Improved Unsupervised POS Induction Using Intrinsic Clustering Quality and a Zipfian Constraint

Roi Reichart
ICNC
The Hebrew University
roiri@cs.huji.ac.il

Raanan Fattal
Institute of computer science
The Hebrew University
raananf@cs.huji.ac.il

Ari Rappoport
Institute of computer science
The Hebrew University
arir@cs.huji.ac.il

Abstract

Modern unsupervised POS taggers usually apply an optimization procedure to a non-convex function, and tend to converge to local maxima that are sensitive to starting conditions. The quality of the tagging induced by such algorithms is thus highly variable, and researchers report average results over several random initializations. Consequently, applications are not guaranteed to use an induced tagging of the quality reported for the algorithm.

In this paper we address this issue using an unsupervised test for intrinsic clustering quality. We run a base tagger with different random initializations, and select the best tagging using the quality test. As a base tagger, we modify a leading unsupervised POS tagger (Clark, 2003) to constrain the distributions of word types across clusters to be Zipfian, allowing us to utilize a perplexity-based quality test. We show that the correlation between our quality test and gold standard-based tagging quality measures is high. Our results are better in most evaluation measures than all results reported in the literature for this task, and are always better than the Clark average results.

1 Introduction

Unsupervised part-of-speech (POS) induction is of major theoretical and practical importance. It counters the arbitrary nature of manually designed tag sets, and avoids manual corpus annotation costs. The task enjoys considerable current interest in the research community (see Section 3).

Most unsupervised POS tagging algorithms apply an optimization procedure to a non-convex function, and tend to converge to local maxima that strongly depend on the algorithm's (usually random) initialization. The quality of the taggings produced by different initializations varies substantially. Figure 1 demonstrates this phenomenon for a leading POS induction algorithm (Clark, 2003). The absolute variability of the induced tagging quality is 10-15%, which is around 20% of the mean. Strong variability has also been reported by other authors (Section 3).

The common practice in the literature is to report mean results over several random initializations of the algorithm (e.g. (Clark, 2003; Smith and Eisner, 2005; Goldwater and Griffiths, 2007; Johnson, 2007)). This means that applications using the induced tagging are not guaranteed to use a tagging of the reported quality.

In this paper we address this issue using an unsupervised test for intrinsic clustering quality. We present a quality-based algorithmic family Q. Each of its concrete member algorithms Q(B) runs a base tagger B with different random initializations, and selects the best tagging according the quality test. If the test is highly positively correlated with external tagging quality measures (e.g., those based on gold standard tagging), Q(B) will produce better results than B with high probability.

We experiment with two base taggers, Clark's original tagger (CT) and *Zipf Constrained Clark (ZCC)*. ZCC is a novel algorithm of interest in its own right, which is especially suitable as a base tagger in the family Q. ZCC is a modification of Clark's algorithm in which the distribution of the number of word types in a cluster (*cluster type size*) is constrained to be Zipfian. This property holds for natural languages, hence we can expect a higher correlation between ZCC and an accepted unsupervised quality measure, perplexity.

We show that for both base taggers, the correlation between our unsupervised quality test and gold standard based tagging quality measures is high. For the English WSJ corpus, the Q(ZCC)

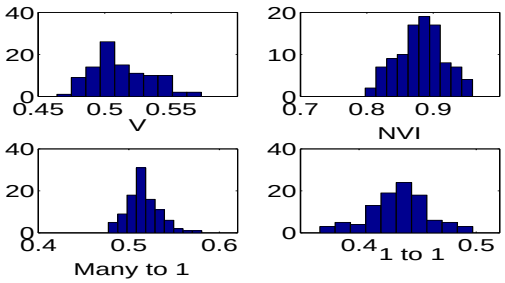

Figure 1: Distribution of the quality of the taggings produced in 100 runs of the Clark POS induction algorithm (with different random initializations) for sections 2-21 of the WSJ corpus. All graphs are 10-bin histograms presenting the number of runs (y-axis) with the corresponding quality (x-axis). Quality is evaluated with 4 clustering evaluation measures: V, NVI, greedy m-1 mapping and greedy 1-1 mapping. The quality of the induced tagging varies considerably.

algorithm gives better results than CT with probability 82-100% (depending on the external quality measure used). Q(CT) is shown to be better than the original CT algorithm as well. Our results are better in most evaluation measures than all previous results reported in the literature for this task, and are always better than Clark's average results.

Section 2 describes the ZCC algorithm and our quality measure. Section 3 discusses previous work. Section 4 presents the experimental setup and Section 5 reports our results.

2 The Q(ZCC) Algorithm

Given an N word corpus M consisting of plain text, with word types $W = \{w_1, \ldots, w_m\}$, the *unsupervised POS induction* task is to find a class membership function g from W into a set of class labels $\{c_1, \ldots, c_n\}$. In the version tackled in this paper, the number of classes n is an input of the algorithm. The membership function g can be used to tag a corpus if it is deterministic (as the function learned in this work) or if a rule for selecting a single tag for every word is provided.

Most modern unsupervised POS taggers produce taggings of variable quality that strongly depend on their initialization. Our approach towards generating a single high quality tagging is to use a family of algorithms Q. Each member Q(B) of Q utilizes a base tagger B, which is run using several random initializations. The final output is selected according to an unsupervised quality test. We focus here on Clark's tagger (Clark, 2003) (CT), probably the leading POS induction algorithm (see Table 3).

We start with a description of the original CT. We then detail ZCC, a modification of CT that constrains the clustering space by adding a Zipf-based constraint. Our perplexity-based unsupervised tagging quality test is discussed next. Finally, we provide an unsupervised technique for selecting the parameter of the Zipfian constraint.

2.1 The Original Clark Tagger (CT)

The tagger's statistical model combines distributional and morphological information with the likelihood function of the Brown algorithm (Brown et al., 1992; Ney et al., 1994; Martin et al., 1998). In the Brown algorithm a class assignment function g is selected such that the class bigram likelihood of the corpus, $p(M|g)$, is maximized. Morphological and distributional information is introduced to the Clark model through a prior $p(g)$. The prior prefers morphologically uniform clusters and skewed cluster sizes.

The probability function the algorithm tries to maximize is:

(1) $p(M, g) = p(M|g) \cdot p(g)$

(2) $p(M|g) = \prod_{i=2}^{i=N} p(g(w_i)|g(w_{i-1}))$

(3) $p(g) = \prod_{j=1}^{n} \alpha_j \prod_{g(w)=j} q_j(w)$

Where $q_j(w_i)$ is the probability of assigning $w_i \in W$ by cluster c_j according to the morphological model and α_j is the coefficient of cluster j, which equals to the number of word types assigned to that cluster divided by the total number of word types in the vocabulary W. The objective of the algorithm is formally specified by:

$$g^* = argmax_g p(M, g)$$

To find the cluster assignment g^* an iterative algorithm is applied. As initialization, the words in W are randomly assigned to clusters (clusters are thus of similar sizes). Then, for each word (words are ordered by their frequency in the corpus) the algorithm computes the effect that moving it from its current cluster to each of the other clusters would have on the probability function. The word is moved to the cluster having the highest positive effect (if there is no such cluster, the word is not moved). The last step is performed iteratively until no improvement to the probability function is possible through a single operation.

The probability function has many local maxima and the one to which the algorithm convergences strongly depends on the initial assignment of words to clusters. The quality of the clusters induced in different runs of the algorithm is highly variable (Figure 1).

2.2 The Cluster Type Size Zipf Constraint

The motivation behind using a Zipfian constraint is the following observation: when a certain statistic is known to affect the quality of the induced clustering and it is not explicitly manipulated by the algorithm, strong fluctuations in its values are likely to imply that there are uncontrolled fluctuations in the quality of the induced clusterings. Thus, introducing a constraint that we believe holds in real data increases the correlation between clustering quality and a well accepted unsupervised quality measure (perplexity).

Our ZCC algorithm searches for a class assignment function g that maximizes the probability function (1) under a constraint on the clustering space, namely constraining the cluster type size distribution induced by g to be Zipfian. This constraint holds in many languages (Mitzenmacher, 2004) and is demonstrated in Figure 3 for the English corpus with which we experiment in this paper.

Zipf's law predicts that the fraction of elements in class k is given by:

$$f(k; s; n) = \frac{1/k^s}{\sum_{i=1}^{n}(1/i^s)}$$

where s is a parameter of the distribution and n the number of clusters.

Denote the cluster type size distribution derived from the algorithm's cluster assignment function g by $T(g)$. The objective of the algorithm is

$$g^{**} = argmax_g p(M, g)\ s.t.\ T(g) \sim Zipf(s)$$

To impose the Zipfian distribution on the induced clusters size, we make two modifications to the original CT algorithm. First, at initialization, words are randomly assigned to clusters in a way that cluster sizes are distributed according to the Zipfian distribution (with a parameter s). Specifically, we randomly select words to be assigned to the first cluster until the fraction of word types in the cluster equals to the prediction given by Zipf's law. We then randomly assign words to the second cluster and so on.

Second, we change the basic operation of the algorithm from moving a word to a cluster to swapping two words between two different clusters. For each word w_i (again, words are ordered by their frequency in the corpus as in CT), the algorithm computes the effect on the probability function of moving it from its current cluster c_{curr} to each of the other clusters. We denote the cluster showing the best effect by c_{best}. Then, we search the words of c_{best} for the word w_j whose transition to c_{curr} has the best effect on the probability function. If the sum of the effects of moving w_i from c_{curr} to c_{best} and moving w_j from c_{best} to c_{curr} is positive, the swapping is performed. If swapping is not performed, we repeat the process for w_i, this time searching for c_{best} among all other clusters except of former c_{best} candidates[1].

2.3 Unsupervised Identification of High Quality Runs

Perplexity is a standard measure for language model evaluation. A language model defines the transition probabilities for every word w_i given the words that precede it. The perplexity of a language model for a given corpus having N words is defined to be

$$\sqrt[N]{\prod_{i=1}^{N} \frac{1}{p(w_i|w_1 \ldots w_{i-1})}}$$

An important property of perplexity that makes it attractive as a measure for language model performance is that in some sense the best model for any corpus has the lowest perplexity for that corpus (Goodman, 2001). Thus, the lower the perplexity of the language model, the better it is.

Clark (2003) proposed a perplexity based test for the quality of his POS induction algorithm. In that test, a bigram class-based language model is trained on a training corpus (using the tagging of the unsupervised tagger) and applied to another test corpus. In such a model the transition probability from a word w_j to a word w_i is given by $p(C(w_i)|C(w_j))$ where $C(w_k)$ is the class assigned by the POS induction algorithm to w_k. In the training phase the bigram transition probabilities are computed using the training corpus, and in

[1] To make the algorithm more time efficient, for each word w_i we perform only three iterations of the searching for c_{best}, and for each c_{best} candidate we compute for at most 500 words the effect on the probability function of the removal to c_{curr}.

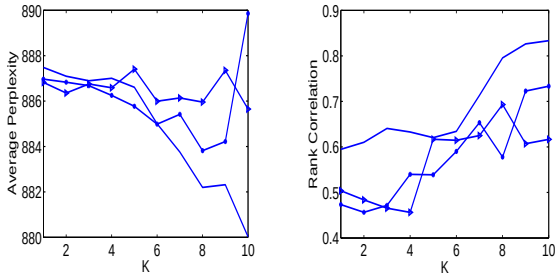

Figure 2: Left: average perplexity vs. the parameter K (tightness of the entropy outliers filter; see text for a full explanation). Right: Spearman's rank correlation between perplexity and an external (many-to-one) quality of the clustering as a function of K. The three curves are for ZCC, using different exponents (triangles: 0.9, circles: 1.3, solid: 1.1). A model whose quality improves (decreased perplexity) with K (left) demonstrates better correlation between perplexity and external quality (right). In all three graphs the x axis is in units of $5K$ (e.g., a graph x value of 2 means that 10 clusterings were removed from the top of the list and 10 from its bottom).

the test phase the perplexity of the learned model is evaluated on the test corpus. Better POS induction algorithms yield lower perplexity language models. However, Clark did not study the correlation between the perplexity measure and the gold standard tagging.

In this paper, we use Clark's perplexity based test as the unsupervised quality test used by the family Q. To provide a high quality prediction, this test should highly correlate with external clustering quality. To the best of our knowledge, such a correlation has not been explored so far.

2.4 Unsupervised Parameter Selection

The base ZCC algorithm has one input parameter, the exponent s of the Zipfian distribution. Virtually all unsupervised algorithms utilize parameters whose values affect their results. While it is methodologically valid to simply determine a value based on reasonable considerations or a development set, to keep the fully unsupervised nature of our work we now present a method for identifying the best parameter assignment. The method also casts some additional interesting light on the nature of the problem.

Like cluster type size, the distribution of cluster instance size in natural languages is also Zipfian (see Figure 3). A naive application of this constraint into the ZCC algorithm would be to allow swapping words between clusters only if they annotate the same number of word instances in the corpus. However, this constraint, either by itself or in combination with the cluster type size constraint, is too restrictive.

We utilize it for parameter selection as follows. Recall that our family of algorithms Q(B) runs a base tagger B several times. Each specific run yields a clustering C_i. The final result is selected from the set of clusterings $C = \{C_i\}$. We do not explicitly address the number of instances contained in a cluster, but we can prune from C those clusterings for which this distribution is very different. Again, imposing a constraint that is known to hold reduces quality fluctuations between different runs.

To measure the similarity between the cluster instance size distribution of two clusterings induced by two runs of the algorithm, we treat the clusters induced by a given run as samples from a random variable. The events of this variable are the induced clusters and the probability assigned to each event is equal to the number of word instances contained in the corresponding cluster, divided by the total number of word instances in the tagged corpus. The entropy of this random variable is used as a statistic for the word instance distribution. Clusterings having similar cluster instance size distributions also have similar values of this statistic.

We apply an entropy outliers filter to the set of clusterings C. In this filter, we sort the members of C (these are clusterings obtained in different runs of the base tagger) according to their cluster instance size entropy, and prune K runs from the beginning and K runs from the end of the list. The perplexity-based quality test described above is applied only to members of C that were not pruned in this step.

Figure 2 (left) shows the average perplexity of a set of clusterings as a function of the parameter K of the entropy-based filter. Results are presented for 100 runs of ZCC[2] with three different exponent values (0.9, 1.1, 1.3). These assignments yield considerably different Zipfian distributions.

While all three models have similar average perplexity over all 100 runs, only the solid line (corresponding to an exponent value of 1.1) consis-

[2] See Section 4 for the experimental setup.

tently decreases (improves) with K. The circled line (corresponding to an exponent value of 1.3) monotonically decreases with K until a certain K value, while the line with triangles (corresponding to an exponent value of 0.9) remains relatively constant.

Figure 2 (right) shows that models for which the entropy-based filter improves perplexity more drastically, exhibit better correlation between perplexity and external clustering quality[3].

Our unsupervised parameter selection method is thus based on finding a value which exhibits a consistent decrease in perplexity as a function of K, the number of clusterings pruned from the beginning and end of the entropy-sorted list. In the rest of this paper we show results where the exponent value is 1.1.

3 Previous Work

Unsupervised POS induction/tagging is a fruitful area of research. A major direction is Hidden Markov Models (HMM) (Merialdo, 1994; Banko and Moore, 2004; Wang and Schuurmans, 2005). Several recent works have tried to improve this model using Bayesian estimation (Goldwater and Griffiths, 2007; Johnson, 2007; Gao and Johnson, 2008), sophisticated initialization (Goldberg et al., 2008), induction of an initial clustering used to train an HMM (Freitag, 2004; Biemann, 2006), infinite HMM models (Van Gael et al., 2009), integration of integer linear programming into the parameter estimation process (Ravi and Knight, 2009), and biasing the model such that the number of possible tags that each word can get is small (Graça et al., 2009).

The Bayesian works integrated into the model information about the distribution of words to POS tags. For example, Johnson (2007) integrated to the EM-HMM model a prior that prefers clusterings where the distributions of hidden states to words is skewed.

Other approaches include transformation based learning (Brill, 1995), contrastive estimation for conditional random fields (Smith and Eisner, 2005), Markov random fields (Haghighi and Klein, 2006), a multilingual approach (Snyder et al., 2008; Snyder et al., 2008) and expanding a partial dictionary and use it to learn disambiguation rules (Zhao and Marcus, 2009).

These works, except (Haghighi and Klein, 2006; Johnson, 2007; Gao and Johnson, 2008) and one experiment in (Goldwater and Griffiths, 2007), used a dictionary listing the allowable tags for each word in the text. This dictionary is usually extracted from the manual tagging of the text, contradicting the unsupervised nature of the task. Clearly, the availability of such a dictionary is not always a reasonable assumption (see e.g. (Goldwater and Griffiths, 2007)).

In a different algorithmic direction, (Schuetze, 1995) applied latent semantic analysis with SVD based dimensionality reduction, and (Schuetze, 1995; Clark, 2003; Dasgupta and NG, 2007) used distributional and morphological statistics to find meaningful word types clusters. Clark (2003) is the only such work to have evaluated its algorithm as a POS tagger for large corpora, like we do in this paper.

A Zipfian constraint was utilized in (Goldwater and et al., 2006) for language modeling and morphological disambiguation.

The problem of convergence to local maxima has been discussed in (Smith and Eisner, 2005; Haghighi and Klein, 2006; Goldwater and Griffiths, 2007; Johnson, 2007; Gao and Johnson, 2008) with a detailed demonstration in (Johnson, 2007). All these authors (except Smith and Eisner (2005), see below), however, reported average results over several runs and did not try to identify the runs that produce high quality tagging.

Smith and Eisner (2005) initialized with all weights equal to zero (uninformed, deterministic initialization) and performed unsupervised model selection across smoothing parameters by evaluating the training criterion on unseen, unlabeled development data. In this paper we show that for the tagger of (Clark, 2003) such a method provides mediocre results (Table 2) even when the training criterion (likelihood or data probability for this tagger) is evaluated on the test set. Moreover, we show that our algorithm outperforms existing POS taggers for most evaluation measures (Table 3).

Identifying good solutions among many runs of a randomly-initialized algorithm is a well known problem. We discuss here the work of (Smith and Eisner, 2004) that addressed the problem in the unsupervised POS tagging context. In this work, deterministic annealing (Rose et al., 1990) was ap-

[3]The figure is for greedy many-to-one mapping and Spearman's rank correlation coefficient, explained in further Sections. Other external measures and rank correlation scores demonstrate the same pattern.

plied to an HMM model for unsupervised POS tagging with a dictionary. This method is not sensitive to its initialization, and while it is not theoretically guaranteed to converge to a better solution than the traditional EM-HMM, it was experimentally shown to achieve better results. The problem has, of course, been addressed in other contexts as well (see, e.g., (Wang et al., 2002)).

4 Experimental Setup and Evaluation

Setup. We used the English WSJ PennTreebank corpus in our experiments. We induced POS tags for sections 2-21 (43K word types, 950K word instances of which 832K (87.6%) are not punctuation marks), using Q(ZCC), Q(CT), and CT. For the unsupervised quality test, we trained the bigram class-based language model on sections 2-21 with the induced clusters, and computed its perplexity on section 23.

In Q(ZCC) and Q(CT), the base taggers were run a 100 times each, using different random initializations. In each run we induce 13 clusters, since this is the number of unique POS tags required to cover 98% of the word types in WSJ (Figure 3)[4]. Some previous work (e.g., (Smith and Eisner, 2005)) also induced 13 non-punctuation tags.

We compare the results of our algorithm to those of the original Clark algorithm[5]. The induced clusters are evaluated against two POS tag sets: one is the full set of WSJ POS tags, and the other consists of the non-punctuation tags of the first set.

Punctuation marks constitute a sizeable volume of corpus tokens and are easy to cluster correctly. Hence, evaluting against the full tag set that includes punctuation artificially increases the quality of the reported results, which is why we report results for the non-punctuation tag set. However, to be able to directly compare with previous work, we also report results for the full WSJ POS tag set. We do so by assigning a singleton cluster to each punctuation mark (in addition to the 13 clusters). This simple heuristic yields very high performance on punctuation, scoring (when all other terminals are assumed perfect tagging) 99.6% in 1-to-1 accuracy.

[4]Some words can get more than one POS tag. In the figure, for these words we increased the counters of all their possible tags.

[5]Downloaded from www.cs.rhul.ac.uk/home/alexc/RHUL/Downloads.html.

In addition to comparing the different algorithms, we compare the correlation between our tagging quality test and external clustering quality for both the original CT algorithm and our ZCC algorithm.

Clustering Quality Evaluation. The induced POS tags have arbitrary names. To evaluate them against a manually annotated corpus, a proper correspondence with the gold standard POS tags should be established. Many evaluation measures for unsupervised clustering against gold standard exist. Here we use measures from two well accepted families: mapping based and information theoretic (IT) based. For a recent discussion on this subject see (Reichart and Rappoport, 2009).

The mapping based measures are accuracy with greedy many-to-1 (M-1) and with greedy 1-to-1 (1-1) mappings of the induced to the gold labels. In the former mapping, two induced clusters can be mapped to the same gold standard cluster, while in the latter mapping each and every induced cluster is assigned a unique gold cluster.

After each induced label is mapped to a gold label, tagging accuracy is computed. Accuracy is defined to be the number of correctly tagged words in the corpus divided by the total number of words in the corpus.

The IT based measures we use are V (Rosenberg and Hirschberg, 2007) and NVI (Reichart and Rappoport, 2009). The latter is a normalization of the VI measure (Meila, 2007). VI and NVI induce the same order over clusterings but NVI values for good clusterings lie in $[0, 1]$. For V, the higher the score, the better the clustering. For NVI lower scores imply improved clustering quality. We use e as the base of the logarithm.

Evaluation of the Quality Test. To measure the correlation between the score produced by the tagging quality test and the external quality of a tagging, we use two well accepted measures: Spearman's rank correlation coefficient and Kendall Tau (Kendall and Dickinson, 1990). These measure the correlation between two sorted lists. For the computation of these measures, we rank the clusterings once according to the identification criterion and once according to the external quality measure.

The measures are given by the equations:

(6) $kendall - tau = \frac{2(n_c - n_d)}{r(r-1)}$

(7) $Spearsman = 1 - \frac{6\sum_{i=1}^{r} d_i^2}{r(r^2-1)}$

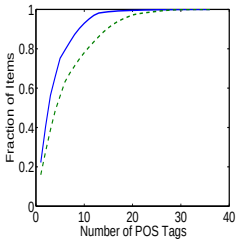

Figure 3: The fraction of word types (solid curve) and word instances (dashed curve) labeled with the k (X axis) most frequent POS tags (in types and tokens respectively) in sections 2-21 of the WSJ corpus.

where r is the number of runs (100 in our case), n_c and n_d are the numbers of concordant and discordant pairs respectively[6] and d_i is the absolute value of the difference between the ranks of item i.

The two measures have the properties that a perfect agreement between rankings results in a score of 1, a perfect disagreement results in a score of -1, completely independent rankings have the value of 0 on the average, the range of values is between -1 and 1, and increasing values imply increasing agreement between the rankings. For a discussion see (Lapata, 2006).

5 Results

Table 1 presents the results of the Q(ZCC) and Q(CT) algorithms, which are both better than those of the original Clark tagger CT. The Q algorithms provide a tagging that is better than that produced by CT in 82-100% (Q(ZCC)) and 75-100% (Q(CT)) of the cases.

The Q(ZCC) algorithm is superior when evaluated with the mapping based measures. The Q(CT) algorithm is superior when evaluated with the IT measures.

Table 3 presents reported results for all recent algorithms we are aware of that tackled the task of unsupervised POS induction from plain text [7]. The settings of the various experiments vary in terms of the exact gold annotation scheme used for evaluation (the full WSJ set was used by all authors except Goldwater and Griffiths (2007) and the GGTP-17 model which used the set of 17 coarse grained tags proposed by (Smith and Eisner, 2005)) and the size of the test set. The numbers reported for the algorithms of other works are the average performance over multiple runs, since no method for identification of high quality taggings was used.

The results of our algorithms are superior, except for the M-1 performance of some of the models of (Johnson, 2007) and of the GGTP-17 and GGTP-45 models of (Graça et al., 2009). Note that the models of (Johnson, 2007) and the GGTP-45 model induce 40-50 clusters compared to our 34 (13 non-punctuation plus the additional 21 singleton punctuation tags). Increasing the number of clusters is known to improve the M-1 measure (Reichart and Rappoport, 2009). GGTP-17 gives the best M-1 results, but its 1-1 results are much worse than those of Q(ZCC), Q(CT), and CT, and the information theoretic measures V and NVI were not reported for it.

Recall that the Q algorithms tag punctuation marks according to the scheme which assigns each of them a unique cluster (Section 4), while previous work does not distinguish punctuation marks from other tokens. To quantify the effect various punctuation schemes have on the results reported in Table 3, we evaluated the *'iHMM: PY-fixed'* model (Van Gael et al., 2009) and the Q algorithms when punctuation is excluded and when both PY-fixed and Q algorithms use the punctuation scheme described in Section 4.

For the PY-fixed, which induces 91 clusters, results are (punctuation is excluded, heuristic is used): V(0.530, 0.608), NVI (0.999, 0.823), 1-1 (0.484, 0.543), M-1 (0.591, 0.639). The results for the Q algorithms are given in Table 1 (top line: excluding punctuation, bottom line: using the heuristic). The Q algorithms are better for the V, NVI and 1-1 measures. For M-1 evaluation, PY-fixed, which induces substantially more clusters (91 compared to our 34) is better.

In what follows, we provide an analysis of the components of our algorithms. To explore the quality of our tagging component, ZCC, table 4 compares the mean, mode and standard deviation of a 100 runs of ZCC with 100 runs of the original CT algorithm[8]. The performance of the tagging

[6] A pair r, t in two lists X and Y is concordant if $sign(X_t - X_r) = sign(Y_t - Y_r)$, where X_r is the index of r in the list X.

[7] VG and GG used 2 as the base of the logarithm in IT measures, which affects VI. We converted the VI numbers reported in their papers to base e.

[8] In mode calculation we treat the 100 runs as samples of a continuous random variable. We divide the results range to 10 bins of the same size. The mode is the center of the bin having the largest number of runs. If there is more than

Alg.	V	NVI	1-1	M-1
Q(ZCC)				
no punct.	0.538 (85, 2.6)	0.849 (82, 3.2)	**0.521 (100, 4.3)**	**0.533 (84, 1.7)**
with punct.	0.637 (85, 1.8)	0.678 (82, 2.6)	**0.58 (100, 3)**	**0.591 (84, 1.18)**
Q(CT)				
no punct.	**0.545 (92, 3.3)**	**0.837 (88, 4.4)**	0.492 (99,1.4)	0.526 (75, 1)
with punct.	**0.644 (92, 2.5)**	**0.662 (88, 4.2)**	0.555 (99, 0.5)	0.585 (75, 0.58)

Table 1: Quality of the tagging produced by Q(ZCC) and Q(CT). The top (bottom) line for each algorithm presents the results when punctuation is not included (is included) in the evaluation (Section 4). The left number in the parentheses is the fraction of Clark's (CT) results that scored worse than our models (% from 100 runs). The right number in the parentheses is 100 times the difference between the score of our model and the mean score of 100 runs of Clark's (CT). Q(ZCC) is better than Q(CT) in the mappings measures, while Q(CT) is better in the IT measures. Both are better than the original Clark tagger CT.

	Data Probability				Likelihood				Perplexity			
	V		m-to-1		V		m-to-1		V		m-to-1	
Alg.	SRC	KT	SRC	KT	SRC	KT	SRC	KT	SRC	KT	SRC	KT
CT	0.2	0.143	0.071	0.045	0.338	0.23	0.22	0.148	**0.568**	**0.397**	**0.476**	**0.33**
ZCC	0.134	0.094	0.118	0.078	0.517	0.352	0.453	0.321	**0.82**	**0.62**	**0.659**	**0.484**

Table 2: Correlation of unsupervised quality measures (columns) with clustering quality of two base taggers (CT and ZCC, rows). Correlation is measured by Spearman (SRC) and Kendall Tau (KT) rank correlation coefficients. The quality measures are data probability (left part), likelihood (middle side) and perplexity (right part), and correlation is between these and two of the external evaluation measures, m-to-1 mapping and V (results for the other two clustering evaluation measures, 1-1 mapping and NVI, are very similar). Results for the perplexity quality test used by family Q are superior; data probability and likelihood provide only a mediocre indication for the quality of induced clustering. Note that the correlation values are much higher for ZCC than for CT.

components are quite similar, with a small advantage to CT in mean and to ZCC in mode.

Our quality test is based on the perplexity of a class bigram language model trained with the induced tagging. To emphasize its strength we compare it to two natural quality tests: the likelihood and value of the probability function to which the tagging algorithm converges (equations (2) and (1) in Section 2.1). The results are shown in Table 2 First, we see that our perplexity quality test is much better correlated with the quality of the tagging induced by both ZCC and CT. Second, the correlation is indeed much higher for ZCC than for CT.

The power of Q(ZCC) lies in the combination between the perplexity-based quality test and the tagging component ZCC. The performance of the tagging component ZCC does not provide a definite improvement over the original Clark tagger. ZCC compromises mean tagging results for an improved correlation between Q's quality measure and gold standard-based tagging evaluation.

6 Conclusion

In this paper we addressed unsupervised POS tagging as a task where the quality of a single tagging is to be reported, rather than the average performance of a tagging algorithm over many runs. We introduced a family of algorithms Q(B) based on an unsupervised test for tagging quality that is used to select a high quality tagging from the output of multiple runs of a POS tagger B.

We introduced the ZCC tagger which modifies the original Clark tagger by constraining the clustering space using a cluster type size Zipfian constraint, conforming with a known property of natural languages.

We showed that the tagging produced by our Q(ZCC) algorithm is better than that of the Clark algorithm with a probability of 82-100%, depending on the measure used. Moreover, our tagging outperforms in most evaluation measures the results reported in all recent works that addressed the task.

In future work, we intend to try to improve

one such bin, we average their centers. We use this technique since it is rare to see two different runs of either algorithm with the exact same quality.

Alg.	V	VI	M-1	1-1
Q(ZCC)	0.637	2.06	0.591	**0.58**
Q(CT)	**0.644**	**2.01**	0.585	0.555
CT	0.619	2.14	0.576	0.543
HK	–	–	–	0.413
J	–	4.23 - 5.74	0.43 - 0.62	0.37 – 0.47
GG	–	2.8	–	–
G-J	–	4.03 – 4.47	–	0.4 – 0.499
VG	0.54 - 0.59	2.49 – 2.91	–	–
GGTP-45	—	—	0.654	0.445
GGTP-17	—	—	**0.702**	0.495

Table 3: Comparison of our algorithms with the recent fully unsupervised POS taggers for which results are reported. HK: (Haghighi and Klein, 2006), trained and evaluated with a corpus of 193K tokens and 45 induced tags. GG: (Goldwater and Griffiths, 2007), trained and evaluated with a corpus of 24K tokens and 17 induced tags. J : (Johnson, 2007) inducing 25-50 tags (the results that are higher than Q in the M-1 measure are for 40-50 tags). GJ: (Gao and Johnson, 2008), inducing 50 tags. VG: (Van Gael et al., 2009), inducing 47-192 tags. GGTP-45: (Graça et al., 2009), inducing 45 tags. GGTP-17: (Graça et al., 2009), inducing 17 tags. All five were trained and evaluated with the full WSJ PTB (1.17M words). Lower VI values indicates better clustering.

Statistic	V	NVI	M-1	1-1
CT				
Mean	0.512	0.881	0.516	0.478
Mode	0.502	0.886	0.514	0.465
Std	0.022	0.035	0.018	0.028
ZCC				
Mean	0.503	0.908	0.512	0.478
Mode	0.509	0.907	0.518	0.47
Std	0.021	0.036	0.018	0.0295

Table 4: Average performance of ZCC compared with CT (results presented without punctuation). Presented are mean, mode (see text for its calculation), and standard deviation (std). CT mean results are slightly better, and both algorithms have about the same standard deviation. ZCC sacrifices a small amount of mean quality for a good correlation with our quality test, which allows Q(ZCC) to be much better than the mean of CT and most of its runs.

our quality measure, experiment with additional languages, and apply the 'family of algorithms' paradigm to additional relevant NLP tasks.

References

Michele Banko and Robert C. Moore, 2003. Part of Speech Tagging in Context. *COLING '04*.

Chris Biemann, 2006. *Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering.* COLING-ACL '06 Student Research Workshop.

Thorsten Brants, 1997. The NEGRA Export Format. *CLAUS Report, Saarland University*.

Eric Brill, 1995. Unsupervised Learning if Disambiguation Rules for Part of Speech Tagging. *3rd Workshop on Very Large Corpora*.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souze, Jenifer C. Lai and Robert Mercer, 1992. Class-Based N-Gram Models of Natural Language. *Computational Linguistics*, 18:467-479.

Alexander Clark, 2003. Combining Distributional and Morphological Information for Part of Speech Induction. *EACL '03*.

Sajib Dasgupta and Vincent Ng, 2007. Unsupervised Part-of-Speech Acquisition for Resource-Scarce Languages. *EMNLP '07*.

Steven Finch, Nick Chater and Martin Redington, 1995. Acquiring syntactic information from distributional statistics. *Connectionist models of memory and language.* UCL Press, London.

Dayne Freitag, 2004. *Toward Unsupervised Whole-Corpus Tagging.* COLING '04.

Jianfeng Gao and Mark Johnson, 2008. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. *EMNLP '08*.

Yoav Goldberg, Meni Adler and Michael Elhadad, 2008. EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start). *ACL '08*

Sharon Goldwater, Tom Griffiths, and Mark Johnson, 2006. Interpolating between types and tokens by estimating power-law generators. *NIPS '06*.

Sharon Goldwater and Tom Griffiths, 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. *ACL '07*.

Joshua Goodman, 2001. A Bit of Progress in Language Modeling, Extended Version. *Microsoft Research Technical Report MSR-TR-2001-72*.

João Graça, Kuzman Ganchev, Ben Taskar and Frenando Pereira, 2009. Posterior vs. Parameter Sparsity in Latent Variable Models. *NIPS '09*.

Maurice Kendall and Jean Dickinson, 1990. Rank Correlation methods. Oxford University Press, New York.

Aria Haghighi and Dan Klein, 2006. Prototype-driven Learning for Sequence Labeling. *HLT-NAACL '06*.

Mark Johnson, 2007. Why Doesnt EM Find Good HMM POS-Taggers? *EMNLP-CoNLL '07*.

Harold W. Kuhn, 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83-97.

Mirella Lapata, 2006. Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 4:471-484.

Sven Martin, Jorg Liermann, and Hermann Ney, 1998. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24:19-37.

Marina Meila, 2007. Comparing Clustering - an Information Based Distance. *Journal of Multivariate Analysis*, 98:873-895.

Bernard Merialdo, 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155-172.

Michael Mitzenmacher , 2004. A Brief History of Generative Models for Power Law and Lognormal Distributions. *Internet Mathematics*, 1(2):226-251.

James Munkres, 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the SIAM*, 5(1):32-38.

Hermann Ney, Ute Essen, and Reinhard Kneser, 1994. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1-38.

Sujith Ravi and Kevin Knight, 2009. Minimized Models for Unsupervised Part-of-Speech Tagging. *ACL '09*.

Roi Reichart and Ari Rappoport, 2009. The NVI Clustering Evaluation Measure. *CoNLL '09*.

Kenneth Rose, Eitan Gurewitz, and Geoffrey C. Fox, 1990. Statistical Mechanics and Phase Transitions in Clustering. *Physical Review Letters*, 65(8):945-948.

Andrew Rosenberg and Julia Hirschberg, 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *EMNLP '07*.

Hinrich Schuetze, 1995. Distributional part-of-speech tagging. *EACL '95*.

Noah A. Smith and Jason Eisner, 2004. Annealing Techniques for Unsupervised Statistical Language Learning. *ACL '04*.

Noah A. Smith and Jason Eisner, 2005. Contrastive Estimation: Training Log-Linear Models on Unlabeled Data. *ACL '05*.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay, 2009. Adding More Languages Improves Unsupervised Multilingual Part-of-Speech Tagging: A Bayesian Non-Parametric Approach. *NAACL '09*.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay, 2008. Unsupervised Multilingual Learning for POS Tagging. *EMNLP '08*.

Jurgen Van Gael, Andreas Vlachos and Zoubin Ghahramani, 2009. *The Infinite HMM for Unsupervised POS Tagging.* EMNLP '09.

Qin Iris Wang and Dale Schuurmans, 2005. Improved Estimation for Unsupervised Part-of-Speech Tagging. *IEEE NLP-KE '05*.

Shaojun Wang, Dale Schuurmans and Yunxin Zhao, 2002. The Latent Maximum Entropy Principle. *ISIT '02*.

Qiuye Zhao and Mitch Marcus, 2009. *A Simple Unsupervised Learner for POS Disambiguation Rules Given Only a Minimal Lexicon.* EMNLP '09.