Model Selection for Part of Speech Tagging With Type Supervision

Abstract

Model selection (picking, for example, a parametric model family, a prior, and an estimation criterion) is crucial for building high-accuracy classifiers. In supervised learning settings, the accuracy of a model can be estimated on a labeled set and used to guide modeling decisions. In unsupervised or type-supervised learning settings, unsupervised model selection criteria are used, but their performance is far from optimal (Smith and Eisner, 2005). Here we propose a new model selection criterion for type-supervised sequence labeling settings, which uses the available weak supervision (type-level constraints) more directly to come up with an accuracy-like metric. We evaluate the effectiveness of the method on type-supervised POS-tagging in nine languages, using both HMM and LDA-based models, and show that it outperforms two unsupervised model selection criteria.

1 Introduction

Fully supervised training of part-of-speech (POS) taggers works well when plenty of labeled examples are available (Manning, 2011). However, corpora labeled with POS tags at the token level are not available for the vast majority of languages.\footnote{We were able to find POS-tagged corpora for only 37 languages, about 1% of the languages listed in the online World Atlas of Language Structures (Dryer and Haspelmath, 2013).} A number of strategies have been used, and sometimes combined, to develop POS-taggers for low-resource languages:

- Unsupervised POS induction (Merialdo, 1994),
- Type-level supervision, also known as a tag dictionary and a POS lexicon (Smith and Eisner, 2005),
- Token-level supervision with few labeled examples (Søgaard, 2011), and
- Cross-lingual supervision by projecting annotations from a high-resource language (Snyder et al., 2008).

Recently, Garrette and Baldridge (2013) showed that, given a limited budget of annotation hours, type-level supervision (e.g., Fig. 1) is more cost-effective than token-level supervision for developing POS taggers for low-resource languages. Li et al. (2012) were the first to use Wiktionary\footnote{As of May 21, 2014, the Wiktionary project covers 158 languages, 103 of which have a sizable dictionary of more than a thousand word-tag entries. Data dumps can be found at http://toolserver.org/~enwikt/definitions/enwikt-defs-20120831-all.tsv.gz} -induced tag dictionaries to train competent POS taggers. We emphasize that this is a realistic setup, especially when it is hard to collaborate with a linguist familiar with the language, or in case of rapid development of NLP tools for disaster response purposes such as (Lewis, 2010; Neubig et al., 2011).
In this paper, we draw on the success of type-level supervision and address an important practical concern: the lack of a robust criterion for model selection in the absence of token-level supervision. We propose using an intuitive criterion for model selection based on a model’s ability to make predictions compatible with unseen entries in the tag dictionary (i.e., entries not used for model training). We experiment with two sequence labeling models in nine languages and show that the proposed criterion outperforms the standard model selection criteria which are based on data log-likelihood.

2 Background: Model Selection in NLP

Model selection is an essential step in the development of NLP predictors. Training statistical NLP models involves searching for optimal values of model parameters which minimize some loss function on a training set. However, before optimizing model parameters, several configurations and modeling decisions need to be made. Examples include the variables modeled (e.g., discriminative vs. generative models), the dependency structure between variables (see Fig. 2), feature templates and regularization strengths. Other decisions which may affect model performance include the algorithms used for decoding (e.g., Viterbi vs. greedy), the optimization procedure (e.g., direct gradient vs. expectation maximization), initialization of model parameters, and the convergence criterion for training. For the rest of this paper, we will use the term hyperparameters to refer to such decisions and configurations.

2.1 Notation

Given a set of hyperparameter settings \( \{ h_1, \ldots, h_M \} \), model selection finds the optimal configuration \( h_{\text{opt}} \) that maximizes an evaluation function, \( \text{eval} \), for a development set \( T_{\text{dev}} \): 
\[
\hat{m} = \arg \max_{m \in \{ 1, \ldots, M \}} \text{eval}(m, T_{\text{dev}})
\]

For supervised POS tagging problems, a common and robust choice of \( \text{eval} \) is token-level prediction accuracy:
\[
\text{eval}_{\text{sup}}(m, T_{\text{dev}}) \propto \sum_{x \in T_{\text{dev}}} \sum_{y \in \text{lex}[x]} \mathbb{1}(y[i] = y_{\text{gold}}[i])
\]

where \((x, y_{\text{gold}})\) consists of an input token sequence and the correct output POS tag sequence.

![Figure 2: The dependency structure among model variables for the two model families we consider: HMM (right) and LDA-based (left).](image)

The function \( \text{predict}_h \) is the decoding algorithm used. In the following section, we discuss choices of \( \text{eval} \) for POS tagging with type supervision.

3 Model Selection Criteria for POS Tagging With Type Supervision

Unfortunately, we cannot use \( \text{eval}_{\text{sup}} \) in the absence of token-supervision, since \( T_{\text{dev}} \) now consists of input token sequences only. Two possible choices of \( \text{eval} \) in POS tagging with type supervision are: a) conditional log-likelihood of “lex-compatible” labels given token sequences, and b) joint log-likelihood of lex-compatible labels and token sequences:
\[
\text{eval}_{\text{joint}}(m, T_{\text{dev}}) = \sum_{x \in T_{\text{dev}}} \sum_{y \in \text{lex}[x]} p_{\theta_m}(x, y)
\]
\[
\text{eval}_{\text{cond}}(m, T_{\text{dev}}) = \sum_{x \in T_{\text{dev}}} \sum_{y \in \text{lex}[x]} p_{\theta_m}(y \mid x)
\]

where \( (\text{lex}) \) is a tag dictionary and \( \text{lex}[x] \) is the set of label sequences compatible with token sequence \( x \), according to the tag dictionary. \( \text{eval}_{\text{joint}} \) evaluates models based on how well they explain input examples \( x \in T_{\text{dev}} \) and the corresponding entries in lex, which is only possible to compute in generative models such as HMMs. On the other hand, \( \text{eval}_{\text{cond}} \) evaluates models based on how well they explain entries of the tag dictionary only conditional on input token sequences. Therefore, \( \text{eval}_{\text{cond}} \) can be used with both generative and discriminative models.

A disadvantage of the joint likelihood criterion is that it cannot be used to compare models with very different independence assumptions or to compare a non-deficient (like HMM) and a deficient (like the LDA) model. Since a deficient model loses mass on impossible token sequences, its joint likelihood is expected to be lower. Also, Smith and Eisner (2005) showed that the joint log-likelihood criterion does not correlate well with accuracy, which motivated us to look for a criterion more directly related to tagging accuracy.
3.1 The Held-Out Lexicon Criterion

We propose an intuitive criterion for model selection based on a model’s ability to make predictions compatible with unseen entries in the tag dictionary (i.e., entries not used for model training). First, we split the POS lexicon into a train lexicon \( \text{lex}_{\text{train}} \) and a held-out lexicon \( \text{lex}_{\text{dev}} \) (see Fig. 1). While training the models (i.e., optimizing model parameters), we only use the train lexicon \( \text{lex}_{\text{train}} \). For model selection, we only use model predictions and the held-out lexicon \( \text{lex}_{\text{dev}} \). For POS tagging, we define \( \text{eval} \) as follows:

\[
\text{eval}_{\text{devlex}}(m, T_{\text{dev}}) \propto \sum_{x \in T_{\text{dev}}} \sum_{y_m = \text{predict}_{\text{h_0}}(x)} \frac{\mathbb{I}(y_m[i] \in \text{lex}_{\text{dev}}[x[i]])}{|\text{lex}_{\text{dev}}[x[i]]|}
\]

The held-out lexicon criterion scores a label prediction for a token \( x \) iff \( x \) appears in \( \text{lex}_{\text{dev}} \). If the predicted label is \( \text{lex}_{\text{dev}} \)-compatible, the score is inverse proportional to the number of \( \text{lex}_{\text{dev}} \)-compatible labels; otherwise the score is zero.

4 Experiments

We experiment with a total of \( M = 54 \) unique configurations of the hyperparameters. The configurations vary in three dimensions: model family (2 values), strength of \( L_2 \) regularization (9 values), and the minimum word frequency for word-tag features (3 values).

Fig. 2 shows the dependency structure of variables in the two model families we consider. Both models are feature-rich locally normalized models (Berg-Kirkpatrick et al., 2010) with a gaussian prior on feature weights. The first-order HMM model uses the feature set described in (Li et al., 2012): transition features, word-tag features \( \langle y_{i-1}, x_i \rangle \) (lowercased words with frequency greater than a threshold), whether the word contains a hyphen, whether the word starts with a capital letter, 2-char and 3-char suffixes, and whether the word contains a digit. The other model (LDA-based) is a feature-rich extension of the baseline PLSA model in (Toutanova and Johnson, 2007), and does not use their ambiguity-set model. Instead of using a feature-rich ambiguity set model to predict a set of possible tags for a word type, our implementation uses a feature-rich distribution over tags conditional on a word token as illustrated in Fig. 2. The LDA-based model uses the same feature set used for HMM, but also uses “displaced emission” features \( \langle y_i, x_{i-1} \rangle \) and \( \langle y_i, x_{i+1} \rangle \). Note that the LDA-based model is deficient; a token \( x_i \) may be generated once conditional on the previous label \( y_{i-1} \) and again conditional on the next label \( y_i \).

Data We use the Danish, Dutch, German, Greek, English, Italian, Portuguese, Spanish and Swedish datasets from CoNLL-X and CoNLL-2007 shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007). All hyperparameter configurations use the plain text of the training section of the datasets (\( T_{\text{plain}} \)) for model training. We map the POS labels in the CoNLL datasets to the universal POS tagset (Petrov et al., 2012) and exclusively use them for evaluation. We use the tag dictionaries provided by Li et al. (2012).

Setup For \( \text{eval}_{\text{cond}} \) and \( \text{eval}_{\text{joint}} \), we use the first 300 sentences in \( T_{\text{plain}} \) as \( T_{\text{dev}} \), and use the remaining sentences for \( T_{\text{train}} \). Then, for each configuration of hyperparameters (\( \theta_m \)), we train the model using \( T_{\text{train}} \) obtaining \( \theta_m \). We then select the winner configuration which maximizes \( \text{eval}_{\text{cond}} \) or \( \text{eval}_{\text{joint}} \), as functions of the held-out dataset \( T_{\text{dev}} \) and the full lexicon \( \text{lex} \). Finally, using the winner configuration \( \theta_{\text{win}} \), we retrain the model using the full dataset \( T_{\text{plain}} \) and report the token-level accuracy of its label predictions for the full dataset (i.e., transductive setting).

For \( \text{eval}_{\text{devlex}} \), we split the tag dictionary \( \text{lex} \) into a train lexicon \( \text{lex}_{\text{train}} \) and a held-out lexicon \( \text{lex}_{\text{dev}} \). We construct the held-out lexicon by subsampling tag dictionary entries such that the token-level coverage of the held-out lexicon is 25%, excluding the most frequent 500 words. Then, for each configuration of hyperparameters (\( \theta_m \)), we train the model using \( T_{\text{plain}} \) obtain-
Figure 4: Token-level accuracy, $eval_{cond}$, $eval_{joint}$, and $eval_{devlex}$ as a function of $L_2$ regularization strength for Italian (top) and Portuguese (bottom) HMM models.

We then select the winner configuration which maximizes $eval_{devlex}$, as functions of the full dataset $\mathcal{T}_{plain}$ and the held-out lexicon $lex_{dev}$. Finally, using the winner configuration $h_{\hat{m}}$, we retrain the model using the full lexicon $lex$ and report the token-level accuracy of its label predictions for the full dataset.

**Results and Discussion** A summary of model selection results for nine languages is shown in Fig. 3. For each language, we use the model selection criteria to select among 54 configurations of hyperparameters and report its token-level accuracy. The best accuracy any model selection criterion could achieve on a dataset is reported as “oracle”. For each model selection criterion, we also compute the average accuracy across languages.

On average, our held-out lexicon criterion ranks first (with average accuracy of 85.5%), closely followed by joint log-likelihood (with average accuracy of 85.2%). Conditional log-likelihood ranks last (with an average accuracy of 83.7%).

In the Danish, Dutch, Italian and Swedish datasets, using the held-out lexicon criterion improves the accuracy of the selected model with an absolute difference of 1.6% accuracy points or more. However, in the German and Greek datasets, using the held-out lexicon criterion substantially degrades accuracy of the selected model, compared to the joint log-likelihood criterion. Accuracy of the model selected with the held-out lexicon criterion vs. the joint log-likelihood criterion are comparable in the remaining datasets: English, Spanish and Portuguese.

In order to develop a better understanding of the results, we fix all dimensions of the hyperparameters except for one: $L_2$ regularization strength. Regularization strength is an important hyperparameter which often has a substantial effect on model performance. Fig. 4 shows how token-level accuracy and the various definitions of $eval$ function vary with regularization strength. The top graph in Fig. 4 shows these correlations for an Italian HMM model. Here, the held-out lexicon criterion $eval_{devlex}$ achieves an accuracy of 87.8%, which is better than the accuracy achieved by $eval_{cond}$ and $eval_{joint}$ (85.9% and 86.1%, respectively), but slightly worse than the oracle accuracy 88.2%. In the bottom graph, which uses a Portuguese HMM model, the held-out lexicon criterion $eval_{devlex}$ achieves an accuracy of 87.7%, while $eval_{joint}$ reaches the oracle accuracy 88.3%. Note the consistent poor performance of the conditional log-likelihood criterion.

We found that the conditional and joint log-likelihood criteria always choose HMM models as the winners. This is likely due to the deficiency of the LDA-based model but might also indicate the inability of these criteria to compare models with very different structures. Although the HMM models are better on average, the LDA-based models are better in four out of nine languages. Our held-out lexicon criterion was able to select, on average, a better model from the pool of HMM and LDA-based models, as compared to selecting from HMM models only. This suggests that the held-out lexicon criterion for model selection may be a more robust criterion when selecting among different families of models.

**Conclusion** We proposed a simple and effective model selection criterion which can be used in type-supervised learning settings. We performed a large comparative study of the new criterion and two unsupervised log-likelihood based criteria used in prior work for type-supervised POS Tagging. We showed that it outperformed the log-likelihood based criteria on average, and presented a focused analysis on the common problem of tuning regularization strength. An advantage of the new criterion is its ability to compare models with widely different structures such as HMM and LDA-based models.

---

3The value of $eval_{cond}$, $eval_{joint}$ and $eval_{devlex}$ are linearly scaled for better visualization.

4The average accuracy increases from 85.2% to 85.5% when we consider both model families instead of HMM models only.
References


