An Online Algorithm for Learning over Constrained Latent Representations using Multiple Views *

Ann Clifton and **Max Whitney** and **Anoop Sarkar**
School of Computing Science
Simon Fraser University
8888 University Drive
Burnaby BC. V5A 1S6. Canada
{ann_clifton, mwhitney, anoop}@sfu.ca

Abstract

We introduce an online framework for discriminative learning problems over hidden structures, where we learn both the latent structure and the classifier for a supervised learning task. Previous work on leveraging latent representations for discriminative learners has used batch algorithms that require multiple passes though the entire training data. Instead, we propose an online algorithm that efficiently jointly learns the latent structures and the classifier. We further extend this to include multiple views on the latent structures with different representations. Our proposed online algorithm with multiple views significantly outperforms batch learning for latent representations with a single view on a grammaticality prediction task.

1 Introduction

Natural language data is implicitly richly structured, and making use of that structure can be valuable in a wide variety of NLP tasks. However, finding these latent structures is a complex task of its own right. Early work used a two-phase pipeline process, in which the output of a structure prediction algorithm (e.g. a noun phrase finder) acts as fixed input features to train a classifier for a different task (e.g. grammaticality prediction). Chang et al. (2009), Das and Smith (2009), Goldwasser and Roth (2008), and Mccallum and Bellare (2005) have shown that this approach can propagate error from the structured prediction to the task-specific classifier. Recent work has combined unsupervised learning of (latent) structure prediction with a supervised learning approach for the task. Work in this vein has focused on jointly learning the latent structures together with the task-specific classifier (Cherry and Quirk, 2008; Chang et al., 2010). Chang et al. (2010) in particular introduce a framework for solving classification problems using constraints over latent structures, referred to as Learning over Constrained Latent Representations (LCLR). We extend this framework for discriminative joint learning over latent structures to a novel online algorithm. Our algorithm learns the latent structures in an unsupervised manner, but it can be initialized with the model weights from a supervised learner for the latent task trained on some (other) annotated data. This can be seen as a form of domain adaptation from the supervised latent structure training data to the different classification task.

We evaluate our algorithm in comparison to the LCLR batch method on a grammaticality test using a discriminative model that learns shallow parse (chunk) structures. Our online method has standard convergence guarantees for a max-margin learner, but attains higher accuracy. Furthermore, in practice we find that it requires fewer passes over the data.

We also explore the use of allowing multiple views on the latent structures using different representations in the classifier. This is inspired by Shen and Sarkar (2005), who found that using a majority voting approach on multiple representations of the latent structures on a chunking task outperformed both a single representation as well as voting between multiple learning models. We show that the multiple-view approach to latent structure learning yields improvements over the single-view classifier.

2 The Grammaticality Task

To evaluate our algorithms, we use a discriminative language modeling task. A well-known limitation of n-gram LMs is that they are informed only by the previously seen word histories of a

*This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the third author.

International Joint Conference on Natural Language Processing, pages 1072–1076,
Nagoya, Japan, 14-18 October 2013.

fixed maximum length; they ignore dependencies between more distant parts of the sentence. Consider examples generated by a 3-gram LM:

- chemical waste and pollution control (amendment) bill , all are equal , and , above all else .
- kindergartens are now .

These fragments are composed of viable trigrams, but a human could easily judge them to be ungrammatical. However, if a language model used latent information like a shallow syntactic parse, it could also recognize the lack of grammaticality.

Discriminative models can take into account arbitrary features of data, and thus may be able to avoid the shortcomings of n-gram LMs in judging the grammaticality of text. In the case of language modeling, however, there is no obvious choice of categories between which the model should discriminate. Cherry and Quirk (2008) show that by following the pseudo-negative examples approach of Okanohara and Tsujii (2007), they can build a syntactic discriminative LM that learns to distinguish between samples from a corpus generated by human speakers (positives) and samples generated by an n-gram model (negatives).

Our approach is similar to Cherry and Quirk (2008), but they use probabilistic context-free grammar (PCFG) parses as latent structure, use a latent SVM as the learning model (we use latent passive-aggressive (PA) learning), and they handle negative examples differently. Instead of PCFG parsing, we use a chunking representation of sentence structure, which can be seen as a shallow parse, in which each word in the sentence is tagged to indicate phrase membership and boundaries.

Our model simultaneously learns to apply multiple sets of chunk tags to produce chunkings representing sentence structure and to prefer the shallow parse features of the human sentences to those sampled from an n-gram LM. The latent chunker will assign chunk structure to examples that yield the widest margin between the positive (grammatical) and negative (ungrammatical) examples.

3 Latent Structure Classifier

Our classifier is trained by simultaneously searching for the highest scoring latent structure while classifying data instances. Here we extend the latent learning framework due to Chang et al. (2010) from a batch setting to an online setting that uses passive-aggressive (PA) updates (Crammer and Singer, 2001).

3.1 PA Learning

The latent structure classifier training uses a decision function that searches for the best structure $z_i^* \in Z(x_i)$ for each training sentence x_i with a space of possible structures $Z(x_i)$ according to feature weights $\mathbf{w}$, i.e.:

$$f_w(x_i) = \arg\max_{z_i} \mathbf{w} \cdot \phi(x_i, z_i) \quad (1)$$

where $\phi(x_i, z_i)$ is a feature vector over the sentence-parse pair. The sign of the prediction $y_i^* \mathbf{w} \cdot \phi(x_i, z_i^*)$ determines the classification of the sentence x_i.

Using PA max-margin training (Crammer and Singer, 2001), we incorporate this decision function into our global objective, searching for the $\mathbf{w}$ that minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{X} \ell(\mathbf{w} \cdot (f(x_i), y_i)), \quad (2)$$

where ℓ is a loss function; we use hinge loss. At each iteration, for each example x_i we find and update according to a new weight vector $\mathbf{w}'$ that minimizes:

$$\frac{1}{2}\|\mathbf{w} - \mathbf{w}'\|^2 + \tau(1 - y_i(\mathbf{w}' \cdot \phi(x_i, z_i^*)), \quad (3)$$

where $\mathbf{w}$ is the previous weight vector, z_i^* is the structure found by Eqn. (1), $y_i \in \{-1, 1\}$ is the example's true label (ungrammatical/grammatical), and $\tau \geq 0$ is a Lagrange multiplier proportional to the example loss, thus penalizing classification examples in proportion to the extent that they violate the margin (see Alg. 1).

3.2 Optimization Method

Since Eqn. (3) contains an inner max over z_i^*, it is not convex for the positive examples, since it is the maximum of a convex function (zero) and a concave function $(1 - y_i(\mathbf{w}' \cdot \phi(x_i, z_i^*))$. In hinge loss, driving the inner function to higher values minimizes the outer problem for negative examples, but maximizes it for the positives. So, as in LCLR, we hold the latent structures fixed for the positive examples but can perform inference to solve the inner minimization problem for the negatives.

3.3 Online Training

Our online training method is shown as algorithm 1. It applies the structured prediction and PA update of section 3 on a per-example basis in a variant of the cutting plane algorithm discussed in

1 initialize $\mathbf{w}_0$
2 **for** $t = 0, ..., T-1$ **do**
3 **for** *each training example* x_i *in* X **do**
4 **repeat**
5 find $z_i^* = \arg\max_{z_i} \mathbf{w}_t \cdot \phi(x_i, z_i)$
6 let $y_i^* = \mathbf{w}_t \cdot \phi(x_i, z_i^*)$
7 let loss $l_t = \max\{0, 1 - y_i y_i^*)\}$
8 let multiplier $\tau_t = \frac{l_t}{\|\phi(x_i, z_i^*)\|^2}$
9 update $\mathbf{w}_{t+1} := \mathbf{w}_t + \tau_t y_i \phi(x_i, z_i^*)$
10 **until** $y_i > 0$ *or* ($y_i^* = y_i$ *if* $y_i < 0$);
11 **return** $\mathbf{w}_T$

Algorithm 1: Online PA algorithm for binary classification with latent structures.

Joachims and Yu (2009). Since for the positive examples the latent structures are fixed per-iteration, it does a single search and update step for each example at each iteration. For negative examples it repeats the prediction and PA update for each example until the model correctly predicts the label (i.e. until $y_i^* = y_i$). Because of the intractability to compute all possible negative structures, we use the approximation of the single-best structure for each negative example. We re-decode the negative examples until the highest scoring structure is correctly labeled as negative. This approximation is analogous to the handling of inference over negative examples in the batch algorithm described in Chang et al. (2010). In the batch version, however, updates for all negative examples are performed at once and all are re-decoded until no new structures are found for any single negative example.

3.4 Multiple Views on Latent Representations

Shen and Sarkar (2005) find that using multiple chunking representations is advantageous for the chunking task. Moreover, they demonstrate that the careful selection of latent structure can yield more helpful features for a task-specific classifier. We thus perform inference separately to generate distinct latent structures for each of their five chunking representations (which are mostly from (Sang and Veenstra, 1999)) at line 5 of Alg. 1; at line 6 we evaluate the dot product of the weight vector with the features from the combined outputs of the different views.

Each of the views use a different representation of the chunk structures, which we will only briefly describe due to space limitations; for more detailed information, please see Shen and Sarkar (2005). Each representation uses a set of tags to label each token in a sentence as belonging to a non-overlapping chunk type. We refer to the chunking schemas as IOB1, IOB2, IOE1, IOE2, and O+C. The total set of tags for each of the representations are B- (current token begins a chunk), I- (current token is inside a chunk), E- (current token ends a chunk), S- (current token is in a chunk by itself), and O (current token is outside of any chunk). All chunks except O append the part-of-speech tag of the token as a suffix. Table 1 shows the different chunking schemas on an example sentence.

Token	IOB1	IOB2	IOE1	IOE2	O+C
In	O	O	O	O	O
early	I	B	I	I	B
trading	I	I	I	E	E
in	O	O	O	O	O
Hong	I	B	I	I	B
Kong	I	I	E	E	E
Monday	B	B	I	E	S
,	O	O	O	O	O
gold	I	B	I	E	S
was	O	O	O	O	O
quoted	O	O	O	O	O
at	O	O	O	O	O
$	I	B	I	I	B
366.50	I	I	E	E	E
an	B	B	I	I	B
ounce	I	I	I	E	E
.	O	O	O	O	O

Table 1: The five different chunking representations for the example sentence "In early trading in Hong Kong Monday , gold was quoted at $ 366.50 an ounce ."

Each of these chunking schemas can be conceived as a different kind of expert. Of the inside/outside schemas, the IOB variants focus on detecting where a chunk begins; the IOE variants focus on the chunk's end. O+C gives a more fine-grained representation of the chunking.

We use dynamic programming to find the best chunking for each representation. The features of $\phi(x, z)$ are 1-, 2-, 3-grams of words and POS tags paired with the chunk tags, as well as bigrams of chunk tags. We use entirely separate chunk tags for each representation. E.g., although each representation uses an "O" tag to indicate a word outside of any phrase, we consider the "O" for each representation to be distinct.

We combine the multiple views in two different ways: 1) we simply concatenate the features from each structured prediction view into a larger feature vector and the weights are trained on the supervised learning task, and 2) before training on the supervised learning task we first convert all representations to a common representation, O+C (since it includes the union of the tagging distinctions from all 5 views, it does not cause loss of

information from any single view), and then we perform a majority vote for each tag in the prediction. We convert the winning sequence of predicted tags back to each representation and concatenate the features from each view as before and train on the supervised learning task.

4 Experiments

For the chunkers we used the CONLL 2000 tagset (23 chunk tags), modified for the five chunking representations of (Shen and Sarkar, 2005). We initialized the weights using a perceptron chunker. The chunker-classifier can either be started with a zero weight vector or with weights from training on the chunking task. For the latter, we used weights from supervised discriminative training against gold-standard chunking. To transfer the weights to the classifier, we scaled them to the range of values observed after training the zero-initialized chunker-classifier. For training data we used the English side of the HK Chinese-English parallel corpus, using 50,000 sentences as positive examples. For negative examples we used the pseudo-negative approach of Okanohara and Tsujii (2007): we trained a standard 3-gram language model on the 50,000 sentences plus 450,000 additional sentences from the same corpus. From this we sampled 50,000 sentences to create the negative training data set.

We evaluated the discriminative LMs on the classification task of distinguishing real grammatical sentences from generated pseudo-negative sentences. As test data we used the Xinhua data from the English Gigaword corpus. We used the first 3000 sentences as positive examples. For negative examples we trained a 3-gram LM on the first 500,000 examples (including those used for positive data).We used this 3-gram LM to generate five separate 3000 example negative data sets. To account for random variation due to using pseudo-negatives, results are reported as a mean over the positive data paired with each negative set. We evaluated our algorithms against LCLR as a baseline.[1] Table 2 shows that our online algorithm with multiple views significantly outperforms the previous approaches. We omit a detailed experimental report of the behaviour of the online algorithm due to lack of space, but our findings were 1) that the batch models were slower to improve than the online versions on test-set accuracy, and 2) the online algorithm requires fewer updates total in training compared to the batch version.

Model	Accuracy %
LCLR	90.27
LCLR-variant	94.55
online single-view	98.75
+ multi-view	98.70
+ majority vote	98.78

Table 2: Classification accuracy after 40 outer iterations.

5 Related and Future Work

As discussed, our work is most similar to Chang et al. (2010). We expand upon their framework by developing an efficient online algorithm and exploring learning over multiple views on latent representations. In terms of the task, max-margin LMs for speech recognition focus on the word prediction task (Gao et al., 2005; Roark et al., 2007; Singh-Miller and Collins, 2007). This focus is also shared by other syntactic LMs (Chelba and Jelinek, 1998; Xu et al., 2002; Schwartz et al., 2011; Charniak, 2001) which use syntax but rely on supervised data to train their parsers. Charniak et al. (2003) and Shen et al. (2010) use parsing based LMs for machine translation which are not whole-sentence models and they also rely on supervised parsers. Our focus is on using unsupervised latent variables (optionally initialized from supervised data) and training whole-sentence discriminative LMs. Our chunker model is related to the semi-Markov model in Okanohara and Tsujii (2007), but ours can take advantage of latent structures. Our work is related to Cherry and Quirk (2008) but differs in ways previously described.

In future work, we plan to apply our algorithms to a wider range of tasks, and we will present an analysis of the properties of online learning algorithms over latent structures. We will explore other ways of combining the latent structures from multiple views, and we will examine the use of joint inference across multiple latent representations.

[1] We implemented two batch baselines. The first is a strict implementation of the LCLR algorithm as in Chang et al. (2010), with per-outer-iteration example caching (LCLR); we use a PA large-margin classifier instead of an SVM. However, we found that this algorithm severely overfits to our task. So, we also implemented a variant ("LCLR-variant") that skips the inference step in the inner loop. This treated the latent structures from the inference step of the outer loop as fixed, but relabeled and updated accordingly until convergence, then resumed the next outer iteration.

References

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Yuancheng Tu. 2009. Unsupervised constraint driven learning for transliteration discovery. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 299–307, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *HLT-NAACL*, pages 429–437.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based Language Models for Machine Translation. In *Proc. of MT Summit IX*.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proc. of ACL 2001*, pages 124–131, Toulouse, France, July. Association for Computational Linguistics.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proc. of ACL 1998*, pages 225–231, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Colin Cherry and Chris Quirk. 2008. Discriminative, syntactic language modeling through latent SVMs. In *Proc. of AMTA 2008*.

Koby Crammer and Yoram Singer. 2001. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, January.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. 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 1 - Volume 1*, ACL '09, pages 468–476, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jianfeng Gao, Hao Yu, Wei Yuan, and Peng Xu. 2005. Minimum sample risk methods for language modeling. In *Proc. of ACL*, pages 209–216, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Dan Goldwasser and Dan Roth. 2008. Transliteration as constrained optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 353–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thorsten Joachims and Chun-Nam John Yu. 2009. Sparse kernel svms via cutting-plane training. *Machine Learning*, 76(2-3):179–193.

Andrew Mccallum and Kedar Bellare. 2005. A conditional random field for discriminatively-trained finite-state string edit distance. In *In Conference on Uncertainty in AI (UAI*.

Daisuke Okanohara and Jun'ichi Tsujii. 2007. A discriminative language model with pseudo-negative samples. In *Proc. of ACL 2007*, pages 73–80, Prague, Czech Republic, June. Association for Computational Linguistics.

Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Computer Speech and Language*, 21(2):373–392.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, EACL '99, pages 173–179, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lane Schwartz, Chris Callison-Burch, William Schuler, and Stephen Wu. 2011. Incremental syntactic language models for phrase-based translation. In *Proc. of ACL-HLT 2011*, pages 620–631, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hong Shen and Anoop Sarkar. 2005. Voting between multiple data representations for text chunking. In *Canadian Conference on AI*, pages 389–400.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-dependency statistical machine translation. *Comput. Linguist.*, 36(4):649–671.

Natasha Singh-Miller and Michael Collins. 2007. Trigger-based Language Modeling using a Loss-sensitive Perceptron Algorithm. In *Proc. of ICASSP 2007*.

Peng Xu, Ciprian Chelba, and Frederick Jelinek. 2002. A study on richer syntactic dependencies for structured language modeling. In *Proc. of ACL 2002*, pages 191–198, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.