Model Selection for Type-Supervised Learning with Application to POS Tagging

Kristina Toutanova
Microsoft Research
Redmond, WA, USA

Waleed Ammar*
School of Computer Science
Carnegie Mellon University

Pallavi Chourdhury
Microsoft Research
Redmond, WA, USA

Hoifung Poon

Abstract

Model selection (picking, for example, the feature set and the regularization strength) is crucial for building high-accuracy NLP models. In supervised learning, we can estimate the accuracy of a model on a subset of the labeled data and choose the model with the highest accuracy. In contrast, here we focus on type-supervised learning, which uses constraints over the possible labels for word types for supervision, and labeled data is either not available or very small. For the setting where no labeled data is available, we perform a comparative study of previously proposed and one novel model selection criterion on type-supervised POS-tagging in nine languages. For the setting where a small labeled set is available, we show that the set should be used for semi-supervised learning rather than for model selection only – using it for model selection reduces the error by less than 5%, whereas using it for semi-supervised learning reduces the error by 44%.

1 Introduction

Fully supervised training of NLP models (e.g., part-of-speech taggers, named entity recognizers, relation extractors) works well when plenty of labeled examples are available. However, manually labeled corpora are expensive to construct in many languages and domains, whereas an alternative, if weaker, supervision is often readily available. For example, corpora labeled with POS tags at the token level are only available for around 35 languages, while tag dictionaries of the form displayed in Fig. 1 are available for many more languages, either in commercial dictionaries or community created resources such as Wiktionary. Tag dictionaries provide type-level supervision for word types in the lexicon. Similarly, while sentences labeled with named entities are scarce, gazetteers and databases are more readily available (Bollacker et al., 2008).

There has been substantial research on how best to build models using such type-level supervision, for POS tagging, super sense tagging, NER, and relation extraction (Craven et al., 1999; Smith and Eisner, 2005; Carlson et al., 2009; Mintz et al., 2009; Johannsen et al., 2014), *inter alia*, focussing on parametric forms and loss functions for model training. However, there has been little research on the practically important aspect of model selection for type-supervised learning. While some previous work used criteria based on the type-level supervision only (Smith and Eisner, 2005; Goldwater and Griffiths, 2007), much prior work used a labeled set for model selection (Vaswani et al., 2010; Soderland and Weld, 2014). We are not aware of any prior work aiming to compare or improve existing type-supervised model selection criteria.

For POS tagging, there is also work on using both type-level supervision from lexicons, and projection from another language (Täckström et al., 2013). Methods for training with a small labeled set have also been developed (Søgaard, 2011; Garrette and Baldridge, 2013; Duong et al., 2014), but there have not been studies on the utility of a small labeled set for model selection versus model training. Our contributions are: 1) a simple and generally applicable model selection criterion for type-supervised learning, 2) the first multi-lingual systematic evaluation of model selection criteria for type-supervised models, 3) empirical evidence that, if a small labeled set is available, the set should be used for semi-supervised

*This research was conducted during the author's internship at Microsoft Research.

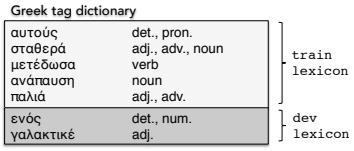

Figure 1: A tag dictionary (lexicon) for Greek. The splits into lex_{train} and lex_{dev} are discussed in §2.

learning and not only for model selection.

2 Model selection and training for type-supervised learning

Notation. In type-supervised learning, we have unlabeled text $\mathcal{T} = \{\mathbf{x}\}$ of token sequences, and a lexicon lex which lists possible labels for word types. Model training finds the model parameters θ which minimize a training loss function $L(\theta; lex, \mathcal{T}, \mathbf{h})$. We use $\mathbf{h}$ to represent the configurations and modeling decisions (also known as *hyperparameters*). Examples include the dependency structure between variables, feature templates, and regularization strengths. Given a set of fully-specified hyperparameter configurations $\{\mathbf{h}_1, \ldots, \mathbf{h}_M\}$, model selection aims to find the configuration $\mathbf{h}_{\hat{m}}$ that maximizes the expected performance of the corresponding model $\theta_{\hat{m}}$ according to a suitable accuracy measure. $\mathbf{x}$ is a token sequence, $\mathbf{y}$ is a label sequence, and $lex[\mathbf{x}]$ is the set of label sequences compatible with token sequence $\mathbf{x}$ according to lex.

Task. For the application in this paper, the task is type-supervised POS tagging, and the parametric model family we consider is that of featurized first order HMMs (Berg-Kirkpatrick et al., 2010). The hyperparameters specify the feature set used and the strength of an L_2 regularizer on the parameters.

Evaluation function. The evaluation function used in model selection is the main focus of this work. We use a function $eval(m, \mathcal{T}_{dev})$ to estimate the performance of the model trained with hyperparameters $\mathbf{h}_m$ on a development set $\mathcal{T}_{dev}$. In the following subsections, we discuss definitions of $eval$ when the development set $\mathcal{T}_{dev}$ is labeled and when it is unlabeled, respectively.

2.1 $\mathcal{T}_{dev}$ is labeled

When the development set $\mathcal{T}_{dev}$ is labeled, a natural choice of $eval$ is token-level prediction accuracy:

$$eval_{sup}(m, \mathcal{T}_{dev}) = \sum_{i=1}^{|\mathcal{T}_{dev}|} \frac{\mathbb{1}(\mathbf{y}_m[i] = \mathbf{y}_{gold}[i])}{|\mathcal{T}_{dev}|}$$

Here, we use i to index all tokens in $\mathcal{T}_{dev}$; $\mathbf{y}_{gold}[i]$ denotes the *correct* POS tag, and $\mathbf{y}_m[i]$ denotes the *predicted* POS tag of the i-th token obtained with hyperparameters $\mathbf{h}_m$.

2.2 $\mathcal{T}_{dev}$ is unlabeled

When token-supervision is not available, we cannot compute $eval_{sup}$. Instead, previous work on POS tagging with type supervision (Smith and Eisner, 2005) used:

$$eval_{cond}(m, \mathcal{T}_{dev}) = \sum_{\mathbf{x} \in \mathcal{T}_{dev}} \log \sum_{\mathbf{y} \in lex[\mathbf{x}]} p_{\theta_m}(\mathbf{y} \mid \mathbf{x}),$$

$$eval_{joint}(m, \mathcal{T}_{dev}) = \sum_{\mathbf{x} \in \mathcal{T}_{dev}} \log \sum_{\mathbf{y} \in lex[\mathbf{x}]} p_{\theta_m}(\mathbf{x}, \mathbf{y})$$

$eval_{cond}$ estimates the conditional log-likelihood of "lex-compatible" labels *given* token sequences, while $eval_{joint}$ estimates the joint log-likelihood of lex-compatible labels *and* token sequences.

The held-out lexicon criterion. We propose a new model selection criterion which estimates prediction accuracy more directly and is applicable to any model type, without requiring that the model define conditional or joint probabilities of label sequences. The idea behind this proposed criterion is simple: we hold out a portion of the lexicon entries and use it to estimate model performance as follows:

$$eval_{devlex}(m, \mathcal{T}) = \sum_{i=1:x_i \in lex_{dev}}^{|\mathcal{T}|} \frac{\mathbb{1}(\mathbf{y}_m[i] \in lex[x_i])}{|lex[x_i]| \times |\mathcal{T}_{x \in lex_{dev}}|}$$

where lex_{dev} is the held-out portion of the lexicon entries, and x_i is the i-th token in $\mathcal{T}$.

The remainder of this section details the theory behind this criterion. The expected token-level accuracy of a model trained with hyperparameters $\mathbf{h}_m$ is defined as $\mathbb{E}_{(x, y_{gold}, y_m) \sim \mathcal{D}}[\mathbb{1}(y_m = y_{gold})]$; where $\mathcal{D}$ is a joint distribution over tokens x (in context), their gold labels y_{gold}, and the predicted labels y_m (for the configuration $\mathbf{h}_m$). Since when no labeled data is available we do not have access to samples from $\mathcal{D}$, we derive an approximation to this distribution using lex and $\mathcal{T}$.

We first split the full lexicon into a training lexicon lex_{train} and a held-out (or development)

lexicon lex_{dev} (see Fig. 1), by sampling words according to their token frequency $c(x)$ in $\mathcal{T}$, and placing them in the development or training portions of the lexicon such that lex_{dev} covers 25% of the tokens in $\mathcal{T}$. The goal of the sampling process is to make the distribution of word tags for words in the development lexicon representative of the tag distribution for all words.

Given $\mathbf{h}_m$, we train a tagging model using lex_{train} and use it to predict labels y_m for all tokens in $\mathcal{T}$. We then use the word tokens covered by the development lexicon and their predicted tags y_m to approximate $\mathcal{D}$ by letting $P(y_{gold} \mid x)$ be a uniform distribution over gold labels consistent with the lexicon for x, resulting in the following approximation $P_{\mathcal{D}}(x, y_{gold}, y_m) \propto$

$$\frac{c(x, y_m) \times \mathbb{1}(x \in lex_{dev}, y_{gold} \in lex_{dev}[x])}{|lex_{dev}[x]|}$$

We then compute the expected accuracy as $eval_{devlex} = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(y_{gold} = y_m)]$, and select the hyperparameter configuration $\hat{m}$ which maximizes this criterion, then re-train the model with the full lexicon lex.[1]

3 How to best use a small labeled set $\mathcal{T}_L$?

Several prior works used a labeled set for supervised hyper-parameter selection even when only type-level supervision is assumed to be available for training (Vaswani et al., 2010; Soderland and Weld, 2014). In this section, we want to answer the question: if a small labeled set is available, what are the potential gains from using it for model selection only, versus using it for both model training and model selection?

A simple way to use a small labeled set for parameter training together with a larger unlabeled set in our type-supervised learning setting, is to do semi-supervised model training as follows (Nigam et al., 2000): Starting with our training loss function defined using a lexicon lex and unlabeled set $\mathcal{T}_U$ $L(\theta; lex, \mathcal{T}_U, \mathbf{h}_m)$, we define a combined loss function using both the unlabeled set $\mathcal{T}_U$ and the labeled set $\mathcal{T}_L$: $L(\theta; lex, \mathcal{T}_U, \mathbf{h}_m) + \lambda L(\theta; lex, \mathcal{T}_L, \mathbf{h}_m)$. We then select parameters θ_m to minimize the new loss function, where λ is now an additional hyperparameter that usually gives more weight to the labeled set. An advantage of this method is that it can be applied to any type-supervised model using less than 100 lines of code.[2] We implement this method for semi-supervised training, and we use the labeled set both for semi-supervised model training and for hyper-parameter selection using a standard five-fold cross-validation approach.[3]

4 Experiments

We evaluate the introduced methods for model selection and training with type supervision in two type-supervised settings: when no labeled examples are available, and when a small number of labeled examples are available.

We use a feature-rich first-order HMM model (Berg-Kirkpatrick et al., 2010) with an L_2 prior on feature weights.[4] Instead of using a multinomial distribution for the local emissions and transitions, this model uses a log-linear distribution (i.e., $p(x_i \mid y_i) \propto \exp \lambda^\top f(x_i, y_i)$) with a feature vector f and a weight vector λ. We use the feature set described in (Li et al., 2012): transition features, word-tag features ($\langle y_i, x_i \rangle$) (lowercased words with frequency greater than a threshold), whether the word contains a hyphen and/or starts with a capital letter, character suffixes, and whether the word contains a digit. We initialize the transition and emission distributions of the HMM using unambiguous words as proposed by (Zhang and DeNero, 2014). **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). We map the POS labels in the CoNLL datasets to the universal POS tagset (Petrov et al., 2012). We use the tag dictionaries provided by Li et al. (2012). **Model configurations.** For each

[1] Note that this criterion underestimates the performance of all models in consideration by virtue of evaluating model versions trained using a subset of the full lexicon, but it can still be useful for ranking the models.

[2] The labeled data loss is the negative joint log-likelihood of the observed token sequences and labels: $-\sum_{\mathbf{x},\mathbf{y} \in \mathcal{T}_L} \log p_\theta(\mathbf{x}, \mathbf{y})$.

[3] We split the labeled set into five folds, and for each setting of the hyper-parameters train five different models on $\frac{4}{5}$-ths of the data, estimating accuracy on the remaining $\frac{1}{5}$-th. We average the accuracy estimates from different folds and use this as a combined estimate of the accuracy of a model trained using the full labeled and unlabeled set, given these hyperparameters. After selecting a configuration of hyperparameters using cross-validation, we then use this configuration and retrain the model on all available data.

[4] We used a first-order HMM for simplicity, but it is possible to obtain better results using a second-order HMM (Li et al., 2012).

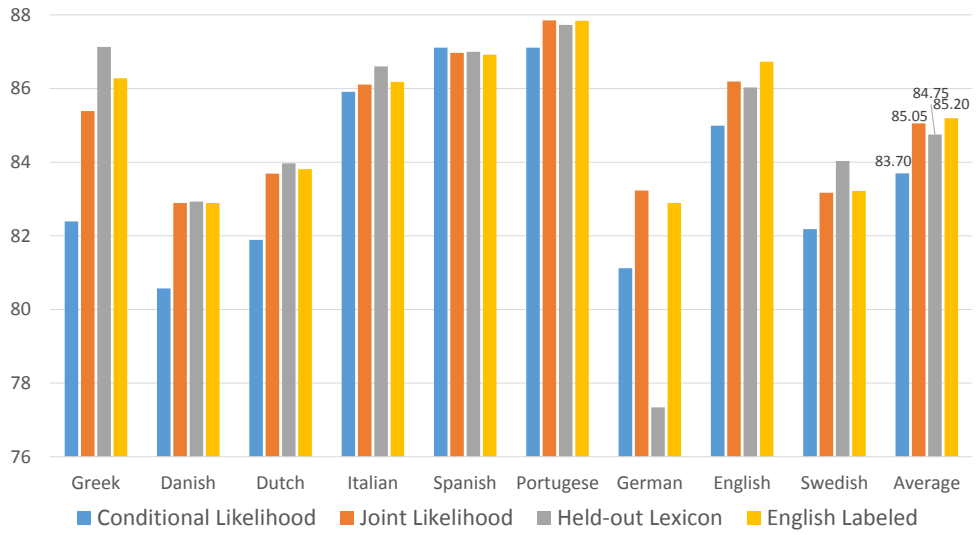

Figure 2: Token-level accuracy when doing training and model selection with no labeled data. Model selection with different criteria (left to right): conditional log-likelihood, joint log-likelihood, held-out lexicon, and English-labeled.

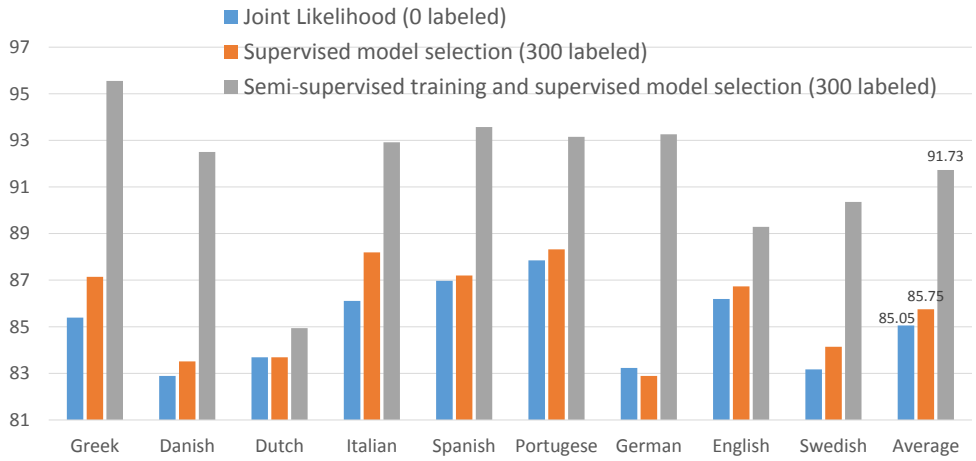

Figure 3: Token-level accuracy with (left to right): no labeled data, model selection on labeled data (300 sentences), semi-supervised training + model selection on labeled data (300 sentences).

language, we consider $M = 15$ configurations of the hyperparameters which vary in the L_2 regularization strength (5 values), and the minimum word frequency for word-tag features (3 values). When a small labeled set is available we additionally choose one of 3 values for the weight of the labeled set (see Section 3). We report final performance of models selected using different criteria using token-level accuracy on an unseen test set.

No labeled examples. When no labeled examples are available, we do model training and selection using only unlabeled text $\mathcal{T}$ and a tagging lexicon lex. We compare three type-supervised model selection criteria: conditional likelihood, joint likelihood, and the held-out lexicon criterion $eval_{devlex}$. Additionally, we include the performance of a method which selects the hyperparameters using labeled data in English and uses these (same) hyperparameters for English and all other languages (we call this method "English Labeled"). Fig. 2 shows the accuracy of the models chosen by each of the four criteria on nine languages, as well as the average accuracy across languages. The lower (upper) bounds on average performance obtained by always choosing the worst (best) hyperparameters is 82.77 (85.83). $eval_{joint}$ outperformed $eval_{cond}$ on eight out of the nine

languages and achieved a significantly higher average accuracy (85.05 vs 83.70). $eval_{devlex}$ outperformed $eval_{joint}$ on six out of nine languages, but did significantly worse on one language (German), which resulted in a slightly lower average accuracy. Choosing the hyper-parameters using English labeled data and using the same hyper-parameters for all languages performed comparably to $eval_{joint}$, with slightly higher average accuracy even when limited to the non-English languages (85.0 vs 84.9). Overall the results showed that the conditional log-likelihood criterion was dominated by the other three, which were comparable in average accuracy. Looking at the eight languages excluding English (since one criterion uses labeled data for English), the newly proposed held-out lexicon criterion was the winning method on five out of eight languages, $eval_{cond}$ was best on one language, $eval_{joint}$ was best (or tied for best) on two, and English-labeled was tied for best on one language.

Few labeled examples. We consider two ways of leveraging the labeled examples: (*i*) type-supervised model training + supervised model selection: only use unlabeled examples to optimize model parameters, then use the labeled examples for supervised model selection with $eval_{sup}$, and (*ii*) semi-supervised model training + supervised model selection (see Section 3 for details). Fig. 3 shows how much we can improve on the method with highest average accuracy from Figure 2 ($eval_{joint}$), when a small number of examples is available. Using the 300 labeled sentences for semi-supervised training and model selection reduced the error by 44.6% (comparing to the model with best average accuracy using only type-level supervision with average performance of 85.05, the semi-supervised average is 91.8). In contrast, using the 300 sentences to select hyper-parameters only reduced the error by less than 5% (the average accuracy was 85.75). Even when only 50 labeled sentences are used for semi-supervised training and supervised model selection, we still see a boost to average accuracy of 89% (results not shown in the Figure).

5 Conclusion

We presented the first comparative evaluation of model selection criteria for type-supervised POS-tagging on many languages. We introduced a novel, generally applicable model selection criterion which outperformed previously proposed ones for a majority of languages. We evaluated the utility of a small labeled set for model selection versus model training, and showed that when such labeled set is available, it should not be used solely for supervised model selection, because using it additionally for model parameter training provides strikingly larger accuracy gains.

Acknowledgments

We thank Nathan Schneider and the anonymous reviewers for helpful suggestions.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proc. of NAACL*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL-X*.

Andrew Carlson, Scott Gaffney, and Flavian Vasile. 2009. Learning a named entity tagger from gazetteers with the partial perceptron. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 7–13.

Mark Craven, Johan Kumlien, et al. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86.

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *Proc. of EMNLP*.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proc. of NAACL-HLT*, pages 138–147.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Annual meeting-association for computational linguistics*.

Anders Johannsen, Dirk Hovy, Héctor Martınez Alonso, Barbara Plank, and Anders Søgaard. 2014. More or less supervised supersense tagging of twitter. *Proc. of* SEM*, pages 1–11.

Shen Li, João Graça, and Ben Taskar. 2012. Wikily supervised part-of-speech tagging. In *Proc. of EMNLP*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39.

Joakim Nivre, Johan Hall, Sandra Kubler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of CoNLL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proc. of LREC*, May.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proc. of ACL*.

Mitchell Koch John Gilmer Stephen Soderland and Daniel S Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proc. of EMNLP*.

Anders Søgaard. 2011. Semisupervised condensed nearest neighbor for part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 48–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Ashish Vaswani, Adam Pauls, and David Chiang. 2010. Efficient optimization of an mdl-inspired objective function for unsupervised part-of-speech tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 209–214. Association for Computational Linguistics.

Hui Zhang and John DeNero. 2014. Observational initialization of type-supervised taggers. In *Proceedings of the Association for Computational Linguistics (Short Paper Track)*.