
Conditional Random Field Autoencoders for Unsupervised Structured Prediction

Waleed Ammar Chris Dyer Noah A. Smith
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{wammar, cdyer, nasmith}@cs.cmu.edu

Abstract

We introduce a framework for unsupervised learning of structured predictors with overlapping, global features. Each input’s latent representation is predicted conditional on the observed data using a feature-rich conditional random field (CRF). Then a reconstruction of the input is (re)generated, conditional on the latent structure, using a generative model which factorizes similarly to the CRF. The autoencoder formulation enables efficient exact inference without resorting to unrealistic independence assumptions or restricting the kinds of features that can be used. We illustrate connections to traditional autoencoders, posterior regularization, and multi-view learning. We then show competitive results with instantiations of the framework for two canonical tasks in natural language processing: part-of-speech induction and bitext word alignment, and show that training the proposed model can be substantially more efficient than a comparable feature-rich baseline.

1 Introduction

Conditional random fields [24] are used to model structure in numerous problem domains, including natural language processing (NLP), computational biology, and computer vision. They enable efficient inference while incorporating rich features that capture useful domain-specific insights. Despite their ubiquity in supervised settings, CRFs—and, crucially, the insights about effective feature sets obtained by developing them—play less of a role in *unsupervised* structure learning, a problem which traditionally requires jointly modeling observations and the latent structures of interest. For unsupervised structured prediction problems, less powerful models with stronger independence assumptions are standard.¹ This state of affairs is suboptimal in at least three ways: (i) adhering to inconvenient independence assumptions when designing features is limiting—we contend that effective feature engineering is a crucial mechanism for incorporating inductive bias in unsupervised learning problems; (ii) features and their weights have different semantics in joint and conditional models (see §3.1); and (iii) modeling the generation of high-dimensional observable data with feature-rich models is computationally challenging, requiring expensive marginal inference in the inner loop of iterative parameter estimation algorithms (see §3.1).

Our approach leverages the power and flexibility of CRFs in unsupervised learning without sacrificing their attractive computational properties or changing the semantics of well-understood feature sets. Our approach replaces the standard joint model of observed data and latent structure with a two-layer **conditional random field autoencoder** that first generates latent structure with a CRF (conditional on the observed data) and then (re)generates the observations conditional on just the predicted structure. For the reconstruction model, we use distributions which offer closed-form maximum

¹For example, a first-order hidden Markov model requires that $y_i \perp x_{i+1} \mid y_{i+1}$ for a latent sequence $\mathbf{y} = \langle y_1, y_2, \dots \rangle$ generating $\mathbf{x} = \langle x_1, x_2, \dots \rangle$, while a first-order CRF allows y_i to directly depend on x_{i+1} .

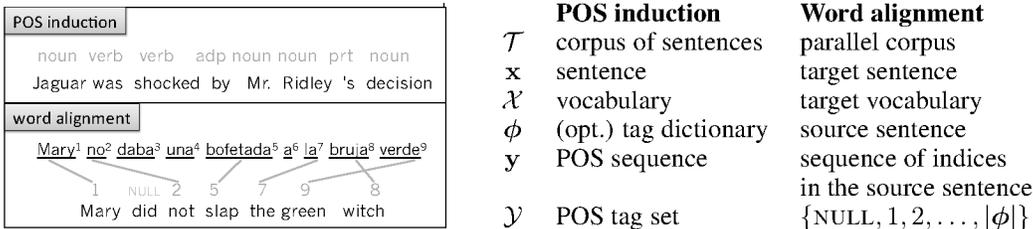


Figure 1: Left: Examples of structured observations (in black), hidden structures (in gray), and side information (underlined). Right: Model variables for POS induction (§A.1) and word alignment (§A.2). A parallel corpus consists of pairs of sentences (“source” and “target”).

likelihood estimates (§2). The proposed architecture provides several mechanisms for encouraging the learner to use its latent variables to find intended (rather than common but irrelevant) correlations in the data. First, hand-crafted feature representations—engineered using knowledge about the problem—provide a key mechanism for incorporating inductive bias. Second, by reconstructing a transformation of the structured input. Third, it is easy to simultaneously learn from labeled and unlabeled examples in this architecture, as we did in [26]. In addition to the modeling flexibility, our approach is computationally efficient in a way achieved by no other unsupervised, feature-based model to date: under a set of mild independence assumptions regarding the reconstruction model, per-example inference required for learning in a CRF autoencoder is no more expensive than in a supervised CRF with the same independence assumptions.

In the next section we describe the modeling framework, then review related work and show that, although our model admits more powerful features, the inference required for learning is simpler (§3). We conclude with experiments showing that the proposed model achieves state-of-the-art performance on two unsupervised tasks in NLP: POS induction and word alignment, and find that it is substantially more efficient than MRFs using the same feature set (§4).

2 Conditional Random Field Autoencoder

Given a training set \mathcal{T} of observations (e.g., sentences or pairs of sentences that are translationally equivalent), consider the problem of inducing the hidden structure in each observation. Examples of hidden structures include shallow syntactic properties (part-of-speech or POS tags), correspondences between words in translation (word alignment), syntactic parses and morphological analyses.

Notation. Let each observation be denoted $\mathbf{x} = \langle x_1, \dots, x_{|\mathbf{x}|} \rangle \in \mathcal{X}^{|\mathbf{x}|}$, a variable-length tuple of discrete variables, $x \in \mathcal{X}$. The hidden variables $\mathbf{y} = \langle y_1, \dots, y_{|\mathbf{y}|} \rangle \in \mathcal{Y}^{|\mathbf{y}|}$ form a tuple whose length is determined by \mathbf{x} , also taking discrete values.² We assume that $|\mathcal{Y}|^{|\mathbf{y}|} \ll |\mathcal{X}|^{|\mathbf{x}|}$, which is typical in structured prediction problems. Fig. 1 (right) instantiates \mathbf{x} , \mathcal{X} , \mathbf{y} , and \mathcal{Y} for two NLP tasks.

Our model introduces a new observed variable, $\hat{\mathbf{x}} = \langle \hat{x}_1, \dots, \hat{x}_{|\hat{\mathbf{x}}|} \rangle$. In the basic model in Fig. 2 (left), $\hat{\mathbf{x}}$ is a copy of \mathbf{x} . The intuition behind this model is that a good hidden structure should be a likely *encoding* of observed data and should permit *reconstruction* of the data with high probability.

Sequential latent structure. In this paper, we focus on *sequential* latent structures with first-order Markov properties, i.e., $y_i \perp y_j \mid \{y_{i-1}, y_{i+1}\}$, as illustrated in Fig. 2 (right). This class of latent structures is a popular choice for modeling a variety of problems such as human action recognition [47], bitext word alignment [6, 45, 4], POS tagging [28, 20], acoustic modeling [19], gene finding [27], and transliteration [36], among others. Importantly, we make no assumptions about conditional independence between any y_i and \mathbf{x} .

Eq. 1 gives the parametric form of the model for sequence labeling problems. λ and θ are the parameters of the encoding and reconstruction models, respectively. \mathbf{g} is a vector of clique-local

²In the interest of notational simplicity, we conflate random variables with their values.

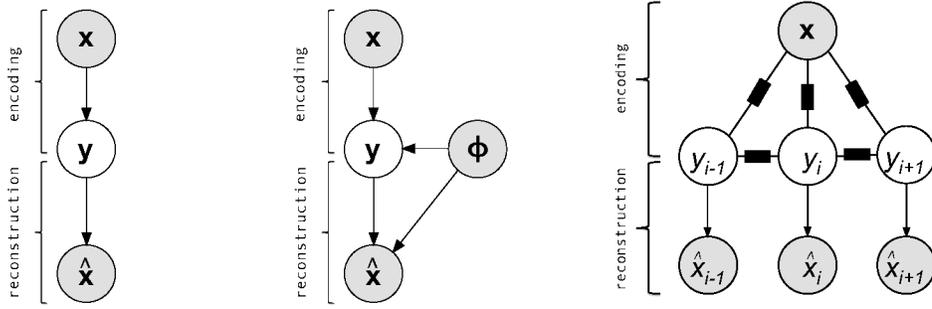


Figure 2: Graphical model representations of CRF autoencoders. Left: the basic autoencoder model where the observation \mathbf{x} generates the hidden structure \mathbf{y} (encoding), which then generates $\hat{\mathbf{x}}$ (reconstruction). Center: side information (ϕ) is added. Right: a factor graph showing first-order Markov dependencies among elements of the hidden structure \mathbf{y} .

feature functions.³

$$\begin{aligned}
 p_{\lambda, \theta}(\hat{\mathbf{x}} | \mathbf{x}) &= \sum_{\mathbf{y}} p_{\lambda}(\mathbf{y} | \mathbf{x}) p_{\theta}(\hat{\mathbf{x}} | \mathbf{y}) = \sum_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{x}|}} \frac{e^{\sum_{i=1}^{|\mathbf{x}|} \lambda^{\top} \mathbf{g}(\mathbf{x}, y_i, y_{i-1}, i)}}{\sum_{\mathbf{y}' \in \mathcal{Y}^{|\mathbf{x}|}} e^{\sum_{i=1}^{|\mathbf{x}|} \lambda^{\top} \mathbf{g}(\mathbf{x}, y'_i, y'_{i-1}, i)}} \prod_{i=1}^{|\mathbf{x}|} p_{\theta}(\hat{x}_i | y_i) \\
 &= \frac{\sum_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{x}|}} \exp\left(\sum_{i=1}^{|\mathbf{x}|} \log \theta_{\hat{x}_i | y_i} + \lambda^{\top} \mathbf{g}(\mathbf{x}, y_i, y_{i-1}, i)\right)}{\sum_{\mathbf{y}' \in \mathcal{Y}^{|\mathbf{x}|}} \exp \sum_{i=1}^{|\mathbf{x}|} \lambda^{\top} \mathbf{g}(\mathbf{x}, y'_i, y'_{i-1}, i)} \quad (1)
 \end{aligned}$$

Encoding and reconstruction. We model the *encoding* part with a CRF, which allows us to exploit features with global scope in the structured observation \mathbf{x} , while keeping exact inference tractable (since the model does not generate \mathbf{x} , only conditions on it). The *reconstruction* part, on the other hand, grounds the model by generating a copy of the structured observations. We use a simple categorical (i.e., multinomial) distribution to independently generate \hat{x}_i given y_i .⁴ Fig. 2 (right) is an instance of the model for POS induction with a sequential latent structure; each \hat{x}_i is generated from $p_{\theta}(\hat{x}_i | y_i)$.

We emphasize the importance of allowing modelers to define intuitive feature templates in a flexible manner. The need to efficiently add inductive bias via feature engineering has been the primary drive for developing CRF autoencoders. For example, morphology, word spelling information, and other linguistic knowledge encoded as features were shown to improve POS induction [38], word alignment [14], and other unsupervised learning problems. The proposed model enables the modeler to define such features at a lower computational cost *and* enables more expressive features with global scope in the structured input. For example, we found that using predictions of other models as features is an effective method for model combination in unsupervised word alignment tasks, and found that conjoining sub-word-level features of consecutive words helps disambiguate their POS labels (see Appendix A for experimental details).

Extension: side information. Our model can be easily extended to condition on more context in the encoding part, the reconstruction part, or in both parts. Let ϕ represent *side information*: additional context which we may condition on in the encoding part, the reconstruction part, or both. In our running example, side information could represent the set of possible tags for each word, a common form of weak “supervision” shown to help unsupervised POS learners. In word alignment, where $y_i = j$ indicates that x_i translates to the j th source token, we treat the source sentence as side information, making its word forms available for feature extraction.

³We define y_0 to be a fixed “start” tag.

⁴Other possible parameterizations of the reconstruction model we would like to experiment with include a multivariate Gaussian for generating word embeddings and a naïve Bayes model for generating individual features of a word independently conditioned on the corresponding label.

Extension: partial reconstruction. In our running POS example, the reconstruction model $p_{\theta}(\hat{x}_i | y_i)$ defines a distribution over words given tags. Because word distributions are heavy-tailed, estimating such a distribution reliably is quite challenging. Our solution is to define a function $\pi : \mathcal{X} \rightarrow \hat{\mathcal{X}}$ such that $|\hat{\mathcal{X}}| \ll |\mathcal{X}|$, and let $\hat{x}_i = \pi(x_i)$ be a deterministic transformation of the original structured observation. We can add indirect supervision by defining π such that it represents observed information relevant to the latent structure of interest. For example, we found reconstructing Brown clusters [5] of tokens instead of their surface forms to improve POS induction. Other possible reconstructions include word embeddings, morphological and spelling features of words.

More general graphs. We presented the CRF autoencoder in terms of sequential Markovian assumptions for ease of exposition; however, this framework can be used to model arbitrary hidden structures. For example, instantiations of this model can be used for unsupervised learning of parse trees [21], semantic role labels [42], and coreference resolution [35] (in NLP), motif structures [1] in computational biology, and object recognition [46] in computer vision. The requirements for applying the CRF autoencoder model are:

- An encoding discriminative model defining $p_{\lambda}(\mathbf{y} | \mathbf{x}, \phi)$. The encoder may be any model family where *supervised* learning from $\langle \mathbf{x}, \mathbf{y} \rangle$ pairs is efficient.
- A reconstruction model that defines $p_{\theta}(\hat{\mathbf{x}} | \mathbf{y}, \phi)$ such that inference over \mathbf{y} given $\langle \mathbf{x}, \hat{\mathbf{x}} \rangle$ is efficient.
- The independencies among $\mathbf{y} | \mathbf{x}, \hat{\mathbf{x}}$ are not strictly weaker than those among $\mathbf{y} | \mathbf{x}$.

2.1 Learning & Inference

Model parameters are selected to maximize the regularized conditional log likelihood of reconstructed observations $\hat{\mathbf{x}}$ given the structured observation \mathbf{x} :

$$\ell(\boldsymbol{\lambda}, \boldsymbol{\theta}) = R_1(\boldsymbol{\lambda}) + R_2(\boldsymbol{\theta}) + \sum_{\langle \mathbf{x}, \hat{\mathbf{x}} \rangle \in \mathcal{T}} \log \sum_{\mathbf{y}} p_{\lambda}(\mathbf{y} | \mathbf{x}) \times p_{\theta}(\hat{\mathbf{x}} | \mathbf{y}) \quad (2)$$

We apply block coordinate descent, alternating between maximizing with respect to the CRF parameters ($\boldsymbol{\lambda}$ -step) and the reconstruction parameters ($\boldsymbol{\theta}$ -step). Each $\boldsymbol{\lambda}$ -step applies one or two iterations of a gradient-based convex optimizer.⁵ The $\boldsymbol{\theta}$ -step applies one or two iterations of EM [10], with a closed-form solution in the M-step in each EM iteration. The independence assumptions among \mathbf{y} make the marginal inference required in both steps straightforward; we omit details for space.

In the experiments below, we apply a squared L_2 regularizer for the CRF parameters $\boldsymbol{\lambda}$, and a symmetric Dirichlet prior for categorical parameters $\boldsymbol{\theta}$.

The asymptotic runtime complexity of each block coordinate descent iteration, assuming the first-order Markov dependencies in Fig. 2 (right), is:

$$O(|\boldsymbol{\theta}| + |\boldsymbol{\lambda}| + |\mathcal{T}| \times |\mathbf{x}|_{max} \times |\mathcal{Y}|_{max} \times (|\mathcal{Y}|_{max} \times |F_{y_{i-1}, y_i}| + |F_{\mathbf{x}, y_i}|)) \quad (3)$$

where F_{y_{i-1}, y_i} are the active “label bigram” features used in $\langle y_{i-1}, y_i \rangle$ factors, $F_{\mathbf{x}, y_i}$ are the active emission-like features used in $\langle \mathbf{x}, y_i \rangle$ factors. $|\mathbf{x}|_{max}$ is the maximum length of an observation sequence. $|\mathcal{Y}|_{max}$ is the maximum cardinality⁶ of the set of possible assignments of y_i .

After learning the $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ parameters of the CRF autoencoder, test-time predictions are made using maximum a posteriori estimation, conditioning on both observations and reconstructions, i.e., $\hat{\mathbf{y}}_{MAP} = \arg \max_{\mathbf{y}} p_{\lambda, \theta}(\mathbf{y} | \mathbf{x}, \hat{\mathbf{x}})$.

3 Connections To Previous Work

This work relates to several strands of work in unsupervised learning. Two broad types of models have been explored that support unsupervised learning with flexible feature representations. Both are

⁵We experimented with AdaGrad [12] and L-BFGS. When using AdaGrad, we accumulate the gradient vectors across block coordinate ascent iterations.

⁶In POS induction, $|\mathcal{Y}|$ is a constant, the number of syntactic classes which we configure to 12 in our experiments. In word alignment, $|\mathcal{Y}|$ is the size of the source sentence plus one, therefore $|\mathcal{Y}|_{max}$ is the maximum length of a source sentence in the bitext corpus.

fully generative models that define joint distributions over \mathbf{x} and \mathbf{y} . We discuss these “undirected” and “directed” alternatives next, then turn to less closely related methods.

3.1 Existing Alternatives for Unsupervised Learning with Features

Undirected models. A Markov random field (MRF) encodes the joint distribution through local potential functions parameterized using features. Such models “normalize globally,” requiring during training the calculation of a partition function summing over all possible inputs and outputs. In our notation:

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x} \in \mathcal{X}^*} \sum_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{x}|}} \exp \boldsymbol{\lambda}^\top \bar{\mathbf{g}}(\mathbf{x}, \mathbf{y}) \quad (4)$$

where $\bar{\mathbf{g}}$ collects all the local factorization by cliques of the graph, for clarity. The key difficulty is in the summation over all possible observations. Approximations have been proposed, including contrastive estimation, which sums over subsets of \mathcal{X}^* [38, 43] (applied variously to POS learning by Haghghi and Klein [18] and word alignment by Dyer et al. [14]) and noise contrastive estimation [30].

Directed models. The directed alternative avoids the global partition function by factorizing the joint distribution in terms of locally normalized conditional probabilities, which are parameterized in terms of features. For unsupervised sequence labeling, the model was called a “feature HMM” by Berg-Kirkpatrick et al. [3]. The local emission probabilities $p(x_i | y_i)$ in a first-order HMM for POS tagging are reparameterized as follows (again, using notation close to ours):

$$p_{\boldsymbol{\lambda}}(x_i | y_i) = \frac{\exp \boldsymbol{\lambda}^\top \mathbf{g}(x_i, y_i)}{\sum_{x \in \mathcal{X}} \exp \boldsymbol{\lambda}^\top \mathbf{g}(x, y_i)} \quad (5)$$

The features relating hidden to observed variables must be local within the factors implied by the directed graph. We show below that this locality restriction excludes features that are useful (§A.1).

Put in these terms, the proposed autoencoding model is a hybrid directed-undirected model.

Asymptotic Runtime Complexity of Inference. The models just described cannot condition on arbitrary amounts of \mathbf{x} without increasing inference costs. Despite the strong independence assumptions of those models, the computational complexity of inference required for learning with CRF autoencoders is better (§2.1).

Consider learning the parameters of an undirected model by maximizing likelihood of the observed data. Computing the gradient for a training instance \mathbf{x} requires time

$$O(|\boldsymbol{\lambda}| + |\mathcal{T}| \times |\mathbf{x}| \times |\mathcal{Y}| \times (|\mathcal{Y}| \times |F_{y_{i-1}, y_i}| + |\mathcal{X}| \times |F_{x_i, y_i}|)),$$

where F_{x_i, y_i} are the emission-like features used in an arbitrary assignment of x_i and y_i . When the multiplicative factor $|\mathcal{X}|$ is large, inference is slow compared to CRF autoencoders.

Inference in *directed models* is faster than in undirected models, but still slower than CRF autoencoder models. In directed models [3], each iteration requires time

$$O(|\boldsymbol{\lambda}| + |\mathcal{T}| \times |\mathbf{x}| \times |\mathcal{Y}| \times (|\mathcal{Y}| \times |F_{y_{i-1}, y_i}| + |F_{x_i, y_i}|) + |\boldsymbol{\theta}'| \times \max(|F_{y_{i-1}, y_i}|, |F_{\mathcal{X}, y_i}|)),$$

where F_{x_i, y_i} are the active emission features used in an arbitrary assignment of x_i and y_i , $F_{\mathcal{X}, y_i}$ is the union of all emission features used with an arbitrary assignment of y_i , and $\boldsymbol{\theta}'$ are the local emission and transition probabilities. When $|\mathcal{X}|$ is large, the last term $|\boldsymbol{\theta}'| \times \max(|F_{y_{i-1}, y_i}|, |F_{\mathcal{X}, y_i}|)$ can be prohibitively large.

3.2 Other Related Work

The proposed CRF autoencoder is more distantly related to several important ideas in less-than-supervised learning.

Autoencoders and other “predict self” methods. Our framework borrows its general structure, Fig. 2 (left), as well as its name, from *neural network* autoencoders. The goal of neural autoencoders has been to learn feature representations that improve generalization in otherwise supervised learning problems [44, 8, 39]. In contrast, the goal of CRF autoencoders is to learn specific *interpretable* regularities of interest.⁷ It is not clear how neural autoencoders could be used to learn the latent structures that CRF autoencoders learn, without providing supervised training examples. Stoyanov et al. [40] presented a related approach for discriminative graphical model learning, including features and latent variables, based on backpropagation, which could be used to instantiate the CRF autoencoder.

Daumé III [9] introduced a reduction of an unsupervised problem instance to a series of single-variable supervised classifications. The first series of these construct a latent structure \mathbf{y} given the entire \mathbf{x} , then the second series reconstruct the input. The approach can make use of any supervised learner; if feature-based probabilistic models were used, a $|\mathcal{X}|$ summation (akin to Eq. 5) would be required. On unsupervised POS induction, this approach performed on par with the undirected model of Smith and Eisner [38].

Minka [29] proposed cascading a generative model and a discriminative model, where class labels (to be predicted at test time) are marginalized out in the generative part first, and then (re)generated in the discriminative part. In CRF autoencoders, observations (available at test time) are conditioned on in the discriminative part first, and then (re)generated in the generative part.

Posterior regularization. Introduced by Ganchev et al. [16], posterior regularization is an effective method for specifying constraint on the posterior distributions of the latent variables of interest; a similar idea was proposed independently by Bellare et al. [2]. For example, in POS induction, every sentence might be expected to contain at least one verb. This is imposed as a soft constraint, i.e., a feature whose expected value under the model’s posterior is constrained. Such expectation constraints are specified directly by the domain-aware model designer.⁸ The approach was applied to unsupervised POS induction, word alignment, and parsing. Although posterior regularization was applied to directed feature-less generative models, the idea is orthogonal to the model family and can be used to add more inductive bias for training CRF autoencoder models.

4 Evaluation

We evaluate the effectiveness of CRF autoencoders for learning from unlabeled examples in POS induction and word alignment. We defer the detailed experimental setup to Appendix A.

Part-of-Speech Induction Results. Fig. 3 compares predictions of the CRF autoencoder model in seven languages to those of a featurized first-order HMM model [3] and a standard (feature-less) first-order HMM, using V-measure [37] (higher is better). First, note the large gap between both feature-rich models on the one hand, and the feature-less HMM model on the other hand. Second, note that CRF autoencoders outperform featurized HMMs in all languages, except Italian, with an average relative improvement of 12%.

These results provide empirical evidence that feature engineering is an important source of inductive bias for unsupervised structured prediction problems. In particular, we found that using Brown cluster reconstructions and specifying features which span multiple words significantly improve the performance. Refer to Appendix A for more analysis.

Bitext Word Alignment Results. First, we consider an intrinsic evaluation on a Czech-English dataset of manual alignments, measuring the alignment error rate (AER; [32]). We also perform an

⁷This is possible in CRF autoencoders due to the interdependencies among variables in the hidden structure and the manually specified feature templates which capture the relationship between observations and their hidden structures.

⁸In a semi-supervised setting, when some labeled examples of the hidden structure are available, Druck and McCallum [11] used labeled examples to estimate desirable expected values. We leave semi-supervised applications of CRF autoencoders to future work; see also Suzuki and Isozaki [41].

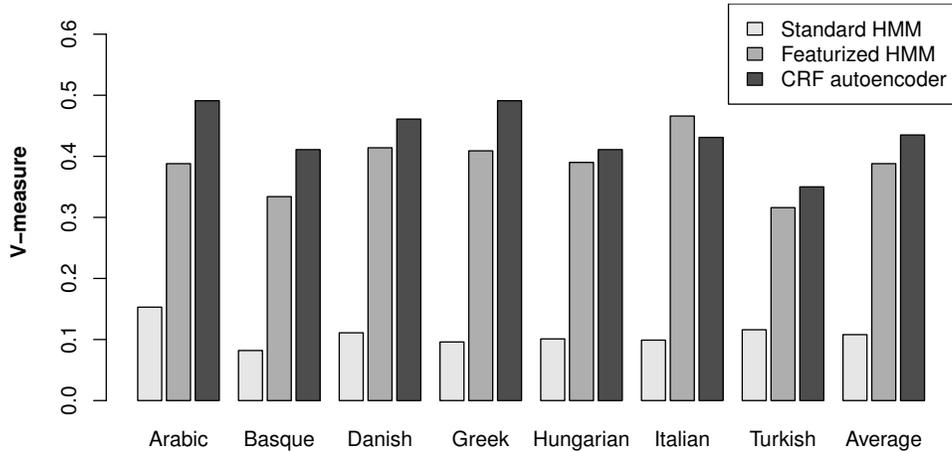


Figure 3: V-measure [37] of induced parts of speech in seven languages. The CRF autoencoder with features spanning multiple words and with Brown cluster reconstructions achieves the best results in all languages but Italian, closely followed by the feature-rich HMM of Berg-Kirkpatrick et al. [3]. The standard multinomial HMM consistently ranks last.

direction	fast_align	model 4	auto	pair	fast_align	model 4	auto
forward	27.7	31.5	27.5	cs-en	15.2±0.3	15.3±0.1	15.5±0.1
reverse	25.9	24.1	21.1	ur-en	20.0±0.6	20.1±0.6	20.8±0.5
symmetric	25.2	22.2	19.5	zh-en	56.9±1.6	56.7±1.6	56.1±1.7

Table 1: Left: AER results (%) for Czech-English word alignment. Lower values are better. . Right: Bleu translation quality scores (%) for Czech-English, Urdu-English and Chinese-English. Higher values are better. .

extrinsic evaluation of translation quality in three language pairs, using case-insensitive Bleu [33] of a machine translation system (cdec⁹ [13]) built using the word alignment predictions of each model.

AER for variants of each model (forward, reverse, and symmetrized) are shown in Table 1 (left). Our model significantly outperforms both baselines. Bleu scores on the three language pairs are shown in Table 1; alignments obtained with our CRF autoencoder model improve translation quality of the Czech-English and Urdu-English translation systems, but not of Chinese-English. This is unsurprising, given that Chinese orthography does not use letters, so that source-language spelling and morphology features our model incorporates introduce only noise here. Better feature engineering, or more data, is called for.

We have argued that the feature-rich CRF autoencoder will scale better than its feature-rich alternatives. Fig. 5 (in Appendix A.2) shows the average per-sentence inference runtime for the CRF autoencoder compared to exact inference in an MRF [14] with a similar feature set, as a function of the number of sentences in the corpus. For CRF autoencoders, the average inference runtime grows slightly due to the increased number of parameters, while it grows substantially with vocabulary size in MRF models [14].¹⁰

5 Conclusion

We have presented a general and scalable framework to learn from unlabeled examples for structured prediction. The technique allows features with global scope in observed variables with favorable asymptotic inference runtime. We achieve this by embedding a CRF as the encoding model in the

⁹<http://www.cdec-decoder.org/>

¹⁰We only compare runtime, instead of alignment quality, because retraining the MRF model with exact inference was too expensive.

input layer of an autoencoder, and reconstructing a transformation of the input at the output layer using simple categorical distributions. The key advantages of the proposed model are scalability and modeling flexibility. We applied the model to POS induction and bitext word alignment, obtaining results that are competitive with the state of the art on both tasks.

Acknowledgments

We thank Brendan O’Connor, Dani Yogatama, Jeffrey Flanigan, Manaal Faruqui, Nathan Schneider, Phil Blunsom and the anonymous reviewers for helpful suggestions. We also thank Taylor Berg-Kirkpatrick for providing his implementation of the POS induction baseline, and Phil Blunsom for sharing POS induction evaluation scripts. This work was sponsored by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF-10-1-0533. The statements made herein are solely the responsibility of the authors.

References

- [1] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine learning*, 1995.
- [2] K. Bellare, G. Druck, and A. McCallum. Alternating projections for learning with expectation constraints. In *Proc. of UAI*, 2009.
- [3] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. Painless unsupervised learning with features. In *Proc. of NAACL*, 2010.
- [4] P. Blunsom and T. Cohn. Discriminative word alignment with conditional random fields. In *Proc. of Proceedings of ACL*, 2006.
- [5] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 1992.
- [6] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. In *Computational Linguistics*, 1993.
- [7] S. Buchholz and E. Marsi. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL-X*, 2006.
- [8] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*, 2008.
- [9] H. Daumé III. Unsupervised search-based structured prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 209–216. ACM, 2009.
- [10] A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.
- [11] G. Druck and A. McCallum. High-performance semi-supervised learning using discriminatively constrained generative models. In *Proc. of ICML*, 2010.
- [12] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.
- [13] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proc. of ACL*, 2010.
- [14] C. Dyer, J. Clark, A. Lavie, and N. A. Smith. Unsupervised word alignment with arbitrary features. In *Proc. of ACL-HLT*, 2011.
- [15] C. Dyer, V. Chahuneau, and N. A. Smith. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. of NAACL*, 2013.
- [16] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049, 2010.
- [17] Q. Gao and S. Vogel. Parallel implementations of word alignment tool. In *In Proc. of the ACL workshop*, 2008.
- [18] A. Haghighi and D. Klein. Prototype-driven learning for sequence models. In *Proc. of NAACL-HLT*, 2006.
- [19] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

- [20] M. Johnson. Why doesn't EM find good HMM POS-taggers? In *Proc. of EMNLP*, 2007.
- [21] D. Klein and C. D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. of ACL*, 2004.
- [22] P. Koehn. *Statistical Machine Translation*. Cambridge, 2010.
- [23] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of NAACL*, 2003.
- [24] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, 2001.
- [25] P. Liang. Semi-supervised learning for natural language. In *Thesis, MIT*, 2005.
- [26] C.-C. Lin, W. Ammar, C. Dyer, and L. Levin. The cmu submission for the shared task on language identification in code-switched data. In *First Workshop on Computational Approaches to Code Switching at EMNLP*, 2014.
- [27] A. V. Lukashin and M. Borodovsky. Genemark. hmm: new solutions for gene finding. *Nucleic acids research*, 26(4):1107–1115, 1998.
- [28] B. Merialdo. Tagging english text with a probabilistic model. In *Comp. Ling.*, 1994.
- [29] T. Minka. Discriminative models, not discriminative training. Technical report, Technical Report MSR-TR-2005-144, Microsoft Research, 2005.
- [30] A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proc. of ICML*, 2012.
- [31] J. Nivre, J. Hall, S. Kubler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proc. of CoNLL*, 2007.
- [32] F. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 2003.
- [33] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, 2002.
- [34] S. Petrov, D. Das, and R. McDonald. A universal part-of-speech tagset. In *Proc. of LREC*, May 2012.
- [35] H. Poon and P. Domingos. Joint unsupervised coreference resolution with Markov logic. In *Proc. of EMNLP*, 2008.
- [36] S. Reddy and S. Waxmonsky. Substring-based transliteration with conditional random fields. In *Proc. of the Named Entities Workshop*, 2009.
- [37] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007.
- [38] N. A. Smith and J. Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proc. of ACL*, 2005.
- [39] R. Socher, C. D. Manning, and A. Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *NIPS workshop*, 2010.
- [40] V. Stoyanov, A. Ropson, and J. Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *Proc. of AISTATS*, 2011.
- [41] J. Suzuki and H. Isozaki. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proc. of ACL*, 2008.
- [42] R. Swier and S. Stevenson. Unsupervised semantic role labelling. In *Proc. of EMNLP*, 2004.
- [43] D. Vickrey, C. C. Lin, and D. Koller. Non-local contrastive objectives. In *Proc. of ICML*, 2010.
- [44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of ICML*, 2008.
- [45] S. Vogel, H. Ney, and C. Tillmann. Hmm-based word alignment in statistical translation. In *Proc. of COLING*, 1996.
- [46] M. Weber, M. Welling, and P. Perona. *Unsupervised learning of models for recognition*. 2000.
- [47] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Proc. of CVPR*, pages 379–385. IEEE, 1992.

A Experimental Details

In this section, we describe our experimental setup for part of speech induction and bitext word alignment in some detail.

A.1 Part-of-Speech Induction Experimental Setup

The first task, part-of-speech (POS) induction, is a classic NLP problem which aims at discovering syntactic classes of tokens in a monolingual corpus, with a predefined number of classes. An example of a POS-tagged English sentence is in Fig. 1.

Data. We use the plain text from CoNLL-X [7] and CoNLL 2007 [31] training data in seven languages to train the models: Arabic, Basque, Danish, Greek, Hungarian, Italian and Turkish. For evaluation, we obtain gold-standard POS tags by deterministically mapping the language-specific POS tags from the shared task training data to the corresponding universal POS tag set [34].¹¹

Setup. We configure our model (as well as baseline models) to induce $|\mathcal{Y}| = 12$ classes. We use zero initialization of the CRF parameters, and initialize the reconstruction model parameters with a basic first-order HMM. In each block-coordinate ascent iteration, we run one L-BFGS iteration (including a line search) to optimize λ , followed by one EM iteration to optimize θ . Training converges after 70 block-coordinate ascent iterations.

Evaluation. Since we do not use a tagging dictionary, the word classes induced by our model are not identifiable. We use two cluster evaluation metrics commonly used for POS induction: (i) many-to-one [20], which infers a mapping across the syntactic clusters in the gold vs. predicted labels (higher is better); and (ii) V-measure [37], an entropy-based metric which explicitly measures the homogeneity and completeness of predicted clusters (again, higher is better).

CRF Autoencoder Model Instantiation. Table 1 (right) describes the symbols and variables we use in context of the POS induction problem. We use a first-order linear CRF for the encoding part with the following feature templates:

- $\langle y_i, y_{i-1} \rangle, \forall i$
- $\langle y_i, \text{sub}_j(x_i) \rangle, \forall i, j$
- $\langle y_i, \text{sub}_j(x_i), \text{sub}_k(x_{i-1}) \rangle, \forall i, j, k$
- $\langle y_i, \text{sub}_j(x_i), \text{sub}_k(x_{i+1}) \rangle, \forall i, j, k$

Where $\text{sub}_j(x_i)$ is one of the following sub-word-level feature percepts:

- Prefixes and suffixes of lengths two and three, iff the affix appears in more than 0.02% of all word types,
- Whether the word contains a digit,
- Whether the word contains a hyphen,
- Whether the word starts with a capital letter,
- Word shape features which map sequences of the same character classes into a single character (e.g., ‘McDonalds’ \rightarrow ‘AaAa’, ‘-0.5’ \rightarrow ‘#0#0’),
- The lowercased word, iff it appears more than 100 times in the corpus.

In the reconstruction model, we generate the Brown cluster of a word [5],¹² conditioned on the POS tag using a categorical (i.e., multinomial) distribution. No side information is used in this model instantiation. Predictions are the best value of the latent structure according to the posterior $p(\mathbf{y} | \mathbf{x}, \hat{\mathbf{x}}, \phi)$.

¹¹<http://code.google.com/p/universal-pos-tags/>

¹²We use `brown-cluster -c 100 v1.3`, available at <https://github.com/percyliang/brown-cluster> [25] with data from <http://corpora.informatik.uni-leipzig.de/>.

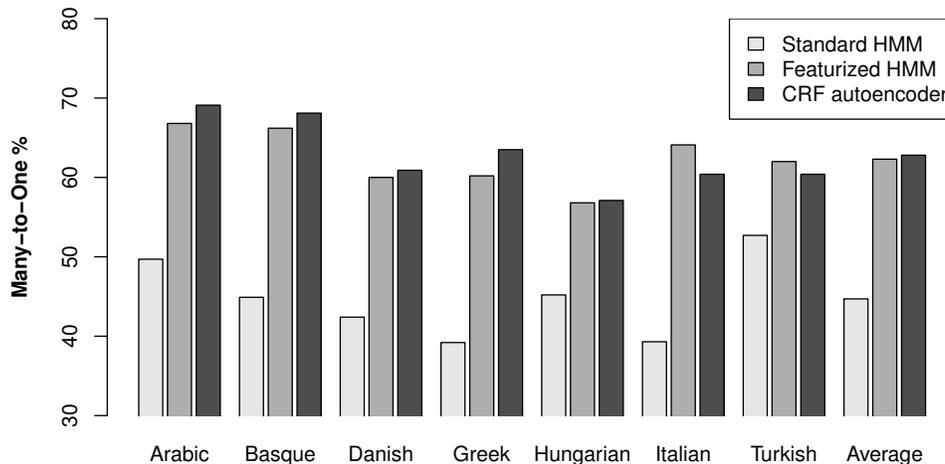


Figure 4: Many-to-one accuracy of POS induction.

Baselines. We consider two baselines:

- `hmm`: a standard first-order hidden Markov model learned with EM;¹³
- `fhmm`: the directed alternative discussed in the main paper, as implemented by Berg-Kirkpatrick et al. [3], with the feature set from Smith and Eisner [38].

Tuning Hyperparameters. In our models, we use a squared L_2 regularizer for CRF parameters λ , and a symmetric Dirichlet prior for categorical parameters θ with the same regularization strength for all languages. The `fhmm` baseline also uses a squared L_2 regularizer for the log-linear parameters. The hyperparameters of our model, as well as baseline models, were tuned to maximize many-to-one accuracy for The English PennTreebank. The `fhmm` model uses L_2 strength = 0.3. The auto model uses L_2 strength = 2.5, $\alpha = 0.1$.

A.2 Bitext Word Alignment Experimental Setup

Word alignment is an essential step in the training pipeline of most statistical machine translation systems [22]. Given a sentence in the source language and its translation in the target language, the task is to find which source token, if any, corresponds to each token in the target translation. We make the popular assumption that each token in the target sentence corresponds to zero or one tokens in the source sentence. Fig. 1 shows a Spanish sentence and its English translation with word alignments. As shown in Table 1 (right), an observation \mathbf{x} consists of tokens in the target sentence, while side information ϕ are tokens in the source sentence. Conditioned on a source word, we use a categorical distribution to generate the corresponding target word according to the inferred alignments.

Data. We consider three language-pairs: Czech-English, Urdu-English, and Chinese-English. For Czech-English, we use 4.3M bitext tokens for training from the NewsCommentary corpus, WMT10 data set for development, and WMT11 for testing. For Urdu-English, we use the train (2.4M bitext tokens), development, and test sets provided for NIST open MT evaluations 2009. For Chinese-English, we use the BTEC train (0.7M bitext tokens), development, and test sets (travel domain).

CRF Autoencoder Model Instantiation. For word alignment, we define the reconstruction model as follows: $p_{\theta}(\hat{\mathbf{x}} | \mathbf{y}, \phi) = \prod_{i=1}^{|\mathbf{x}|} \theta_{\hat{x}_i | \phi_{y_i}}$, where \hat{x}_i is the Brown cluster¹⁴ of the word at position i in the target sentence. We use a squared L_2 regularizer for the log-linear parameters λ and a

¹³Among 32 Gaussian initializations of model parameters, we use the HMM model which gives the highest likelihood after 30 EM iterations.

¹⁴We again use [25] with 80 word classes.

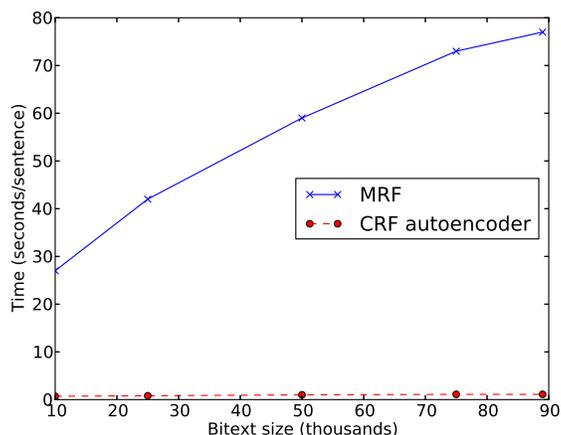


Figure 5: Average inference runtime per sentence pair for word alignment in seconds (y -axis), as a function of the number of sentences used for training (x -axis).

symmetric Dirichlet prior for the categorical parameters θ with the same regularization strength for all language pairs (L_2 strength = 0.01, Dirichlet $\alpha = 1.5$). The hyperparameters were optimized to minimize alignment error rate (AER) on a development dataset of French-English bitext. The reconstruction model parameters θ are initialized with the parameters taken from IBM Model 1 after five EM iterations [6]. In each block-coordinate ascent iteration, we use L-BFGS to optimize λ , followed by two EM iterations to optimize θ . Training converges when the relative improvement in objective value falls below 0.03 in one block-coordinate ascent iteration, typically in less than 10 iterations of block-coordinate ascent.

We follow the common practice of training two word alignment models for each dataset, one with English as the target language (forward) and another with English as the source language (reverse). We then use the grow-diag-final-and heuristic [23] to symmetrize alignments before extracting translation rules.

Features. We use the following features: deviation from diagonal word alignment $|\frac{y_i}{|\phi|} - \frac{i}{|x|}|$; log alignment jump $\log |y_i - y_{i-1}|$; agreement with forward, reverse and symmetrized baseline alignments of `mgiza++` and `fast_align`; Dice measure of the word pair x_i and ϕ_{y_i} ; difference in character length between x_i and ϕ_{y_i} ; orthographic similarity between x_i and ϕ_{y_i} ; punctuation token aligned to a non-punctuation token; punctuation token aligned to an identical token; 4-bit prefix of the Brown cluster of x_i conjoined with 4-bit prefix of the Brown cluster of ϕ_{y_i} ; forward and reverse probability of the word pair x_i, ϕ_{y_i} with `fast_align`, as well as their product. The outputs of other unsupervised aligners are standard (and important!) features in supervised CRF aligners [4]; however, they are nonsensical in a joint model over alignments and sentence pairs.

Baselines. Due to the cost of estimating feature-rich generative models for unsupervised word alignment on the data sizes we are using (e.g., `fmm` and `dyer-11`), we only report the per-sentence computational cost of inference on these baselines. For alignment quality baselines, we report on results from two state-of-the-art baselines that use multinomial parameterizations which support M-step analytic solutions, rather than feature-rich parameterizations: `fast_align` [15]¹⁵ and `model 4` [6]. `fast_align` is a recently proposed reparameterization of IBM Model 2 [6]. `model 4`, as implemented in `mgiza++` [17] is the most commonly used word alignment tool in state-of-the-art machine translation systems.

Evaluation. When gold standard word alignments are available (i.e., for Czech-English), we use AER [32] to evaluate the alignment predictions of each model. We also perform an extrinsic evaluation of translation quality for all data sets, using case-insensitive Bleu [33] of a hierarchical MT system built using the word alignment predictions of each model.

¹⁵<https://github.com/clab/fastalign>