AAAI Technical Report FS-12-05
Information Retrieval and Knowledge Discovery in Biomedical Text

Global and Local Approach of Part-of-Speech Tagging for Large Corpora

Shi Yu
Institute for Genomics and Systems Biology
University of Chicago

Robert Grossman and **Andrey Rzhetsky**
Institute for Genomics and Systems Biology
Computation Institute
University of Chicago

Abstract

We present Global-Local POS tagging, a framework to train generative stochastic Part-of-Speech models on large corpora. Global Taggers offer several advantages over their counter parts trained on small, curated corpus, including the ability to automatically extend and update their models to new text. Global Taggers also avoid a fundamental limitation of current models, whose performance heavily relies on curated text with manually assigned labels. We illustrate our approach by training several Global Taggers, implemented with generative stochastic models, on two large corpora using high performance computing architecture. We further demonstrate that global taggers can be improved by incorporating models trained on curated text, called Local Taggers, for better tagging performance derived from specific topics.

Background

As digital libraries approach the totality of the global bibliome, automated mining of knowledge from large collections of free text becomes increasingly important. The design of probabilistic text-mining applications is typically a substantial collective effort; it often involves the joint work of experts in computational linguistics, computer science, and statistical modeling. When the target corpus is specialized, replete with jargon, with exotic named entities, and thousands of novel relations types, as the case in biomedicine, domain experts are generally required as well. A deep analysis of sentence structure and meaning extraction almost invariably requires part-of-speech tagging (POS tagging) of sentences. Once upon a time this task was performed exclusively by hand (Greene and Rubin 1971; Klein and Simmons 1963), but nowadays, it is mostly accomplished by algorithms which have undergone intense evolution over the past four decades.

POS taggers are currently widely used and readily available for practical applications. Earlier successful taggers, such as Brill's algorithm (Brill 1992), incorporated a manually crafted set of deterministic rules. More recently, the technology has evolved towards the use of stochastic taggers based on explicit statistical models. For example, some of more popular stochastic taggers currently utilize Hidden Markov Models (HMM) (Brants 2000), Maximum entropy Markov Models (MEMM) (Toutanova et al. 2003) and Conditional Random Field (CRF) Models (Lafferty 2001). According to Wikipedia, the best performance for English text is currently obtained by such probabilistic models. The state-of-the-art accuracy performance evaluated on the Treebank Wall Street Journal (Treebank WSJ) corpus is near 97.3% accuracy per token, which is hypothetically close to the upper limit of optimal performance achievable by taggers implemented using statistical modeling (Manning 2011).

In this study, we examine stochastic POS tagging and address the following problems. First, we compare *global* training performance of a set of stochastic POS tagging models by optimizing their parameters with respect to very large corpora. Second, we suggest some new variations of previously implemented generative models, and demonstrate the utility of high performance computing architectures for iterative training over very large datasets. Finally, we document an improvement of POS tagging performance for diverse corpora.

Most of the currently utilized stochastic POS taggers were trained on small hand-tagged corpora such as Brown (Kucera and Francis 1967), Treebank WSJ (Marcus, Marcinkiewicz, and Santorini 1993), and GENIA (Tsuruoka and Tsujii 2005). As a result, models trained on small manually curated corpora are often applied to analysis of unseen text that is several orders of magnitude larger than the training set.

For example, the MEDLINE database currently includes 26 million publication entries(as of April 2012), approximately 0.1 billion sentences, and 2.3 billion token occurrences (8.2 million unique words). Likewise, Wikipedia contains a collection of 60 million sentences and 8.7 million unique words. By comparing corpora of this size to those used to train existing POS taggers (shown in Table 1), it is obvious that those taggers were trained on comparably tiny pieces of text. Therefore, when such models are applied to large amounts of previously unseen text, their performance is very likely to degrade significantly because most of the words are novel. In fact, prior studies have demonstrated that the per token accuracy performance of state-of-the-art taggers degrades to 91.0% when applied to previously unseen words(Toutanova et al. 2003). To avoid this scenario when

Table 1: Commonly used combinations of POS taggers and the training corpora.

Name	Training Corpus	Word Occurrences	Samples	Corpus Year
Brill tagger [1]	Brown Corpus [2]	1,01 million	500 samples	1967
TnT tagger [3]	Treebank WSJ [4]	1.20 million	50,000 sentences	1993
Stanford Tagger 1.0 [5] SVM Tool [6]	Treebank 3 WSJ	1.04 million	43,746 sentences	1999
GENIA Tagger [7]	Treebank 3 WSJ	1.04 million	43,746 sentences	1999
	GENIA corpus [7]	0.52 million	18,534 sentences	2005
	PennBioIE corpus [8]	2,257 entities	N/A	2004
MedPost [9]	manually curated MEDLINE	0.16 million	5,700 sentences	2004

applying POS taggers to new text that is millions of times larger than their training sets, we were motivated to use as much text as possible for a POS tagger training. We called such a tagger constructed in this fashion a **Global Tagger**, in that it literally covers all words in the corpora of interest such that the performance drop on unknown words should be greatly reduced. The Global Tagger idea seemed very attractive because theoretical analysis and empirical evaluations have both shown that performance improves consistently upon training with larger text. Moreover, the Global Tagger should be able to assign tags to new words as its underlying model automatically and continuously adapts as new data arrives. If this procedure were to go on continuously, the Global Tagger should be able to learn POS information from an exhaustive set of language-specific sentences.

We present our approach to training Global POS taggers using corpora derived from Wikipedia and MEDLINE, which we define as our *training corpus*. The corpora were tagged automatically by two high quality Off-the-shelf POS taggers, denoted as *initial taggers*; then, the tagged training corpora were used to reconstruct a set of novel POS Taggers, implemented using generative models. This training procedure of Global Tagger does not require any curated text; its performance only relies on the quality of the initial taggers and the richness of training data. The approach is similar to the construction of unsupervised or semi-supervised POS taggers. The obtained Global Tagger should work well given that the initial tagger is adequate and the training corpus is large. Moreover, the generative model underlying the Global Tagger should enable it to adapt to the new text iteratively, i.e., the generative model becomes updated and enriched as parameters estimates for new words are obtained.

The performance of the Global Tagger can be further improved when curated text is available. In the literature, stochastic taggers generally outperform rule-based ones; moreover, discriminative models usually perform better than generative models [10]. The advantage of discriminative models, such as entropy based models (Berger, Pietra, and Pietra 1996) is their ability to incorporate a lot of features extracted from curated text. To achieve this, discriminative models require pre-tagged corpora of high quality (such tags are denoted as *true labels*). Unfortunately, in text mining, labeled data is always in short supply especially when the size of text is large, supporting the merit of our Global Tagger. If the curated text is available, we can combine generative and discriminative approaches and interpret these techniques within a single framework. An intuitive strategy is to implement a new tagger built upon the Global Tagger by incorporating more features generated from curated labels. We call this new method a **Local Tagger**, where the term “local” implies that the curated text is a specific subset of the entire text universe. It also characterizes the fact that curated labels and the extracted features are corpus-specific. For example, the labels and features extracted from curated MEDLINE text are specific to biomedical corpora, which could be disparate from those extracted from other text sources, such as the Wall Street Journal. The advantage of Local Taggers is that they can improve the performance of a Global Tagger in certain cases, especially when the text originates from a source similar to the one used to create the Local Tagger. There is also another advantage: information contributed by the Local Tagger can be fed back to the Global Tagger in a form of *active learning* that lets the true labels provided by the human curators correct the mistakes made by the Global Tagger.

We present our Local Tagging approach using curated Treebank WSJ and GENIA text, and we propose a **Global-Local** approach to constructing POS taggers using large corpora. The performance of the Global Taggers and Local Taggers was evaluated on the Treebank WSJ test set and the GENIA corpus. For Global Taggers, the simplest model achieved 96.46% per token accuracy on the WSJ test corpus and 98.25% on the GENIA corpus. The best performance on the WSJ corpus obtained by the set of Global Taggers was based on a third order bidirectional HMM (96.80%). The best performance on the GENIA corpus was obtained by a third order unidirectional HMM (98.37%). When using curated label information, the performance of Local Taggers increased significantly, the simplest model obtained 96.76%

[1] (Brill 1992)

[2] (Kucera and Francis 1967)

[3] (Brants 2000)

[4] (Marcus, Marcinkiewicz, and Santorini 1993)

[5] (Toutanova et al. 2003)

[6] (Gimnez and Mrquez 2004)

[7] (Tsuruoka and Tsujii 2005)

[8] (Mcdonald et al. 2004)

[9] (Smith, Rindflesch, and Wilbur 2004)

[10] http://aclweb.org/aclwiki

and the best performer obtained 96.94% on WSJ, which are both comparable to the state-of-the art performance given in the literature need reference and numbers for the state-of-the art. The idea of tagger adaptation has been formulated with regard to hand-curated domain-specific resources. For example, the Chamiak-Lease parser (Lease and Charniak 2005) retrains the originally WSJ-trained tagger on hand-annotated MEDLINE corpus. In contrast to using only small hand-labeled corpora, we propose in addition utilizing for the tagger re-training enormous unlabeled corpora to obtain global probability estimates that improves the POS tagging of unseen words over the off-the-shelf taggers. Our approach relies solely on tag predictions and probability estimates obtained from a very large corpus. To our knowledge, no previous approach has attempted to process corpora of this scale or has proposed similar ideas.

Experimental Setup

Two large *training corpora* were used in our experiments. We constructed a large Wiki corpus (roughly 7.76 GB in total size) from a snapshot of the all English-language articles in Wikipedia (as of October 2011). We downloaded Wikipedia articles, converted the resulting XML files to flat text using the WP2TXT software (http://wp2txt.rubyforge.org/), and substituted all the non-unicode characters with white spaces. We also retrieved a snapshot of the entire MEDLINE corpus (roughly 14 GB in total size) from NCBI (in April 2012). We applied two high-quality *initial taggers*, the Stanford Tagger (model: left3words-wsj-0-18.tagger) (Toutanova et al. 2003) and the GENIA Tagger (Tsuruoka et al. 2005), on both corpora. The obtained tagged corpora cannot be considered as curated, as they likely contain errors, however, we assumed that such tagged corpora likely contain enough reliable information to reconstruct high-quality *Global Taggers*.

We created the nine Global Taggers by implementing different generative models that assign hidden variables (POS tags) $\mathbf{Y}$ that maximize the likelihood $P(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta})$, where $\mathbf{X}$ is the observed corpus $\mathbf{X}$ and $\mathbf{\Theta}$ is vector of all model parameters. We assume independence among M sentences in corpus,

$$P(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta}) = \prod_{j=1}^{M} P(\mathbf{x}^j, \mathbf{y}^j|\mathbf{\Theta}), \qquad (1)$$

where $\mathbf{x}^j$ and $\mathbf{y}^j$ are respectively the sequence of tokens and tag sequence of the $j-$th sentence. The baseline model was simplest one assuming each observed token along with its context in a sentence being independently emitted by a latent tag variable, according to

$$\begin{aligned} P(\mathbf{x}^j, \mathbf{y}^j|\mathbf{\Theta}) &= P(\mathbf{x}^j|\mathbf{y}^j, \mathbf{\Theta})P(\mathbf{y}^j|\mathbf{\Theta}) & (2) \\ &= \prod_{i=1}^{N_j} P(\vec{x}_i^j|y_i^j, \mathbf{\Theta})P(y_i^j|\mathbf{\Theta}), & (3) \end{aligned}$$

where $\vec{x}_i^j$ is the set of token-level features of word x_i^j at the $i-$th word of sentence j, y_i^j is the $i-$th latent variable of sentence j. Within each sentence, we further define the emission probability and drop notations j and $\mathbf{\Theta}$ for simplicity,

$$\begin{aligned} P(\vec{x}_i|y_i) = P(\underline{x_i}|y_i)P(\underline{x_{i-1}x_i}|y_i)P(\underline{x_ix_{i+1}}|y_i) \times \\ P(\underline{x_{i-1}x_ix_{i+1}}|y_i), \qquad (4) \end{aligned}$$

as independent combination of emission probabilities of a single token $\underline{x_i}$, two token bigrams $\underline{x_{i-1}x_i}$ and $\underline{x_ix_{i+1}}$, and one token trigram $\underline{x_{i-1}x_ix_{i+1}}$. Statistics reflecting sizes of the corpora are given in Table 2.

Table 2: The sizes of training corpora and token-level features in our experiments.

	Wiki	**MEDLINE**
Corpus size	7.76 G	14 G
No. of sentences	59,695,940	100,290,897
$\underline{x_i}$	8,722,021	8,224,949
$\underline{x_{i-1}x_i}$	105,010,091	100,751,602
$\underline{x_ix_{i+1}}$	100,006,646	94,377,260
$\underline{x_{i-1}x_ix_{i+1}}$	379,376,196	462,750,572

The other eight models are variants of the Hidden Markov formalism, defining $P(\mathbf{y}^j|\mathbf{\Theta})$ in (2) by high order transition probabilities and various inference directions. We compared first order, second order and third order transitions (orders higher than three were not considered for the sake of efficiency). We compared the inference directions such as left-to-right, right-to-left, and bidirectional. All models used the same set of token level features as the baseline models adopted in (4).

We also investigated the effects of the *training corpus* and *initial tagger* on the performance of the Global Taggers. With two training corpora and two initial taggers, we benchmarked four configurations for each Global Tagger. To evaluate performance, we utilized two curated corpora, Treebank WSJ and GENIA. In both global and local tagger evaluations, we used Treebank WSJ 22-24 as the test set, which is the standard setting adopted by many POS taggers, allowing performance to be systematically compared. However, the GENIA abstracts might be derived from part of the MEDLINE abstracts, but we did not explicitly address this issue. For global tagger training, neither curated corpus was used. For local taggers, WSJ 0-21 and a random sample of 70% of GENIA corpus was used in training and the performance is validated on WSJ 22-24 and the remaining 30% of GENIA corpus.

Computational Infrastructure

The computational infrastructure used in our experiments was the Open Cloud Consortium's OCC-Y Cluster, which consists of 60 computational nodes (each node has 8 cores and 32 G memory, connected by gigabit ethernet). The main pipeline of text mining was designed and implemented in Java Apache Hadoop 0.20.203 (http://hadoop.apache.org/). The Wiki, MEDLINE, and the parametric data required for model learning were stored within the Hadoop distributed file system with 1.02 PB configured capacity. The algorithms for corpus tagging, parametric estimation, and model

Table 3: Performance of the Global Taggers. We compared four different global taggers and validated their performance on two annotated corpora: 1. A global tagger constructed by Wiki corpus and Stanford tagger validated on WSJ ([WS]W); 2. MEDLINE corpus and GENIA based tagger validated on GENIA ([MG]G); 3. MEDLINE and Stanford tagger validated on GENIA corpus ([MS]G); 4. Wiki and GENIA validated on WSJ ([WG]W). Their performance is shown in Column 3 to Column 6. The performance of [WG]W on WSJ validation is higher than [WS]W probably because that the Off-the-shelf GENIA tagger was also trained on the WSJ test data.

Model	**Transition Probabilities** $P(\mathbf{y})$	**[WS]W**	**[MG]G**	**[MS]G**	**[WG]W**
Baseline	$\prod_{i=1}^{N} P(y_i)$	0.9646	0.9825	0.9352	0.9672
HMM L 1	$P(y_1) \prod_{i=2}^{N} P(y_i \vert y_{i-1})$	0.9665	0.9829	0.9356	0.9698
HMM R 1	$P(y_N) \prod_{i=1}^{N-1} P(y_i \vert y_{i+1})$	0.9666	0.9831	0.9358	0.9699
HMM L 2	$P(y_1)P(y_2 \vert y_1) \prod_{i=3}^{N} P(y_i \vert y_{i-1}, y_{i-2})$	0.9671	0.9835	0.9362	0.9710
HMM R 2	$P(y_N)P(y_{N-1} \vert y_N) \prod_{i=1}^{N-2} P(y_i \vert y_{i+1}, y_{i+2})$	0.9654	0.9823	0.9359	0.9672
HMM LR 1	$P(y_1)P(y_N) \prod_{i=2}^{N-1} P(y_i \vert y_{i-1}, y_{i+1})$	0.9676	0.9832	0.9360	0.9703
HMM L2 R1	$P(y_1)P(y_2 \vert y_1)P(y_N) \prod_{i=3}^{N-1} P(y_i \vert y_{i-2}, y_{i-1}, y_{i+1})$	**0.9680**	0.9833	0.9366	0.9703
HMM L1 R2	$P(y_1)P(y_N)P(y_{N-1} \vert y_N) \prod_{i=2}^{N-2} P(y_i \vert y_{i-1}, y_{i+1}, y_{i+2})$	0.9676	0.9830	0.9364	0.9703
HMM L3	$P(y_1 P(y_2 \vert y_1)P(y_3 \vert y_1, y_2) \prod_{i=4}^{N} P(y_i \vert y_{i-3}, y_{i-2}, y_{i-1}))$	0.9676	**0.9837**	0.9363	0.9711

training were implemented in the Map/Reduce framework (Dean and Ghemawat 2008), where data was shuffled, sorted and aggregated as text based key-value pairs. The Stanford Tagger was implemented in Java so we embedded its APIs directly in our pipeline. The GENIA tagger was originally written in C++, we parallelized it on large corpora using Hadoop Streaming. In our platform, tagging the entire Wiki corpus took approximately about 30 minutes for GENIA tagger and 20 minutes for Stanford left3words tagger. Tagging the entire MEDLINE corpus took 45 minutes and 30 minutes for the GENIA Tagger and the Stanford Tagger, respectively. When training the models over the entire Wiki and MEDLINE corpora, the space required to store all the required parameters was quite large. To speed up the computation, when the trained models were applied to a specific set of new data, we extracted all of the feature occurrences from the new text, looked up the parametric data corresponding to the extracted features, and then stored the results either in memory or in a database.

Performance of Global Taggers

As shown in Table 3, the performance of the generative models reconstructed on large text was very good. The simple baseline model obtained a performance of 96.46%, which was probably better than some complicated state-of-the-art models trained on small datasets. The good performance is likely attributed to the fact that large text transforms a lot of unknown words to known words. In automatically tagged training corpora, the predictions of unknown words may be mistaken in some contexts, yet if these words are frequently observed and correctly tagged in other contexts, their POS tags may still be learned relatively accurately given the generous amounts of data. In other words, imagine that if we have unlimited manpower such that we can curate the true labels for large training corpora. Given this scenario, a global tagger based on a simple model should have very good performance (possibly even better than 96.46%).

Additionally, we increased the complexity of the Global Tagger by adding more transition probabilities to its underlying model (from the second row in Table 3). We compared the order of transitions within the hidden markov chain (HMM), and we also compared the directionality of inference. We name each model according to its order and inference direction, for example, HMM 1L denotes a first order HMM with transitions from left-to-right, and HMM 1R implies a first order model with directionality from right-to-left. Each column in Table 3 displays tagger performance after training according to a single configuration. Because the corpora are fixed in both validations (WSJ 22-24 and GENIA) so there is no variation shown in Table 3. We also tried 10 random repetitions using 50% of the Wiki and MEDLINE corpora in tagger constructions and the variances of obtained performance were respectively smaller than 0.0003 and 0.0002; thus, the improvements were significant over baseline models. Generally, higher order transitions appeared to steadily improve the performance on all evaluations. The direction of inference also matters: left-to-right inference generally outperforms right-to-left one. We also investigated three HMM models based on bidirectional inference (HMM LR1, HMM L2R1, and HMM L1R2). The training of these bidirectional, HMM models can be achieved by a mean field approximation (Forbes and Peyrard 2003).

We cross-compared performance of Global Taggers using four different configurations presented in Table 3. On one hand, Global Taggers trained with Wiki corpus performed better than those trained with MEDLINE in WSJ evaluation. On the other hand, those trained with MEDLINE performed better than Wiki in GENIA validation. Using Stanford Tagger as the initial tagger, the global tagger trained on MEDLINE corpus performed significantly better in GENIA validation than the one trained on Wiki. The Off-the-shelf Stanford L3W tagger performance on GENIA corpus was 90.53% whereas our [MS]G models all performed higher than 93%. This indicates that when applying a non-biomedical tagger to a large biomedical corpus, the global probability estimates could improve the POS tagging of biomedical terms over an off-the-shelf tagger.

Performance of Local Taggers

Our experiments showed that the information contained within the curated data can be transformed to a new set of discriminative features, as used in MEMM discriminative models (Toutanova et al. 2003), by conditioning on the tokens and tags simultaneously. Combing these features with the token-level features extracted from a global tagger, the performance of a local tagger can be further improved. We obtained new discriminative features for local taggers by implementing the two steps as illustrated in Figure 1: First, we tagged the curated data using global taggers and obtained intermediate tags y_i. Next, we modeled the tokens x_i and the intermediate tags y_i as observations and inferred the set of hidden variables z_i associated with the local taggers. The statistical models of Local Taggers are exactly the same as the Global Taggers, where the only difference is that the Local Taggers incorporate more features obtained from the curated text.

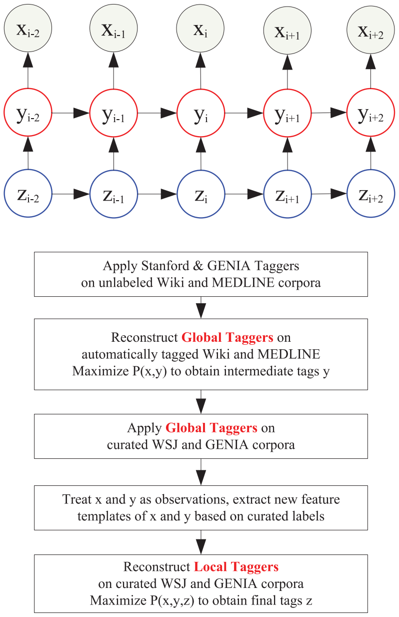
x_{i-2}
x_{i-1}
x_i
x_{i+1}
x_{i+2}
y_{i-2}
y_{i-1}
y_i
y_{i+1}
y_{i+2}
z_{i-2}
z_{i-1}
z_i
z_{i+1}
z_{i+2}
Apply Stanford & GENIA Taggers on unlabeled Wiki and MEDLINE corpora
Reconstruct Global Taggers on automatically tagged Wiki and MEDLINE Maximize P(x,y) to obtain intermediate tags y
Apply Global Taggers on curated WSJ and GENIA corpora
Treat x and y as observations, extract new feature templates of x and y based on curated labels
Reconstruct Local Taggers on curated WSJ and GENIA corpora Maximize P(x,y,z) to obtain final tags z

Figure 1: The Global-Local approach of POS tagging.

A Local Tagging model maximizes the likelihood

$$P(\mathbf{X},\mathbf{Y},\mathbf{Z}|\mathbf{\Theta}) = \prod_{j=1}^{M} P(\mathbf{x}^j,\mathbf{y}^j|\mathbf{z}^j,\mathbf{\Theta})P(\mathbf{z}^j|\mathbf{\Theta}) \quad (5)$$

regarding $\mathbf{x}$, $\mathbf{y}$ as observations and $\mathbf{z}$ as hidden variables. We assumed that the features of the global tagger and the local tagger are independent, thus the emission probability now becomes

$$P(\mathbf{x}^j,\mathbf{y}^j|\mathbf{z}^j,\mathbf{\Theta}) = \prod_{i=1}^{N} P(< x_i^j, y_i^j > |\mathbf{z}^j,\mathbf{\Theta})P(\mathbf{z}^j|\mathbf{\Theta}) \quad (6)$$

where $P(< x_i^j, y_i^j > |\mathbf{z}^j,\mathbf{\Theta})$ is the emission probability generating a set of observations represented by x_i^j and y_i^j. In our approach, we define (we omit j and $\mathbf{\Theta}$ for simplicity)

$$P(< x_i, y_i > |\mathbf{z}) = \prod_{a=1}^{4} f_a(\vec{x}_i|z_i) \prod_{b=1}^{12} f_b(\overrightarrow{xy}_i|z_i) \quad (7)$$

where f_a are the conditional probabilities estimated using the features listed in Table 1 defined in (4), f_b are the conditional probabilities using features listed in Table 4. To obtain the estimates of f_b, we counted the frequencies of these features in curated text, then normalized them by the frequencies of the true labels. Thus, we obtained new features which were used to estimate the joint likelihood of the generative models underlying the Local Tagger, where tokens and intermediate tags are all treated as observations.

Table 4: Feature templates using tokens and intermediate tags in Local Taggers.

No.	Template	No.	Template
$f_b(1)$	$\langle x_i, y_i \rangle$	$f_b(7)$	$\langle x_{i-1}, y_i \rangle$
$f_b(2)$	$\langle y_{i-1}, y_i \rangle$	$f_b(8)$	$\langle x_{i+1}, y_i \rangle$
$f_b(3)$	$\langle y_{i-2}, y_{i-1}, y_i \rangle$	$f_b(9)$	$\langle x_i, y_{i-1}, y_i \rangle$
$f_b(4)$	$\langle y_{i-1}, y_i, y_{i+1} \rangle$	$f_b(10)$	$\langle x_i, y_i, y_{i+1} \rangle$
$f_b(5)$	$\langle y_i, y_{i+1} \rangle$	$f_b(11)$	$\langle x_{i-1}, x_i, y_i \rangle$
$f_b(6)$	$\langle y_i, y_{i+1}, y_{i+2} \rangle$	$f_b(12)$	$\langle x_i, x_{i+1}, y_i \rangle$

The performance of the Local Taggers was significantly improved by human curation of POS labels. The baseline model achieved a 96.76% per token accuracy on the WSJ corpus with the best performer reaching 96.94%, which was implemented with several third-order HMM models. We also validated the Off-the-shelf MEMM taggers using the same training and test sets. The MEMM models were trained for 100 iterations using the Stanford POS tagger toolbox. The performance of our Local Taggers was close to the MEMM left-3-word (L3W) model, but slightly lower than the bidirectional model, probably because our initial taggers were based on L3W models. In GENIA validation, our Local Taggers outperformed both L3W and Bidi Off-the-Shelf models, which were trained on a combined corpus of the entire WSJ and a random 70% sample of GENIA (with 10 repetitions). The obtained performance without splitting training and test data independently was exceptionally high (99.10% with the Off-the-Shelf GENIA tagger and 99.89% with our Local Taggers). In our further validation using new annotated corpora apart from WSJ and GENIA, the Global and Local Taggers all significantly outperformed the Off-the-Shelf taggers.

Conclusion

We present Global and Local POS tagging, a framework to train generative stochastic Part-of-Speech models on large corpora. This new framework is not a competitor to any existing approach. Instead, it is a generic approach that should be able to enhance a diverse set of existing taggers. The Global Tagger, powered by high performance computing, enables us to leverage very large corpora. The Local Tagger,

Table 5: Performance of Local Taggers and Comparison to Off-the-shelf taggers

Model	[WS]W	[MG](G30%)
Baseline	0.9676	0.9886 ± 0.0002
HMM L 1	0.9685	0.9887 ± 0.0002
HMM R 1	0.9685	0.9888 ± 0.0001
HMM L 2	0.9689	0.9887 ± 0.0002
HMM R 2	0.9678	0.9886 ± 0.0002
HMM LR 1	0.9691	0.9889 ± 0.0002
HMM L2 R1	0.9691	**0.9890** ± 0.0002
HMM L1 R2	**0.9694**	0.9889 ± 0.0002
HMM L3	0.9693	**0.9890** ± 0.0002
MEMM L3W	0.9691	0.9847 ± 0.0002
MEMM Bidi	0.9718	0.9869 ± 0.0003

refined by curated tags of high quality, improves the performance in specific domains. The interplay of the Global and Local taggers provides us a rich framework for improving the accuracy of POS tagging problems in large corpora.

The proposed framework also leaves plenty of room for further improvement. The strong assumption of independence among features can be relaxed so that the weights of features can be optimized iteratively, as it is done in models based on CRFs and MEMMs. The effect of label information contained within the curated text is not restricted to the Local Tagger. If we find that the estimates of the token level features are inconsistent with the estimates provided by the curated text, we could tailor and update the Global Tagger model as well. This correction can also be accomplished iteratively, and can even involve human curation such that the framework of Global-Local tagging embeds an active learning paradigm. Moreover, the proposed framework is ready for consensus learning and data fusion in a wide range of data mining applications.

Acknowledgement

The work was supported by National Institutes of Health (Grants NIGMS/NIH P50GM081892-03A1 and P50MH094267) and by the National Science Foundation (Grant OISE - 1129076). This work also used the Open Cloud Consortium's OCC-Y Cluster, which was donated by Yahoo! Inc.

References

Berger, A. L.; Pietra, S. A. D.; and Pietra, V. J. D. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22:39–71.

Brants, T. 2000. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*.

Brill, E. 1992. A simple rule-based part of speech tagger. In *In Proceedings of the ANLC 92*, 152–155. Association for Computational Linguistics.

Dean, J., and Ghemawat, S. 2008. Mapreduce: simplified data processing on large clusters. *Commun. ACM* 51(1):107–113.

Forbes, F., and Peyrard, N. 2003. Hidden markov random field model selection criteria based on mean field-like approximations. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(9):1089–1101.

Gimnez, J., and Mrquez, L. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 43–46.

Greene, B. B., and Rubin, G. M. 1971. Automated grammatical tagging of english. Technical Report NA-92-08.

Klein, S., and Simmons, R. 1963. A computational approach to grammatical coding of english words. *JACM* 10:334–337.

Kucera, H., and Francis, W. N. 1967. *Computational Analysis of Present-Day American English*. Brown University Press.

Lafferty, J. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *In Proceeding of ICML01*, 282–289. Morgan Kaufmann.

Lease, M., and Charniak, E. 2005. Parsing biomedical literature. In *Proceedings of the the Second International Joint Conference on Natural Language Processing*, IJCNLP05, 58–69.

Manning, C. D. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *In Proceedings of CICLing 2011*, 171–189. Lecture Notes in Computer Science, Springer.

Marcus, M. P.; Marcinkiewicz, M. A.; and Santorini, B. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.* 19(2):313–330.

Mcdonald, R. T.; Winters, R. S.; Mandel, M.; Jin, Y.; White, P. S.; and Pereira, F. 2004. An entity tagger for recognizing acquired genomic variations in cancer literature. *Bioinformatics* 20(17):3249–3251.

Smith, L.; Rindflesch, T.; and Wilbur, W. J. 2004. Medpost: a part-of-speech tagger for biomedical text. *Bioinformatics* 20(14):2320–2321.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the NAACL03*, NAACL '03, 173–180. Stroudsburg, PA, USA: Association for Computational Linguistics.

Tsuruoka, Y., and Tsujii, J. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, 467–474. Stroudsburg, PA, USA: Association for Computational Linguistics.