Bilingual Part of Speech Tag Induction with Markov Random Fields

Desai Chen
Carnegie Mellon University
Pittsburgh, PA 15289, USA
desaic@andrew.cmu.edu

Chris Dyer **Noah Smith**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh , PA 15217,USA
{cdyer, nasmith}@cs.cmu.edu

Abstract

Unsupervised multilingual learning has been shown to be effective for NLP(natural language processing) tasks such as POS(part of speech) tag induction [Snyder and Barzilay (2010)] and grammar induction [Cohen and Smith (2009)] [Berg-Kirkpatrick and Klein (2010)]. This thesis follows the work of Ben Snyder that improves unsupervised part of speech tags with the help of word alignments. Word alignments are useful for part of speech tag induction because there are very regular patterns for the tags of aligned words. They are also very easy to obtain because there exists many dataset with parallel sentences and automatic word aligners work quite well. Another focus of the thesis is that the model we use is an undirected Markov random field. The behaviour and capacity of such models are not well-understood. I will discuss some performance issues with Markov random fields.

1 Introduction

The performance of unsupervised POS tag induction systems is significantly worse than supervised systems. A lot of research has been done to improve unsupervised techniques. The advantage of unsupervised methods is that it's cheap to obtain a large amount of unannotated data when annotated data is not available. Multilingual learning has recently been shown to be effective for these unsupervised NLP tasks. Our work will follow the baseline established by Ben Snyder et al.[Snyder and Barzilay (2010)]. The system will take a dataset of parallel sentences with word alignments as input, and then output POS tagging. Word alignments are links drawn between words of two languages that translate to each other. The training and inference procedure of Markov random fields is an interesting application of approximate inference techniques.

2 Related work

The idea of using multilingual data to improve unsupervised systems is a hot topic in NLP research [Snyder and Barzilay (2010)] [Cohen and Smith (2009)] [Berg-Kirkpatrick and Klein (2010)]. The most relevant work for my task is the Bayesian (Hidden Markov Model) HMM proposed by Ben Snyder in [Snyder et al. (2008)]. Relevant work on word alignment can be found in [DeNero et al. (2008)].

The model most relevant to my work is the one presented in Ben Snyder's work. The part of speech induction model can be represented as a directed graphical model as in Figure 1. Suppose we have two sentences **s** and **t** that translates to each other. Call one of them the source sentence and the other the target sentence. Let N_s and N_t be the lengths of the source sentence and the target sentence. Let x_i be the POS tag for the ith word in the source sentence and y_i be the POS tag for the ith word in the target sentence. Given a monotonic alignment a (alignment with no crossing edges), the joint probability of a tagging $(x_1, ..., x_{N_s}), (y_1, ..., y_{N_t})$ and the sentences is

$$P(x_1, ..., x_{N_s}, y_1, ..., y_{N_t}, \mathbf{s}, \mathbf{t})$$

$$= \prod_{(i,j)\in a} P(x_i, y_j | x_{i-1}, y_{j-1}) P(\mathbf{s}_i|x_i) P(\mathbf{t}_j|y_j) \cdot$$

$$\prod_{\text{unaligned } i} P(x_i|x_{i-1}) P(\mathbf{s}_i|x_i) \cdot$$

$$\prod_{\text{unaligned } j} P(y_j|y_{j-1}) P(\mathbf{t}_j|y_j).$$

That is , the tag of a word is generated by the tag of the previous word. The Bayesian HMM based on the above model has additional coupling parameters that represent the probability of a tag in the source language being aligned to another tag in the target language. In order to guide the unsupervised learning process, there are Dirichlet priors over the parameters as commonly done with Bayesian models. The inference procedure is a Gibbs sampling based approach.

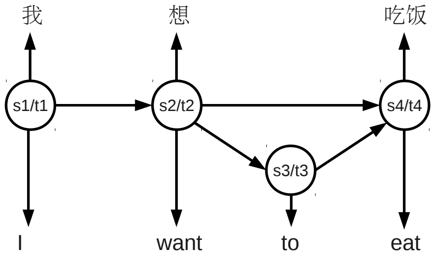

Figure 1. Bilingual HMM-like POS induction model

3 Markov Random Field

The model I'm using for solving the problem is a Markov random fields that builds off the alignment structure of parallel sentences. An example of such a graphical model is shown in figure 2.

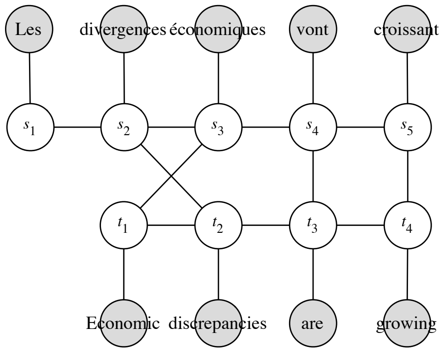

Figure 2. Undirected graphical model that permits crossing edges in word alignment.

In the Figure 2, each node represents a random variable and each link between the nodes represent the relations between the variables. Nodes shaded grey are observed random variables. In this case, the words are observed random variables. The rest of nodes are unobserved and we are interested in figuring out their best values.

Define a feature function $\mathbf{f}(n_1, n_2) = (f_1(n_1, n_2), f_2(n_1, n_2), ...)$ that computes a feature vector based on the configuration of the nodes n_1 and n_2. Define a weight vector $\mathbf{w} = (w_1, w_2, ...)$ where w_i is the weight for feature function f_i.

The unnormalized probability of a configuration of all the nodes $\mathbf{N}_i$ is defined as the product

$$p(\mathbf{N}_i) = \prod_{j,k} \exp(\mathbf{w} \cdot \mathbf{f}(n_j, n_k)). \qquad (1)$$

The normalized probability is obtained by normalizing over all possible configurations

$$P(\mathbf{N}_i) = \frac{p(\mathbf{N}_i)}{\sum_j p(\mathbf{N}_j)}.$$

In theory one can define an arbitrary way of computing the probabilities for each edge. Exponentiating the linear combination of feature values is preferred because the gradient of the weights with respect to the marginal probability of the observed variables work out to have very simple form. Let $\mathbf{X}_i$ be a configuration of the observed variables , then the marginal probability of $\mathbf{X}_i$ is given by

$$P(\mathbf{X}_i) = \frac{\sum_j P(\mathbf{N}_j, \mathbf{X}_i)}{\sum_{j,k} P(\mathbf{N}_j, \mathbf{X}_k)}.$$

The gradient with respect to a weight w_j is then

$$\frac{\partial P(\mathbf{X}_i)}{\partial w_j} = \mathbf{E}[f_j | \mathbf{X}_i] - \mathbf{E}[f_j].$$

That is, the gradient for w_j is the expected value of feature f_j conditioned on $\mathbf{X}_i$ minus the expected value of f_j over all possible configuration of random variables.

We decided to use Gibbs sampling for training and inference. Previous work on linear chain structured MRF [Haghighi and Klein (2006)] has used a more exact inference algorithm. In our case, since the structure is not simple, dynamic programming tricks don't apply. We are also interested in seeing how approximate inference algorithm would perform.

3.1 Parameter Estimation

The parameter estimation procedure is a combination of Gibbs sampling and contrastive estimation. To estimate the weights $\mathbf{w}$, the training algorithm iterates between sampling part of speech tags sampling permutations of words to compute the expected value of features. The first term of the gradient in Equation(1) is the expected values of feature functions conditioned on the observed variables . This can be computed by keeping the words fixed and sampling tags many times. The second term of the gradient in Equation(1) is the expected values averaged over all possible word sequences. In practice, we found that sampling all possible word sequences didn't work well. We used the idea of contrastive estimation and limited the sampler to only sample permutations of the words in the sentence. Gibbs sampling procedure is very straightforward: To sample a tag , keep all other tags and words fixed, and then compute the probability of choosing a tag conditioned on the adjacent nodes including the word, the preceding and succeeding tags and possibly aligned tags in the other language. A random tag is picked according to the conditional probability. A similar procedure works for words. At each step, the sampler decides whether to swap two adjacent words or not based while keeping other tags and words fixed. In general, we can compute the distribution of a node n conditioned on its neighbors $\mathbf{N}$. We first define the unnormalized probability for state n_i as

$$p(n_i) = \prod_{n_j \in \mathbf{N}} \exp(\mathbf{w} \cdot \mathbf{f}(n_i, n_j)).$$

Then the probability for picking n_i is obtained by normalizing among the $p(n_i)$s.

$$P(n_i) = \frac{p(n_i)}{\sum_j p(n_j)}.$$

In the current implementation, since we only have binary valued features, to get the expectation, just count how many times each feature function f_i takes value 1 and then divide each of the counts by the number of times we sample.

In the original Bayesian HMM , the author makes lots of assumptions and heuristics to get a clean and simplified word alignment where the alignment is at most one-to-one and has no crossing links. We hope to improve over the constraints imposed on the alignment and get more information from crossing links. For example, Figure 2 shows a crossing link between “Economic discrepancies ” and “divergences economiques ”. This crossing link is indicative about the relation between “Economic” and “economiques”.

4 Results and Analysis

I compared our model to the Bayesian HMM. The data set that the author reports on is translations of the novel 1984 in English, Bulgarian, Slovene and Serbian. The data set is manually annotated with part of speech tags. The word alignments are generated using programs. The data shows very regular patterns of tags that are aligned together. Words with the same tag in two languages tend to be aligned with each other.

language pair	Bayesian HMM	MRF1	MRF 2
bg	94.48	93.3	90.5
en	92.0	91.6	91.3
sr	91.8	88.1	91.8
sl	95.1	87.7	95.0
en	92.01	91.9	92.7
sl	88.54	87.8	95.0
bg	91.95	93.4	90.7
sr	86.58	88.7	85.0
en	91.01		91.6
sr	90.06		89.2
bg	90.91		90.2
sl	88.20		88.0

Table 1. Unsupervised bilingual results with complete tag dictionary

There are 14 part of speech tags, two of which are punctuations. A complete tag dictionary is provided. That is, each word has only a small number of tags it can possibly use. The baseline of choosing random tags for each word gives an accuracy of around 85% except for English. English has an extended tag dictionary obtained from the Wall Street Journal. The random baseline gives an accuracy of around 55%.

As a very primitive comparison, I trained a supervised MRF model, which is also called conditional random fields(CRF) to compare to the supervised results of HMMs. The training procedure is also sampling based. The only difference is that there is no need to sample the words because the tags are the only random variables. CRF and HMM give very close performance with difference in accuracy less than 0.1%. This shows that the CRF is capable of representing an equivalent model represented by the HMM.

A comparison of unsupervised results between Bayesian HMM and MRF is shown in Table 1.

In Table 1,Bayesian HMM is the results reported by the original author. MRF1 and MRF2 are two runs of my model initialized randomly. Even though the level of ambiguity is low, we can still see oscillations in the range of about 5% in both my model and the Bayesian HMM. The reason as I studied is that there are a few very common words in the data such as “the”, “is” and equivalent words in the other languages. These words are almost always aligned to each other and therefore word alignments are not indicative of the tagging. Labeling these words completely right or completely wrong are both local optima to the model.

In case of MRF, these local optima differ in the weights of only a few features. Flipping the values of those weights during initialization would lead to completely different solutions. The model will be stuck at whatever local optimum it started at. The difference in initialization would eventually lead to significant difference in accuracy. Such effects of initialization for unsupervised models are well known phenomena. For an example, refer to [Johnson (2007)].

The training procedure of the model is tricky to tune. The model is originally trained with stochastic gradient descent with on-line update and a Metropolis Hastings step for sampling the words. It turns out that on-line update almost always guide the model to a local optimum with low accuracy. The effect of on-line update is very hard to study and is not well-understood by research community. Then I switched to gradient descent without on-line update and the behaviour is more regular. I control the step size by limiting the maximum absolute value of partial derivatives. In this task, regularization seems to only hurt the performance. The magnitude of the weights are already limited by the sampling step.

I also compared the results when only a small portion of the tag dictionary is available. This set of result is more interesting because the random tagging baseline is much lower. There is much more for the models to learn compared to the case with complete tag dictionaries. The tag dictionaries only contain the top 100 most frequent words for each language. The results are shown in Table 2.

language pair	HMM	MRF
en bg	71.34 62.55	72.3 60±5
sr sl	54.08 59.68	52±2 62.0
en sl	66.48 53.77	73.0 53±2
bg sr	54.22 56.91	53±2 57.0
en sr	68.22 54.73	71.77 57.20
bg sl	55.88 58.50	58±1 62.9

Table 2. Unsupervised bilingual results with tag dictionary only for the top 100 frequent words

The results are not satisfactory even though they are still comparable to the HMM baseline. The model was much worse when trained with the likelihood objective. We tried using exact inference instead of sampling to optimize the weights and found that the likelihood objective has lots of bad local optima. The model easily gets stuck in those local optima. The bad solutions makes the model use less tags when more tags are available. This behaviour is the opposite of that of a directed model. A directed model tends to use more tags whenever it can.

To make the objective function easier to optimize, we switched to contrastive estimation. The intuition is that word ordering is more important than picking words from the vocabulary for learning syntax of a language. The contrastive objective works surprisingly well compared to the full objective. The weights learned by the model shows that the model is focusing much more on transition features and alignment features rather than emission features. The transition features and alignment features are very powerful for modelling word ordering.

One potential advantage of an undirected model is that it allows arbitrary features. In the case with complete tag dictionaries, I experimented with prefix and suffix but only got worse performance. With more features, the model is more likely to over-fit. Since every word already has a small list of possible tags, prefix and suffix features is not going to help at all. When I switched to using a small portion of the tag dictionary, the performance is very different. A comparison is show in Table 3.

Another potential advantage of MRFs is that they allow crossing links. However, in this particular task, crossing links don’t make a significant difference. The reason is that these languages are all very similar and there are very few crossing

language pair	Basic Feature	Prefix suffix feature
en	72.1	72.3
bg	56.2	60±5
sr	47.2	52±2
sl	52.7	62.0

Table 3. Effect of prefix and suffix features on the 1984 data for two language pairs.

language pair	Random	Basic Feature	Prefix suffix feature
fr	63.6	89.8	
en	72.0	90.4	
de			
en			
cs			
en			

Table 4. MRF results trained on the first 10000 sentences of EUROPARL data and tested on treebank tagged data.

links. They are too few to make a difference. I'm hoping to see a more significant effect with language pairs that have more crossing links. French and English is a promising language pair to look at.

5 Experiments to Come

I have more data to run the model with. We have parallel data from European Parliament in English, French, Czech and German with $50,000$ sentences in each language. It's a more realistic data set than the novel 1984. The code runs in reasonable amount of time. It takes about two hours to train with 4000 sentences. I think I still have time to run my code on that data once the problems are fixed.

I'm also planning on running the model under different setting. Using only 14 doesn't seem that useful. The Penn Tree Bank has about 50 tags. A more fine-grained tag set distinguishes between tags such as verb in present tense or past tense. Training a model with more fine-grained tags is more useful for other tasks such as grammar induction. I trained English with 34 treebank tags. The results are in Table 4.

The model can also be ran under a projection setting where the tags for one of the language pair is observed. All the model needs to do is to figure out tags for the other language. This task is useful because , for example, we have a pretty good tagger and a large amount of data for English part of speech. But such data is not available for other languages. Then it would be nice if English part of speech can induce part of speech tags for other languages.

6 Extensions

The orignal proposal of the thesis is to jointly induce part of speech tagging and word alignment. Given the difficulties in training the model, I haven't completely given up the idea yet, but it's imaginable that the model will be even harder to optimize and understand. The idea is still worth trying even after finishing my thesis.

References

Taylor Berg-Kirkpatrick and Dan Klein. Phylogenetic grammar induction. In *Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics, 2010.

Shay B. Cohen and Noah A. Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL*, 2009.

John DeNero, Alexandre Bouchard-Cote, and Dan Klein. Sampling alignment structure under a bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 320–327, New York City, USA, June 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N06/N06-1041`.

Mark Johnson. Why doesnt em find good hmm pos-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 296–305. EMNLP-CoNLL, 2007.

Benjamin Snyder and Regina Barzilay. Climbing the tower of babel: Unsupervised multilingual learning. In *ICML*, 2010.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Unsupervised multilingual learning for pos tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.