node identified as a cluster. The heuristic which performed best in the experiments described here is defined in terms of the similarity measure $\hat{\mathrm{s}} : C_k \times T_k \rightarrow \mathbb{R}$ given by Equation 23, which results in a variant of the exponential form for membership distributions used by Pereira et al. (1993) and attributed to Jaynes (1983). Let $d_k : \mathcal{P}(\mathbb{R}^{n_k}) \times \mathcal{P}(\mathbb{R}^{n_k}) \rightarrow \mathbb{R}$ be the distance function over sets of target vectors in clustering stage $k$, and let $\beta_k > 0$ be an *inverse temperature* parameter for stage $k$, then:

$$\hat{\mathrm{s}}_k(c, w) = \exp(-\beta_k d_k(c, \vec{w}_k)) \tag{23}$$

The parameter $\beta_k$ was set to $\frac{1}{k}$ at each stage $k$, which corresponds to a progressive "heating" of the system, and therefore to increasing uncertainty with respect to the reliability of the classifications made in successive stages. It should be noted that this behavior – justified by Zipf's law in that later-stage targets must be classified on the basis of a smaller sample than early-stage targets and thus produce less reliable classifications – is precisely the opposite of the *simulated annealing* techniques used by Pereira et al. (1993) and Lee (1997). By the same analogy to physical processes, the manipulation of $\beta_k$ used here results rather in a *simulated melting* procedure.

After some experimentation, it was determined that limiting the number of clusters to which a target may be considered to belong with nonzero probability provided a useful limitation of the search space. The natural language pendant of such a restriction is an *a priori* upper limit on the degree of syntactic ambiguity of any given word. Let $m$ be a parameter specifying the maximum number of clusters to which a word $w$ may belong with nonzero probability,[12] and let $\mathrm{rank}_{d_k,w}(c)$ be the rank of the distance between the target $w$ and the cluster $c$ according to the distance function $d_k$,[13] then Equation 23 becomes:

$$\hat{\mathrm{s}}_k(c, w) = \begin{cases} \exp(-\beta_k d_k(c, \vec{w}_k)) & \text{if } 1 \leq \mathrm{rank}_{d_k,w}(c) \leq m \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Given $\hat{\mathrm{s}}$, the membership probability distribution $\hat{p}_k$ can be estimated by:

$$\hat{p}_k(c|w) = \frac{\hat{\mathrm{s}}_k(c, w)}{Z_{k,w} = \sum_{c' \in C_k} \hat{\mathrm{s}}_k(c', w)} \tag{25}$$

### 3.1.3 Attachment Stages, $k > 1$

After the initial prototyping clustering stage, the boundary set $B_k$ is defined as the set $C_{k-1}$ of clusters output by the previous clustering stage. This restriction results in a smaller

[12] In the experiments described below, the parameter $m$ was set to 4.

[13] Given a list $d_k(c_1, w), ..., d_k(c_{|C_k|}, w)$ sorted in ascending order of cluster distances with respect to $w$, where $i \neq j$ implies $c_i \neq c_j$, $\mathrm{rank}_{d_k,w}(c) = i$ for $1 \leq i \leq |C_k|$. In the rank computation method used here, equidistant clusters were assigned distinct ranks based on the order in which the clusters were established during the prototyping stage.

memory footprint and fewer computations at each stage, compared to simply including all previous targets as unique elements of the boundary set in their own right. Additionally, the use of clusters as boundary elements allows the system to capitalize on knowledge gained by previous stages, and can be considered in this sense an extension of Schütze's (1995) dual stage word-type clustering procedure.

The mechanics of estimating expected bigram frequencies with respect to clusters are detailed in Section 3.1.3.1. Section 3.1.3.2 describes the method for by which cluster centroids are acquired. Finally, the procedure for classifying new targets with respect to the acquired centroids is described in Section 3.1.3.3.

**3.1.3.1 Cluster-based Boundary Profiling** Since $B_k = C_{k-1}$ for stage $k > 1$, and since the clusters $b \in C_{k-1}$ are not annotated directly in the input corpus, Equations 2 and 3 cannot directly be used to compute bigram frequencies, in terms of which the instantiation of the characteristic feature-vectors for the clustering algorithm is defined. The membership probability distribution $\hat{p}_{<k}$ returned by the preceding clustering stage however provides a natural starting point for the incorporation of previous solutions into an *expected frequency*. Let $T_{<k} = \bigcup_{i=1}^{k-1} T_i$, then for $w \in T_k$ and $b \in B_k$, define:

$$f_{\ell,k}(w,b) \quad = \quad \sum_{v \in T_{<k}} \hat{p}_{<k}(b|v) f_0(v,w) \tag{26}$$

$$f_{r,k}(w,b) \quad = \quad \sum_{v \in T_{<k}} \hat{p}_{<k}(b|v) f_0(w,v) \tag{27}$$

The modelling assumptions which allow us to derive Equations 26 and 27 from word-type bigram distributions are made explicit in Equations 28–31 for left bigrams; the case for right bigrams is analogous.

$$f_{\ell,k}(w,b) \quad = \quad p_{\ell,k}(w,b) N_{\ell,k} \tag{28}$$

$$p_{\ell,k}(w,b) \quad = \quad \sum_{v \in T_{<k}} p_{\ell,k}(v,w,b) \tag{29}$$

$$p_{\ell,k}(w,v) \quad = \quad \frac{f_0(v,w)}{N_{\ell,k}} \tag{30}$$

$$p_{\ell,k}(b|v,w) \quad = \quad \hat{p}_{<k}(b|v) \tag{31}$$

Here, Equations 28 and 30 are just maximum-likelihood estimators for bigram probabilities. Equation 29 simply defines the joint target-cluster bigram distribution $p_{\ell,k}(w,b)$ as the marginal distribution obtained by summing over all potential cluster members $v \in T_{<k}$. Finally, Equation 31 asserts the conditional independence of boundary clusters $b \in B_k$ from target words $w \in T_k$ given boundary words $v \in T_{<k}$, allowing the use of the membership probability distribution $\hat{p}_{<k}$ to compute the marginal distributions.

Characteristic feature vectors $\vec{w}_k$ for each new target $w$ are then instantiated by computing the relevant target features and concatenating left- and right-subvectors, as in the prototyping stage.

**3.1.3.2 Centroid Acquisition** In order to classify new targets in terms of existing clusters, a *centroid profile* matrix $\hat{M}_k$ analogous to the target profile matrix $M_k$ is created for the clusters output by the preceding stage.

Let $\pi_{<k} : T_{<k} \to C_{k-1}$ be the "hard" partitioning of previous targets computed by the previous clustering stages, and let $\pi_{<k}^{-1} : C_{k-1} \to \mathcal{P}(T_{<k})$ be its inverse relation: $\pi_k^{-1}(c) = \{w \in T_{<k} \mid \pi_{<k}(w) = c\}$. Then, expected frequencies for cluster centroids $c \in C_{k-1}$ are estimated by summing over the frequencies of their elements:

$$f_{z,k}(c,b) \quad = \sum_{w \in \pi_{<k}^{-1}(c)} f_{z,k}(w,b) \tag{32}$$

Important to note here is that the fuzzy cluster-membership heuristic is used only to smooth a previous cluster's behavior as a boundary element, and not its behavior as a centroid: for the frequency profiling of cluster centroids, "hard" clusters are assumed, which increases the distinguishability of centroid signatures.

Characteristic feature vectors $\vec{c}_k$ for each centroid $c \in C_{k-1}$ are instantiated by the same procedure used for new targets, giving rise to a $2|B_k| \times |C_{k-1}|$ centroid profile matrix $\hat{M}_k$ over the same features as the target profile matrix $M_k$.

**3.1.3.3 Attachment** Given a target profile matrix $M_k$ and a centroid profile matrix $\hat{M}_k$, new targets can be assigned to existing clusters by an application of the *Assign-to-Nearest* operator described by Cutting et al. (1992b), which simply assigns each target to the "nearest" centroid, in the sense defined by the clustering distance function $d$.

Membership probabilities $\hat{p}_k(c|w)$ for new targets $w \in T_k$ can be directly computed by Equation 25. Membership probabilities for previously clustered targets are left unchanged:

$$\hat{p}_{\leq k} = \bigcup_{i=1}^{k} \hat{p}_i \tag{33}$$

Since $T_i \cap T_j = \emptyset$ for $i \neq j$, the membership distribution $\hat{p}_{\leq k}$ is a function with:

$$\hat{p}_{\leq k}(c|w) \quad = \quad \begin{cases} \hat{p}_k(c|w) & \text{if } w \in T_k \\ \hat{p}_{<k}(c|w) & \text{if } w \in T_{<k} \\ undefined & \text{otherwise} \end{cases}$$

The membership probabilities thus estimated can be passed to the next clustering iteration for the classification of new targets, or may be exported directly to the ambiguity resolution phase.

## 3.2 Ambiguity Resolution Phase

Two methods for context-dependent token-level ambiguity resolution were considered: application of the Baum-Welch algorithm to a first-order Hidden Markov Model initialized with emission probabilities computed from the membership distribution $\hat{p}_K$, and a variant of the trigram clustering technique presented by Schütze (1995).

### 3.2.1 Hidden Markov Model Reestimation

The membership distribution $\hat{p}_{\leq K}$ returned by the final clustering stage may be used to initialize the emission probabilities of a Hidden Markov Model by an application of Bayes' Rule. For $w \in T_{\leq K}, c \in C_K$:

$$\hat{p}_{\leq K}(w|c) = \frac{\hat{p}_{\leq K}(c|w)\hat{p}_{\leq K}(w)}{\hat{p}_{\leq K}(c)} \tag{34}$$

where:

$$\begin{aligned}\hat{p}_{\leq K}(w) &= \frac{\mathrm{P}_{\ell,K}(w) + \mathrm{P}_{r,K}(w)}{2} & (35)\\ \hat{p}_{\leq K}(c) &= \sum_{w \in T_{\leq k}} \hat{p}_{\leq K}(w,c) & (36)\\ &= \sum_{w \in T_{\leq k}} \hat{p}_{\leq K}(c|w)\hat{p}_{\leq K}(w)\end{aligned}$$

Transition and initial probabilities for the HMM were initialized with uniform distributions, and the resulting models passed through several iterations of the Baum-Welch algorithm for parameter reestimation. The multi-sequence form of the Baum-Welch algorithm as described by Rabiner (1989) was used here, for which each sentence of the reestimation corpus was treated as a single observation sequence.

### 3.2.2 Trigram Clustering

A different method for context-dependent resolution was explored by Schütze (1995), who classifies word-type *trigrams* by reference to the left- and right-context vectors of their component word types. For word types $w_1, w_2, w_3 \in \mathcal{A}$, let $w_{1..3} = \langle w_1, w_2, w_3 \rangle \in \mathcal{A}^3$, then:

$$\overrightarrow{w_{1..3}} = \overrightarrow{(w_1)}_r \circ \overrightarrow{(w_2)}_\ell \circ \overrightarrow{(w_2)}_r \circ \overrightarrow{(w_3)}_\ell \tag{37}$$

In Schütze's method, prototype target trigrams were randomly selected, and context vectors were populated with raw frequency features. In the experiments described here, a variation was considered in which target trigrams were selected on the basis of their components' co-occurrence frequencies with word-type targets clustered by the word-type clustering phase, and context vectors were populated as in the word-type attachment substages by reference to expected co-occurrence frequencies with word-type clusters:

$$\overrightarrow{(w_{1..3})}_K = \overrightarrow{(w_1)}_{r,K} \circ \overrightarrow{(w_2)}_{\ell,K} \circ \overrightarrow{(w_2)}_{r,K} \circ \overrightarrow{(w_3)}_{\ell,K} \tag{38}$$

After clustering a selection of prototype trigrams, all remaining trigrams from the test corpus were attached to the nearest trigram cluster centroid, thus providing a limited mechanism for context-dependent ambiguity resolution.

### 3.3 Computational Complexity

Collection of unigram and bigram statistics from the training corpus has linear time complexity, $\mathcal{O}(|S|)$, where $|S|$ is the size of the training corpus. Ranking of word-types by frequency using a good sorting algorithm will be $\mathcal{O}(|\mathcal{A}| \log |\mathcal{A}|)$ on average. A clustering schedule can be generated in constant time given the alphabet size.

The time complexity of the agglomerative hierarchical clustering algorithm used in the prototyping stage is on average $\mathcal{O}(|T_1|^2 \log |T_1|)$ (Jain et al., 1999; Cutting et al., 1992b), or more precisely $\mathcal{O}(|B_1| \times |T_1|^2 \log |T_1|)$.

Attachment of new targets to existing centroids is $\mathcal{O}(|B_k| \times |T_k| \times |C_{k-1}|)$. Let $\mathcal{T} = \frac{1}{K} \sum_{k=1}^{K} |T_k|$ be the average number of targets clustered at any stage, let $\mathcal{C} = |C_1| = \cdots = |C_K|$ be the number of target clusters. Then, the total time complexity of the attachment phases is $\mathcal{O}(K\mathcal{C}^2\mathcal{T})$. Important to note is that the number of bounds (features) plays a nontrivial role in the complexity of the clustering subphase, motivating the identity $B_k = C_{k-1}$ used in the procedure described above, since $\mathcal{C}$ is likely to be small. Further, if only a small number of initial targets and bounds are chosen ($|T_1| = |B_1| \leq \sqrt{\mathcal{C}\mathcal{T}}$), the total complexity of the prototyping and attachment phases will be $\mathcal{O}(K\mathcal{C}^2\mathcal{T})$, which does not exceed $\mathcal{O}(\mathcal{C}^2|\mathcal{A}|)$.

Estimation of the membership probabilities $\hat{p}(\cdot|\cdot)$ can likewise be reduced to $\mathcal{O}\left(K\mathcal{C}^2\mathcal{T}\right)$, since no full sort is required for a fixed value of the maximal ambiguity rate parameter $m$, and since membership probabilities do not change for any given target once it has been clustered.

The Baum-Welch algorithm has time complexity $\mathcal{O}\left(\mathcal{C}^2 \times \mathcal{S}\right)$, where $|S|$ is the size of the reestimation corpus. Trigram profiling with respect to word-type cluster centroids is $\mathcal{O}\left(\mathcal{C}|\mathcal{A}|\right)$. Clustering of $\mathcal{T}_3$ trigram prototypes is then $\mathcal{O}\left(\mathcal{C}^2\mathcal{T}_3| \log \mathcal{T}_3\right)$. Analogous to the argument given above for the choice of initial word-type targets, trigram prototypes can be chosen so that the total complexity of the trigram clustering and attachment procedure does not exceed $\mathcal{O}\left(\mathcal{C}^2 \times |\mathcal{A}_3|\right)$, where $\mathcal{A}_3 = \{uvw \in \mathcal{A}^3 \mid f_0(uvw) > 0\}$ is the set of all actually occurring trigrams.

# 4 Results & Discussion

The system described in the previous section was implemented in Perl and C, and evaluated on both German and English corpora. For German, the NEGRA corpus (Skut et al., 1997) was used, which after preprocessing contained 355,096 tokens of 48,924 distinct or-

| **Tag** | **SUSANNE Tag(s)** | **Description** |
|---|---|---|
| ADV | FA* FB* LE* XX | Adverb |
| DET | A* D* | Determiner |
| CARD | M* | Cardinal number |
| CCONJ | CC* | Coordinating conjunction |
| POS | G* | Genitive marker |
| MISC | FO* FU* FW* UH ZZ* | Miscellaneous |
| NOUN | N* BTO22 | Nominal |
| PREP | I* BTO21 | Preposition |
| PRON | P* EX | Pronominal |
| PUNC | Y* | Punctuation |
| SCONJ | CS* | Subordinating conjunction |
| TO | TO | Infinitival *to* |
| VFIN | V* (except V*0, V*G*) | Finite verb form |
| VINF | VB0 VD0 VH0 | Infinitive verb form |
| VING | VBG VDG VHG VVG* | -ing verb form |

Table 1: Tagset reduction scheme used for the SUSANNE corpus. The character "*" is used a wildcard.

thographic forms. For English, a composite corpus comprised of the SUSANNE corpus (Sampson, 1995) and the novel *Great Expectations* (Dickens, 2002) was used, which after preprocessing contained 374,640 tokens of 20,600 distinct orthographic forms. 10% of the sentences in each corpus were randomly selected and reserved for testing. For English, the test corpus sentences were drawn exclusively from the tagged SUSANNE corpus. Training corpora were annotated only with sentence- and token-boundaries. All alphabetic characters were converted to lower case. Punctuation marks were preserved as individual tokens.

The primary method of evaluation was a simple meta-modelling strategy similar to that used by Schütze (1995): words in the linguistically motivated gold standard corpora were replaced with tag identifiers for the induced clusters, and a supervised unigram tagger was trained to tag the induced clusters with linguistically motivated "gold-standard" tags. The accuracy of the meta-model is assumed to be an indicator for the precision of the induced classification.[14] For both languages, a reduced tagset was used as the gold-standard for meta-tagging. For English, the tagset reduction scheme given in Table 1 was used. For German, the tagset reduction scheme given in Table 2 was applied.

The use of an additional model to evaluate the induced classification is undesirable for a number of reasons, primarily because the meta-model introduces an additional potential source of error. For this reason, the simplest meta-model – a unigram model – was chosen as a meta-tagger. While certainly a limited model, its capabilities and limitations are

[14] Non-targets were treated as incorrect tag assignments.

| Tag | STTS Tag(s) | Description |
|---|---|---|
| ADJ | ADJA ADJD PIDAT | Adjective |
| ADV | ADV APPO APZR PAV PROAV PTKA PTKANT PTKNEG PTKVZ PWAV | Adverb |
| CARD | CARD | Cardinal number |
| CCONJ | KON | Coordinating conjunction |
| DET | ART PDAT PIAT PPOSAT PRELAT PWAT | Determiner |
| MISC | FM ITJ XY | Miscellaneous |
| NOUN | NE NN TRUNC | Nominal |
| PREP | APPRART APPR | Preposition |
| PRON | PDS PIS PPER PPOSS PRELS PRF PWS | Pronominal |
| SCONJ | KOKOM KOUI KOUS | Subordinating conjunction |
| TO | PTKZU | Infinitival *zu* |
| VFIN | VAFIN VAIMP VMFIN VVFIN VVIMP | Finite verb |
| VINF | VAINF VAPP VMINF VMPP VVINF VVIZU VVPP | Infinitive, participle |
| $, | $, | Comma |
| $. | $. | Sentence-final punctuation |
| $( | $( | Sentence-internal punctuation |

Table 2: Tagset reduction scheme used for the STTS source tagset.

well known, and the data it returns is used solely to compare variants of the classification procedure.

More difficult to address are principled rejections of the comparison of an induced classification with a linguistically motivated hand-annotated tagset, such as the argument given by Clark (2001). The core of the argument is that an induced classification may well reveal real and useful distributional properties of the target language which are not encoded in the linguistic-theoretically motivated corpus used as a gold-standard, thus rendering the results of any comparison between the two meaningless. The use of a reduced tagset for meta-tagging is intended to address these concerns by keeping theoretical bias to a minimum.

## 4.1 Clustering Phase

The multi-stage clustering procedure described above was first compared to a number of single-stage procedures each using a fixed set of boundary words. Data from these experiments are presented in Figure 1. From these data, it is easy to see that the multi-stage approach is considerably more stable for a large number of targets than any of the single-stage procedures to which it was compared. This fact is believed to be a direct consequence of the integration of previous clustering solutions into the feature selection process for the attachment stages, thus providing a workaround for the well-known sparse data problem.

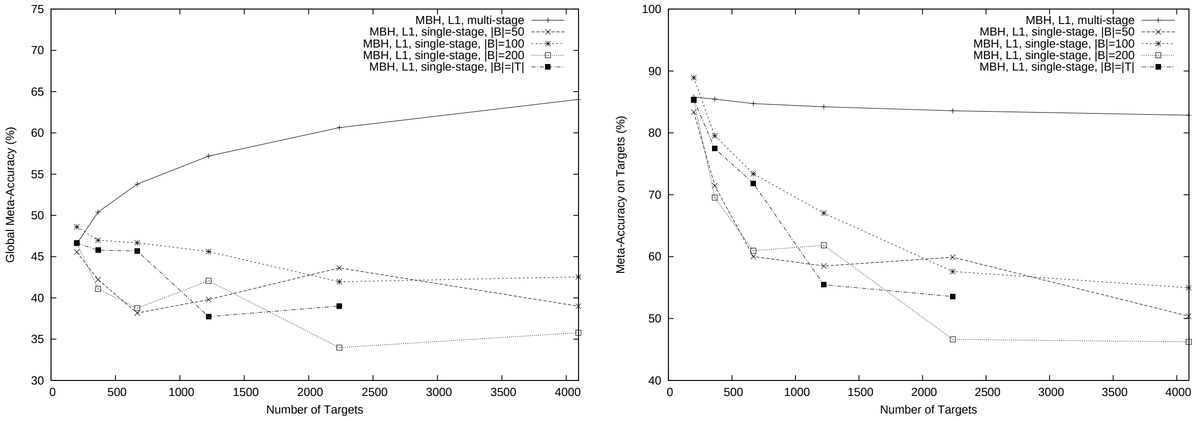

Figure 1: Multi- *vs.* single-stage clustering meta-accuracy for German, for all tokens (left) and for targets only (right). Hardware limitations prevented collection of the full range of data for the single-stage condition $|B| = |T|$, in which target and boundary sets were considered identical.

Use of the fuzzy membership heuristic provided only a comparatively small additional improvement. Happily, the multi-stage procedure also has much more modest time and space requirements (linear, as opposed to quadratic) than a single-stage procedure, as discussed in Section 3.3.

Meta-tagging accuracy for selected combinations of feature instantiation method, distance function, and clustering link method is shown for all test-corpus tokens in Figure 2, and for clustering targets in Figure 3. For both English and German, pairwise average linkage using the vector cosine performed most poorly, and maximum-link clustering using Spearman's rank correlation between monotonic Bernoulli entropy vectors performed best, resulting in a meta-tagging accuracy on targets of 81.56% after the final clustering stage for German, and 80.12% for English. Interesting to note is that for English, Finch and Chater's (1993) method performed similarly well, returning a meta-tagging accuracy on targets of 79.71% after the final stage, while this configuration resulted in a target meta-tagging accuracy of only 76.17% for the German corpus.

This discrepancy may stem from the relatively fixed word order of English compared to German, in that highly predictive, high-probability boundary elements are more likely to occur only to one side of an English target word, while such bounds may be distributed among both left and right contexts in German. German morphology – in particular its high productivity with respect to nominal composites – may also play a role in the poor performance of empirical conditional probabilities here, as it leads to a large alphabet size and thus compounds the sparse data problem: the German corpus contained over twice as many word types as the English corpus, despite the fact that the corpora were of similar size in terms of tokens.

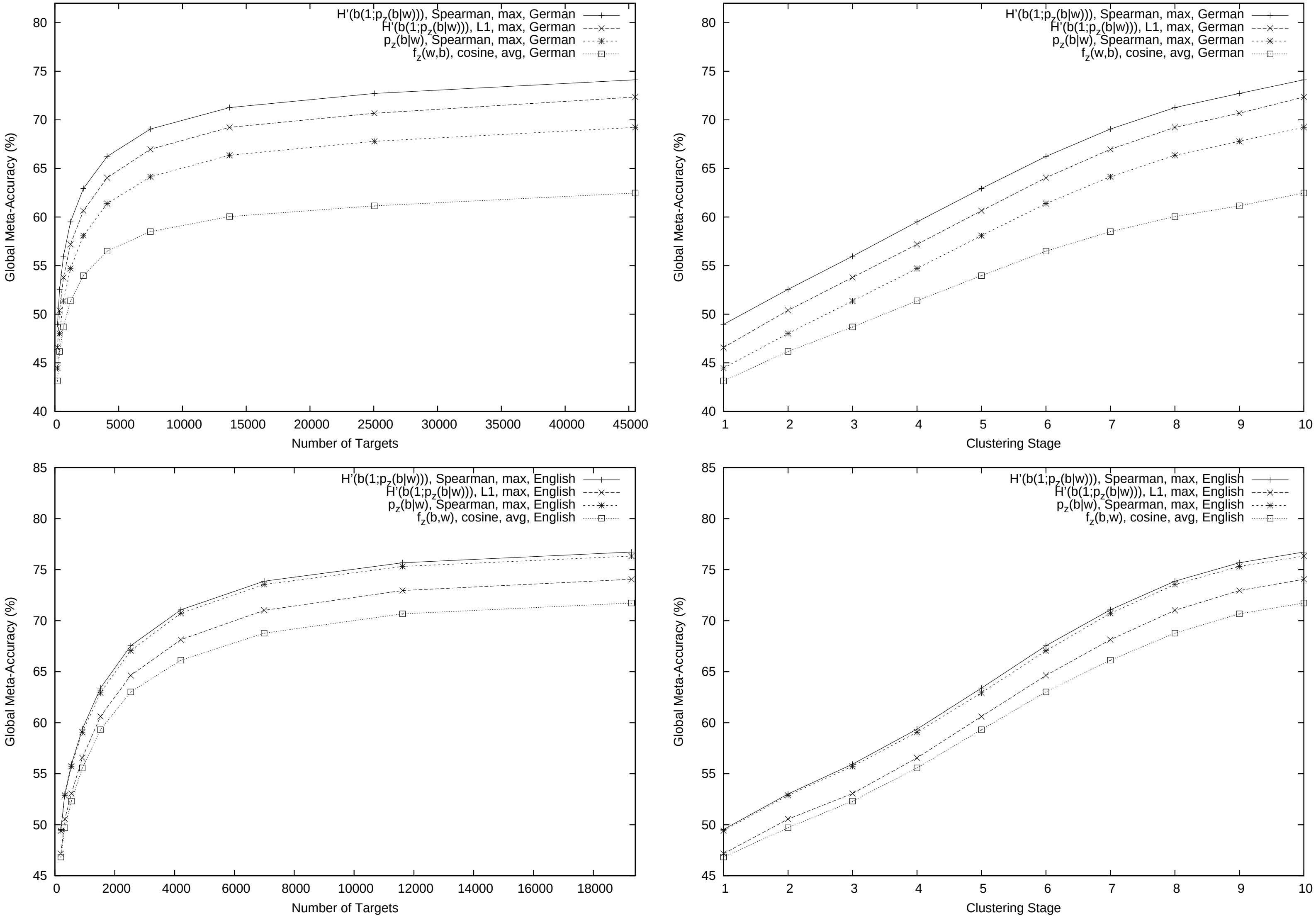

Figure 2: Meta-accuracy for selected clustering configurations for German (top)and English (bottom), by number of targets (left) and by clustering stage (right).

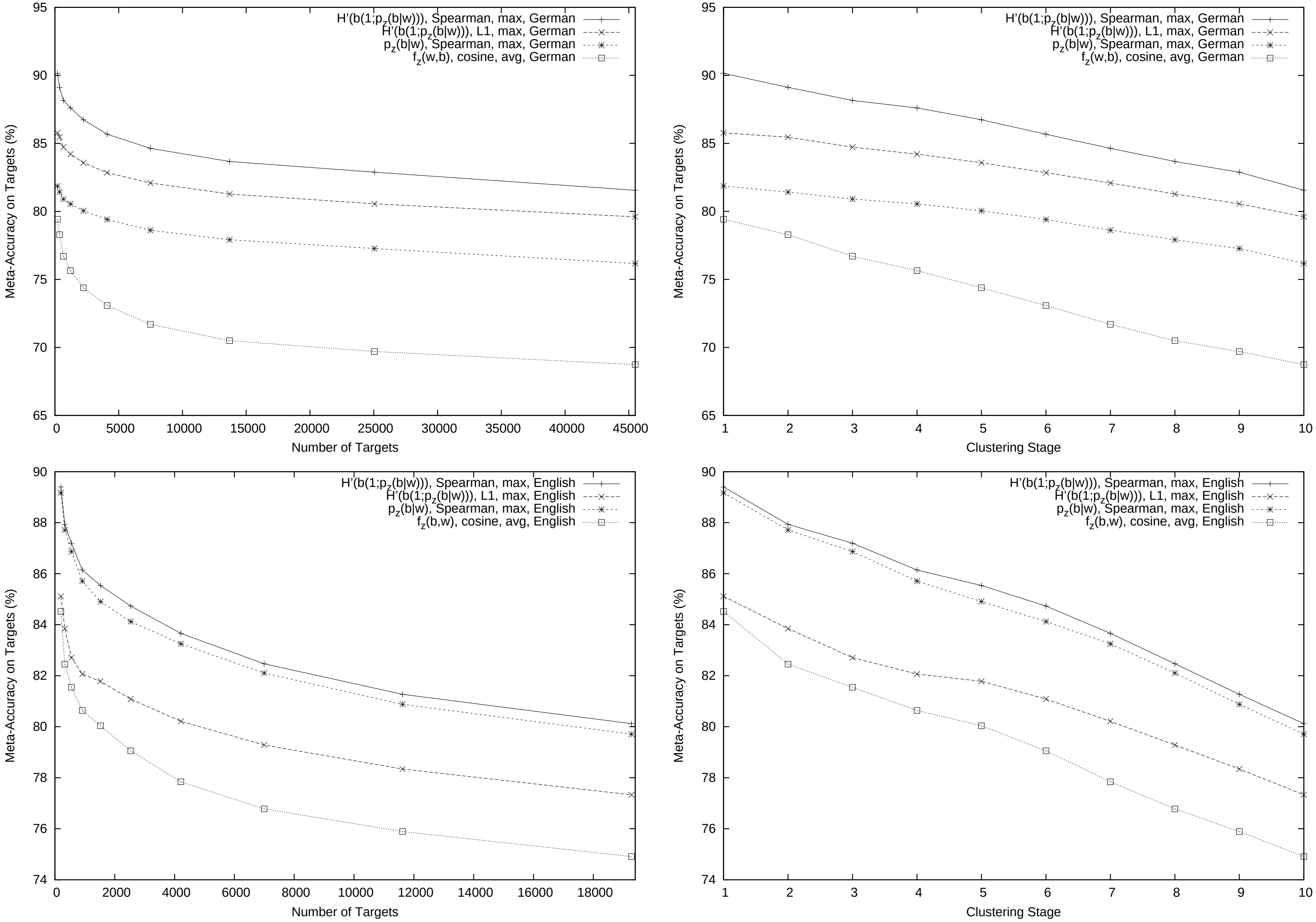

Figure 3: Meta-accuracy on targets for selected clustering configurations for German (top) and for English (bottom), by number of targets (left) and by clustering stage (right).

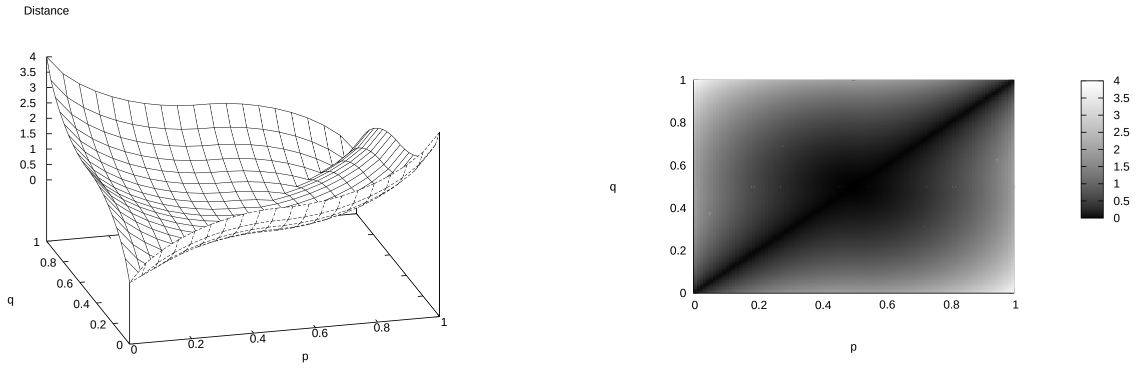

Figure 4: $d_{L1}(\hat{\mathrm{H}}(p), \hat{\mathrm{H}}(q))$: 1-norm distance between monotonic Bernoulli entropy values for probability parameters $p$ and $q$.

Use of monotonic Bernoulli entropy to populate target vectors performs similarly well for both German and English, which supports the hypothesis that it is capable of capturing linguistically useful distributional properties. In Section 3.1.2.1, it was suggested that monotonic Bernoulli entropy could be intuitively understood as an estimator for the mnemonic utility of “chunking” a boundary event into a target event, since it is sensitive both to absolute probability value and to outcome predictability.

The $L1$ distance between monotonic Bernoulli entropies is presented as a surface projection and as a palette map in Figure 4. From these presentations, it is easy to see that small probability differences result in particularly large distance differences near the extrema 0 and 1, thus expressing $\hat{\mathrm{H}}$’s sensitivity to bounds as predictors. Sensitivity to absolute probability value is expressed by the asymmetries of the $p$ and $q$ axes. The “pocket” in which little differentiation is made – roughly in the range $0.4 \leq p, q \leq 0.6$ – is unlikely to be occupied in the case of the empirical distributions under consideration, providing an *a posteriori* account of the method’s utility.

## 4.2 Ambiguity Resolution Phase

The primary goal of the ambiguity resolution phase was to recover token-level ambiguity on a context-dependent basis. In addition to meta-tagging accuracy therefore, the average ambiguity rates for each of the ambiguity resolution methods were considered. Ambiguity rates were computed as the number of $\langle word, analysis \rangle$ pairs in the test corpus as tagged by the ambiguity resolving method under consideration divided by the total number of word-types in the test corpus. Input to the ambiguity resolution procedures was the membership probability distribution $\hat{p}_k$ returned by the final or some intermediate clustering stage. Ambiguity resolution was tested only for maximum-link word-type clustering us-

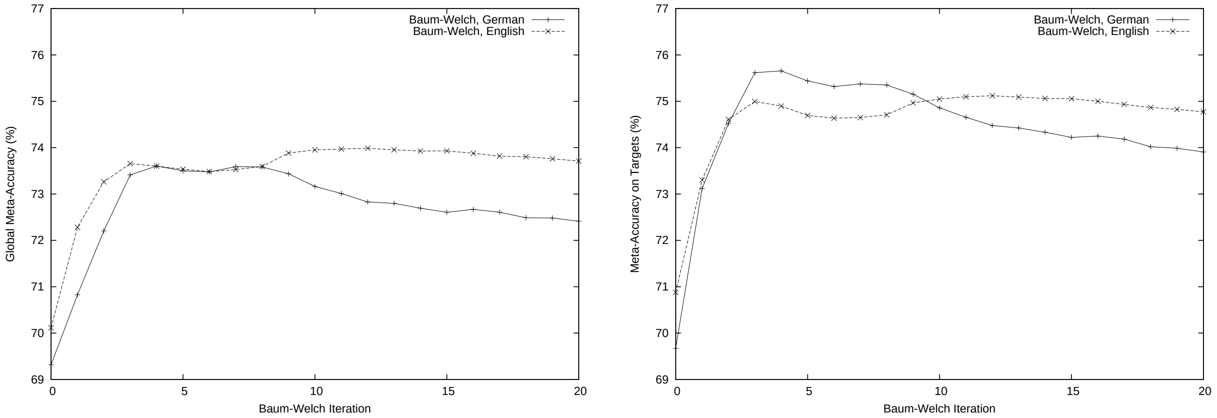

Figure 5: Meta-tagging accuracy during HMM reestimation

.

| **Baum-Welch** | **Global** | | **Targets** | |
|---|---|---|---|---|
| **Iteration** | **Amb. Rate** | **Meta-Acc.** | **Amb. Rate** | **Meta-Acc.** |
| $\hat{p}_K(c\|w)$ | *1.00* | ***74.13 %*** | *1.00* | ***81.56 %*** |
| 0 | 1.00 | 69.32 % | 1.00 | 69.67 % |
| 4 | 1.22 | **73.60 %** | 1.31 | **75.66 %** |
| 8 | 1.29 | 73.58 % | 1.40 | 75.35 % |
| 12 | 1.32 | 72.83 % | 1.46 | 74.48 % |
| 16 | 1.34 | 72.67 % | 1.49 | 74.25 % |
| 20 | 1.35 | 72.41 % | 1.50 | 73.91 % |

Table 3: Ambiguity resolution performance of HMM reestimation for the German corpus.

ing Spearman's rank correlation distance between context vectors of monotonic Bernoulli entropies. The training and test corpora were the same as those used in the clustering phase.

### 4.2.1 Hidden Markov Model Reestimation

Baum-Welch reestimation of first-order Hidden Markov Models initialized as described in Section 3.2.1 do indeed provide an increasing degree of context-dependent token-level ambiguity resolution, but performed poorly with respect to meta-tagging accuracy.[15] Selected data for Hidden Markov Model reestimation ambiguity resolution are given in Tables 3 and 4, and graphically presented in Figure 5.

[15] The Baum-Welch algorithm for HMM reestimation was applied to the test corpus. For model evaluation, the Viterbi (1967) algorithm was used to determine the optimal tag sequence for each input sentence.

| **Baum-Welch** | **Global** | | **Targets** | |
|---|---|---|---|---|
| **Iteration** | **Amb. Rate** | **Meta-Acc.** | **Amb. Rate** | **Meta-Acc.** |
| $\hat{p}_K(c\|w)$ | *1.00* | ***76.72 %*** | *1.00* | ***80.12 %*** |
| 0 | 1.00 | 70.11 % | 1.00 | 70.88 % |
| 4 | 1.37 | 73.60 % | 1.46 | 74.90 % |
| 8 | 1.48 | 73.60 % | 1.59 | 74.71 % |
| 12 | 1.56 | **73.99 %** | 1.68 | **75.12 %** |
| 16 | 1.61 | 73.88 % | 1.75 | 75.00 % |
| 20 | 1.65 | 73.71 % | 1.80 | 74.77 % |

Table 4: Ambiguity resolution performance of HMM reestimation for the English corpus.

Unfortunately, none of the reestimated models exceeded the meta-tagging accuracy of the final clustering phase (labelled $\hat{p}_K(c|w)$ in Tables 3 and 4). The large discrepancy (ca. 5%) in meta-tagging accuracy between unambiguous cluster assignment and the iteration-0 HMM initialized only with emission probabilities is particularly unintuitive. One major cause of this phenomenon are the large number of zero probabilities introduced by the ambiguity-limiting parameter $m$, which result in the allocation of too great a probability mass to small clusters during the initialization of emission probabilities by Equation 34.[16]

Overall, the pattern displayed by reestimation of the HMMs initialized with clustering output distributions appears to correspond to that which Elworthy (1994) termed "early maximum". This may be due to the fact that the fuzzy membership heuristic described in Section 3.1.2.5 is too lenient, leading the "expect" step of the Baum-Welch algorithm to assign greater expected frequencies than are desired to non-optimal $\langle word, cluster \rangle$ pairs. Additional experiments employing an intermediate "shock-freezing" step to stabilize the cluster membership distribution[17] prior to HMM initialization produced good unambiguous initial models, but showed a clear "initial maximum" pattern for reestimation, with meta-tagging accuracy dropping ca. 15% after the first Baum-Welch iteration.

Another likely and deeper-lying source of error are the independence assumptions given in Equation 31 used during the clustering phase to approximate word-type ambiguity, and used implicitly in Equation 34 in the initialization of the Hidden Markov Model emission probabilities. These assumptions – that a cluster (state) is independent of neighboring words (emission symbols) given the current word – are simply not compatible with the assumptions required for first-order hidden Markov modelling, according to which the probability of a state depends not only on the emitted symbol, but also on the preceding state. It is therefore doubtful whether the clustering procedure can be made compatible

[16] Additional experiments in which the ambiguity limit was discarded for the reestimation phase supported this claim, but the resulting models performed even more poorly during Baum-Welch reestimation than those for which the ambiguity limit was applied.

[17] By the same physical analogy responsible for "simulated annealing", "shock freezing" was computed as: $\hat{p}'_k(c|w) = \hat{p}_k(c|w)^{\beta'}/Z'_{k,w}$, for an inverse temperature parameter $\beta' > 1$.

| Clustering Stage | German |  | English |  |
|---|---|---|---|---|
|  | **Amb. Rate** | **Meta-Acc.** | **Amb. Rate** | **Meta-Acc.** |
| $\hat{p}_K(c\|w)$ | *1.00* | *74.13 %* | *1.00* | ***76.72 %*** |
| 7 | 1.25 | 73.57 % | 1.52 | 73.86 % |
| 8 | 1.23 | 75.25 % | 1.53 | 74.57 % |
| 9 | 1.24 | **77.08 %** | 1.51 | 74.49 % |
| 10 | 1.25 | 75.59 % | 1.55 | **74.83 %** |

Table 5: Global meta-accuracy and ambiguity rates for trigram clustering.

with first-order HMM assumptions without directly estimating transition probabilities prior to HMM initialization.

### 4.2.2 Trigram Clustering

The trigram clustering technique proposed by Schütze (1995) — which might be indirectly used to provide better estimates of HMM transition probabilities — was also implemented and evaluated as a method for context-dependent token-level ambiguity resolution. The German corpus contained 311,389 distinct trigram types with nonzero frequency, and the English corpus contained 276,604. Of these, 4000 trigram types were selected for prototype clustering based on their components' co-occurrence frequencies with previously clustered targets.

Component context vectors were constructed by expected frequency profiling as in the word-type clustering attachment stages. The resulting $4\mathcal{C} \times 4000$ matrix was then clustered with an agglomerative hierarchical maximum-link clustering procedure using Spearman's rank correlation distance. The resulting tree was cut to produce 50 clusters. Each of the remaining trigram types from the test corpus was then profiled and assigned to the nearest cluster centroid. For evaluation purposes, the cluster assigned to a trigram $w_1w_2w_3 \in \mathcal{A}^3$ was interpreted as an induced syntactic category for the token $w_2$ in the immediate context $w_1\_w_3$.

Meta-tagging accuracies for trigram clustering after selected word-type clustering stages are given in Table 5. The label $\hat{p}_K(c|w)$ indicates the unambiguous baseline output of the word-type clustering stage. No clear conclusions can be immediately drawn from these data. While token-level ambiguity resolution was reintroduced by trigram clustering, this often occurred at the expense of global meta-tagging accuracy. Clear gains in meta-tagging accuracy were observed for the German corpus, but in the case of the English corpus, the unambiguous output of the word-type clustering phase outperformed all of the trigram clustering configurations tested. It is believed that a more sophisticated technique for the selection of prototype trigrams may be of help in this regard.

# 5 Outlook

A two-phase method was presented for learning a function which assigns induced syntactic categories to individual word tokens which combines an iterative clustering method over word types with existing methods for inducing context-dependent token-level ambiguity resolution. Iterative application of standard clustering techniques using the output of previous stages in a simulated melting procedure was shown to be both more efficient and more stable over a larger number of target words with respect to meta-tagging accuracy than previously employed single-pass clustering methods.

Baum-Welch reestimation for Hidden Markov Models initialized with cluster membership probabilities did indeed recover token-level ambiguity resolution, but many of the ambiguities induced by this method were spurious, causing it to display an "early maximum" pattern in the sense of Elworthy (1994). Trigram clustering using the output of the word clustering stage for component profiling also reintroduced token-level ambiguity resolution, but only exceeded the baseline meta-tagging accuracy of the word-type clustering stage for the German corpus.

Future research will investigate more sophisticated methods for trigram prototype selection, as well as additional combinations of the clustering and ambiguity resolution techniques described above. Interpolation of reestimated Hidden Markov Models with a cluster unigram model estimated from the clustering phase's membership distribution may mitigate the sharp drop in meta-tagging accuracy for iteration-0 HMMs. Alternately, trigram clustering over the training corpus might provide better initial estimates of HMM transition probabilities prior to reestimation, thus mitigating the effects of unwarranted independence assumptions in the word-type clustering phase.

# References

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

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

K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of ANLP-1988*, pages 136–143, 1988.

A. Clark. Inducing syntactic categories by context distribution clustering. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 91–94, 2000.

A. Clark. *Unsupervised Language Acquisition: Theory and Practice.* PhD thesis, University of Sussex, 2001.

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proceedings of ANLP-1992*, 1992a.

D. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992b.

M. J. L. de Hoon, S. Imoto, J. Nolan, and S. Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004.

A. P. Dempster, N. M. Laird., and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

S. DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39, 1988.

C. Dickens. *Great Expectations.* Project Gutenberg, Champaign, IL, May 2002. URL `http://www.gutenberg.org/etext/1400`. Originally published 1861.

K. Elghamry. *A Generalized Cue-Based Approach to the Automatic Acquisition of Subcategorization Frames.* PhD thesis, Indiana University, Bloomington, Indiana, 2004.

D. Elworthy. Does Baum-Welch re-estimation help taggers? In *Proceedings of ANLP-1994*, 1994.

S. Finch and N. Chater. Learning syntactic categories: a statistical approach. In *Neurodynamics and Psychology*, pages 295–322. Harcourt Brace, London, 1993.

I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), July 1989.

J. Hughes. *Automatically Acquiring a Classification of Words.* PhD thesis, School of Computer Studies, University of Leeds, 1994.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

E. T. Jaynes. Brandeis lectures. In *E. T. Jaynes: Papers on Probability, Statistics and Statistical Physics*, pages 40–76. D. Reidel, Dordrecht, 1983.

E. E. Korkmaz and G. Üçoluk. A method for improving automatic word categorization. In *Proceedings of CoNLL97*, Madrid, Spain, July 1997.

E. E. Korkmaz and G. Üçoluk. Choosing a distance metric for automatic word categorization. In D. M. W. Powers, editor, *Proceedings of NeMLaP3/CoNLL98*, pages 111–120. Association for Computational Linguistics, Somerset, New Jersey, 1998.

L. Lee. *Similary-Based Approaches to Natural Language Processing.* PhD thesis, Harvard University, Cambridge, MA, 1997.

J. G. McMahon and F. J. Smith. Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics*, 22(2):217–247, 1996.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *ACL 31*, pages 183–190, 1993.

L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

A. Roberts. Automatic acquisition of word classification using distributional analysis of content words with respect to function words. Technical report, School of Computing, University of Leeds, 2002.

G. R. Sampson. *English for the Computer: The SUSANNE Corpus and Analytic Scheme.* Clarendon Press, 1995.

H. Schütze. Part-of-Speech induction from scratch. In *ACL 31*, pages 251–258, 1993.

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

C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, IL, 1949.

W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of ANLP-97*, Washington, DC, 1997.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, pages 260–269, April 1967.

G. K. Zipf. *Human Behaviour and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA, 1949.