

Information Extraction in Diverse Settings

Sandeepkumar B. Satpal

Roll No. 05329004

Kanwal Rekhi School of Information Technology,
Indian Institute of Technology-Bombay

As a part of the first stage of M.Tech Thesis Project
Under the guidance of **Prof. Sunita Sarawagi**

July 25, 2006

Abstract

Information Extraction (IE) is one of the interesting problems in machine learning. IE employs several statistical models like Maximum Entropy Markov Model (MEMM), Hidden Markov Model (HMM), Conditional Random Fields (CRF). Typically it builds a model which involves learning patterns from the training data, and applying it on test data. In real life applications, training and testing data may come from different domains. Hence a model must learn general patterns, and not specific to a particular domain.

In this project, we propose to design a model, based on Conditional Random Fields in such a manner that a model built using training sequences from a domain can be successfully applied on test sequences from other domain. In this report, we present preliminary work done in this direction.

1 Introduction

In information extraction we identify entities from the given sequence. Its applications include Named Entity Recognition(NER), Part Of Speech tagging (POS), Shallow parsing. In NER, we try to find out the name (label) of each entity in the given sequence. In POS, we tag parts of the text. Shallow parsing means to segment or group the sequence into phrases which are semantically related.

Classification techniques of data mining like Decision tree classifier, Bayesian classifier etc can also be used for information extraction, but they make independent decision for each label. Hence they face label bias problem. Also they work on fixed number of attributes. So the variable length records have to be converted into some fixed length records by some feature selection techniques. Hence they fail to capture the full information of the sequence. On the other hand, statistical models are flexible and can handle variable length records.

Statistical learning models can be broadly classified into generative models like HMM, and conditional models like MEMMs, CRFs and variants of CRF. HMM maximize the joint probability of a data sequence x being generated from a label sequence y . In HMMs long range dependencies are not considered. This provides the motivation for conditional models. These models are explained in detail in Section 4.2.

The report is structured as follows. Section 2 describes the motivation behind the problem. In Section 3 we describe the problem and list the task to be done. In Section 4 we summarize the literature related to our problem. Some probable approaches towards the problem explained in Section 5. Section 6 is our approach towards the problem. Experiments and conclusion are discussed in Section 7.

2 Motivation

Standard CRF generates features from the training data and learns (assign weights to the features) the model from it. Hence there is a heavy dependence on the training sequences. Although lot of training

data is readily available in most domains, some of them may not have enough training sequences to effectively learn the model. In such cases, we would like to train CRF model in a particular domain and use it on another one. This method faces one serious problem: “features” which are not relevant in test data may get higher weights in training data and vice-versa. This may happen because some features may be more relevant in one domain and not so important in the other. For example: feature for “person name” with previous word labeled as “subject” get higher weights in email sequence while it is less relevant to publication data sequence. Hence CRF model built from one domain cannot be directly applied to other domain.

The problem described earlier motivates our project to find methods so that model built on one set of training sequences works well on a different set. The idea is to find those relevant features and develop techniques to re-learn or re-build the model with minimum efforts.

3 Problem Description

From the above discussion it is evident that a model trained on sequences from one domain generally works well when applied to test sequences in the same domain. Our goal is to build a model trained in such a fashion that it performs well on test sequences from varied domains. We discuss different approaches for this problem in Section 5. We wish to evaluate these methods and find out if there exist some methods which yield better results.

4 Literature Survey

Here we present some existing work related to our problem. The section is organized as follows. We first describe sequence labeling and segmentation in brief. We review CRF and see how it is advantageous over other statistical models for information extraction. Semi-CRF, an extension of CRF is explained in subsection 4.4.

4.1 Segmentation and Sequence Labelling

A sequence $X = x_1x_2x_3 \dots x_n$ consist of multiple tokens x_i where $1 \leq i \leq n$. Segment is defined as contiguous tokens of size greater than or equal to one. For example, a *publication record* is a sequence and a *title* in it consisting of multiple tokens forms a segment. The task of finding segments is called **segmentation** and the task of assigning labels to each segment is called **sequence labelling**.

4.2 Conditional Random Field (CRF)

CRF is a framework to build probabilistic models for segmenting and labeling sequence data [1, 2, 3]. CRFs are undirected graphical models which allow specification of a single joint probability distribution over Y , given X .

$$P(y|x) = \frac{1}{Z(x)} e^{WF(y,x)} \tag{1}$$

where $F(y, x) = \sum_i \mathbf{f}(\mathbf{y}, \mathbf{x}, i)$ is a global feature vector for input sequence \mathbf{x} , $Z = \sum_{y'} e^{WF(y',x)}$ is a global normalization factor, W are weights of the features to be estimated and f is the set of global feature functions [3].

$$P(y|x) = \frac{1}{Z(x)} e^{\sum_k w_k f_k(y,x)} \tag{2}$$

CRF is trained by maximizing the log-likelihood of probability of given training set.

$$LL(W) = \sum_l \log P(y_l|x_l, W) \tag{3}$$

CRF works on undirected graphical linear model where each token is represented as a node. Viterbi algorithm is used to calculate the potential of each node. Detailed mathematical formulation is given in [3].

The advantage of CRF over other conditional models is that CRFs do not face label bias problem since Z is a global normalizer and hence $P(y|x)$ is a single probability distribution. While MEMM use a local normalizer [4] and hence it faces label bias problem.

4.3 Maximum Entropy Markov Model

In MEMM, probability of y given x is,

$$P(y|x) = \prod_{i=1}^n P(y_i|y_{i-1}, x_i) \quad (4)$$

and

$$P(y_i|y_{i-1}, x_i) = \frac{1}{Z(x_i)} e^{\sum_k \lambda_k f_k(y_i, y_{i-1}, x_i)} \quad (5)$$

where Z is local normalization factor and f is set of local feature functions at position i .

4.4 Semi-Markov Conditional Random Fields

CRF assigns a label to each token while Semi-CRF which is a variant of CRF assigns a label to a segment s_i (one or more contiguous token) instead of individual tokens x_i in the sequence [5]. Let $s = \langle s_1, s_2, \dots, s_n \rangle$ denote a segmentation of x , where each segment $s_i = \langle t_j, u_j, y_j \rangle$ consists of a start position t_j , end position u_j and a label y_j .

In Semi-CRF, conditional probability is given as,

$$P(s|x) = \frac{1}{Z(x)} e^{WG(s,x)} \quad (6)$$

$$= \frac{1}{Z(x)} e^{\sum_k w_k g_k(s,x)} \quad (7)$$

where, $G(x, s) = \sum_j^{|s|} g(j, x, s)$, is a global segment feature vector for input sequence x , $Z = \sum_{y'} e^{WG(y', x)}$ is global normalization factor, and g is the set of global segment feature functions [5].

5 Probable Approaches

In this section, we discuss some approaches towards the problem described in Section 3. Each approach has some advantages and disadvantages over other approaches.

5.1 Semi-supervised CRF

Traditional CRF needs labeled data to build a model. However, obtaining labeled instances is often expensive, time consuming and difficult while unlabeled instances are obtained easily. Semi-supervised techniques address this problem by using a large amount of unlabeled data, together with labeled data to build a classifier. Different techniques based on semi-supervised CRF are explained below.

5.1.1 Semi-supervised CRF for Improved Sequence Segmentation and Labeling

The standard supervised CRF training procedure is based upon maximizing the log likelihood of the labeled examples $D^l = ((X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N))$. Let $D^u = ((X_{N+1}, Y_{N+1}), (X_{N+2}, Y_{N+2}), \dots, (X_M, Y_M))$ be the unlabeled data.

$$LL(W) = \sum_{i=1}^N \log p_w(y_i|x_i) - U(W) \quad (8)$$

where $U(W)$ is an standard regularizer on W .

The overlap between the probability distribution over a label sequence y and the empirical distribution of $\tilde{p}(x)$ on the unlabeled data D^u can be measured by KL divergence $KL(p_w(y|x)\tilde{p}(x)||\tilde{p}(x))$ [6]. As the overlap decreases KL divergence increases. Hence to consider the effect of unlabeled data, we need to minimize KL divergence. Minimizing KL divergence implies maximizing the overlap between two distributions. The total overlap over all possible label sequences can be defined by

$$\sum_y D(p_w(y|x)\tilde{p}(x)||\tilde{p}(x)) = \sum_{x \in D^u} \tilde{p}(x) \sum_y p_w(y|x) \log p_w(y|x) \quad (9)$$

Hence the new likelihood for semi-supervised CRF proposed by [6] is given by

$$RL(W) = \sum_{i=1}^N \log p_w(y_i|x_i) - U(W) \quad (10)$$

$$+ \gamma \sum_{i=N+1}^M \sum_y p_w(y|x_i) \log p_w(y|x_i) \quad (11)$$

Where, γ is a tradeoff parameter that controls the influence of the unlabeled data. For detailed mathematical expressions please refer [6].

5.1.2 Semi-supervised Learning using Markov Random Field

[7] characterizes situations in which test data hurts accuracy when the natural clustering of the data is not in good correspondence with the class label. This paper claims that a good distance metric is a key requirement for the success in semi-supervised learning.

Let D^l and D^u be the labeled and unlabeled data described above. Let X_l and X_u be the data sequences from labeled and unlabeled set respectively, and Y_l and Y_U (unknown), the labels for X_l and X_U respectively.

Equation 3 explained above, depends only on the labeled data. Hence unlabeled data does not improve the parameter estimation as shown in [7]. For unlabeled data to make a difference in conditional-likelihood training, we must allow the unlabeled training data X_u to have some impact on labels Y_l . This is accomplished by introducing dependencies between X_u and Y_l .

[7] suggested one way, by adding additional features and weights on pairs of examples — (x_i, y_i) and (x_j, y_j) across all pairs (including both labeled data, and unlabeled data with hypothesized labels). The new data-likelihood model is

$$P(y_l, y_u|x_l, x_u) = \frac{1}{Z(x)} e^{(\sum_i \sum_k w_k f_k(x_i, y_i) + \sum_{i < k} \sum_{k'} w_{k'} f_{k'}(x_i, x_j, y_i, y_j))} \quad (12)$$

where, $Z(x) = \sum_{y'} e^{(\sum_i \sum_k w_k f_k(x_i, y'_i) + \sum_{i < k} \sum_{k'} w_{k'} f_{k'}(x_i, x_j, y'_i, y'_j))}$ is the new normalization function.

The expression is simplified by ignoring the individual values of y_i and y_j , and only determining whether or not two classes are equal. Let $y_{ij} = 1$ iff $y_i = y_j$, hence $f_{k'}(x_i, x_j, y_i, y_j) = f_{k'}(x_i, x_j, y_{ij})$. Detailed log-likelihood formulation is explained in [7].

5.1.3 Learning on Test Data: Leveraging “Unseen” Features

[8] addresses the problem in which training and testing data comes from different distribution. It proposes a method to identify and utilize unseen features. The approach is to associate with each unseen feature a hidden variable (like “meta-features”) that encodes the influence this feature has on its class label. The value of that variable is unknown, and must be “learned” from the test data, without knowledge of the labels in the test data. For example, the phrase “XXX said today“ appears in training data and “XXX”

is a “person name”. Then “XXX” may be a different name in test data but due to meta features this entity may be labeled as “person name”.

Assume the data instances are sampled from some set of scopes, each of which is associated with some data distribution. Also assume that the different distributions share a probabilistic model for some set of global features, but can contain a different probabilistic model for a scope-specific set of local features. Let X denote global features, Z denote local features, and Y the class variable. For each global feature X_i , there is a parameter γ_i . Additionally, for each scope and each local feature Z_i , there is a parameter λ_i^S . Then, the distribution of Y given all the features and weights is

$$P_S(y|x, z, \gamma, \lambda^S) \propto e^{(\gamma \cdot yx + \lambda^S \cdot yz)} \quad (13)$$

The global feature parameters γ are the same across scopes, while the local feature parameters λ^S depend on the scope S . As is often the case, we assume that the global weights can be learned from the training data, so that their values are fixed when we encounter a new scope. The key issue that they try to address in this paper is that the local features are unknown.

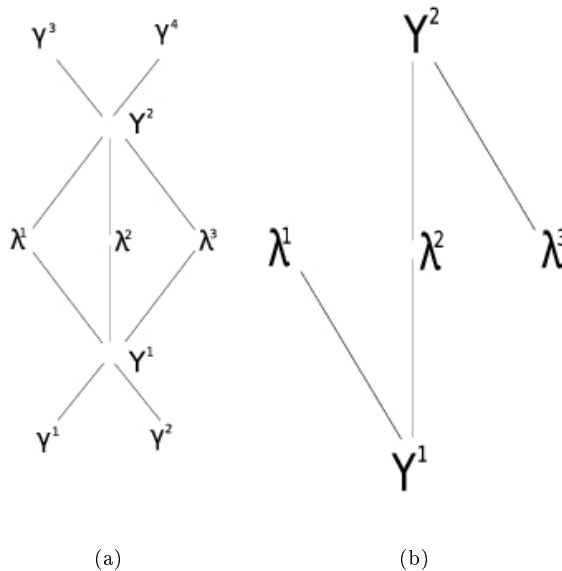


Figure 1: (a) Markov network for two instances, two global and three local features, (b) Conditional Markov network for same settings

Figure 1(a) shows the graphical model where Y represent labels. Since global feature weights are already known and each label depends on few local features, the graphical model simplifies to figure 1(b). In this model, we can see that the labels of all of the instances are correlated with the local feature weights of features they contain, and thereby with each other. Thus, for example, if we obtain evidence (from global features) about the label Y^1 , it would change our posterior beliefs about the local feature weight λ_2 , which in turn, would change our beliefs about the label Y^2 . Thus, by running probabilistic inference over this graphical model, we obtain updated beliefs both about the local feature weights and about the instance labels.

Semi-supervised techniques solves the problem partially. It assumes that test data though unlabeled is from the same domain from which training data is.

5.2 Multiple CRFs

Another approach is to train multiple CRF models, one for each domain. Each model has some properties specific to that domain. During testing, our first task is to identify the domain and apply appropriate

model specific to that domain. If the test data does not belong to the domain in which model is built, then apply a model which is closer to its domain. The time and space complexity of this technique is quite high. Also, by looking at test sequence, identifying the domain and searching the model whose domain is close to it is not trivial.

Another similar approach is to train single model from different domains. In this case, only limited features which are relevant to test sequence will get fired during testing. But it may face severe problems when same set of features are used for different labels on different domains. For example, a word “Branch” may refer to “Tree” in a sequence while “Bank” in another.

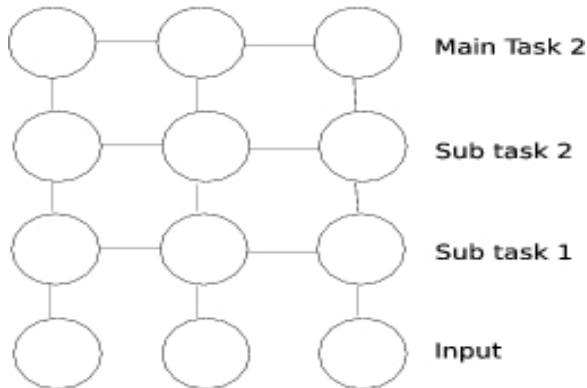


Figure 2: Graphical model for jointly decoded CRF

[11] has suggested one way of combining multiple CRFs into a single grid-shaped factorial CRF as shown in figure 2. For a series of N cascaded tasks, we train individual CRFs separately on each task, using the prediction of the previous CRF as a feature. The feature function for CRF i is of the form $f_k^i(s_{t-1}^i, s_t^i, s_{t-1}^{i-1}, x, t)$. This means that they depend not only on the observed input x and the transition $(s_{t-1}^i \rightarrow s_t^i)$ but also on the state s_{t-1}^{i-1} of the previous CRF. For example, a feature returns value 1 if the current state is SPEAKER NAME, the previous CRF predicted it as PERSON NAME, and the previous word is HOST. In case the prediction made by first CRF is incorrect then the error may flow downwards and gives incorrect prediction.

[11] explained three different ways in which we can combine CRFs during training and testing, viz. Cascaded training and testing, Joint training and testing, Joint testing with cascaded training. Each technique has some advantages and disadvantages.

Techniques suggested by [11] are not directly applicable to our problem. In our case, we can train multiple CRFs for different domains instead of different tasks. Hence cascading technique may not help much but joint technique might give good results.

5.3 Regularisation in CRF

Sometimes due to noise, some features may get deleted from the data. For example in handwritten digit recognition, some pixels may “die” over the course of time. Hence if the model attaches too much weight to a single pixel, it might suffer a performance loss. In other words, if some features get high weight during training and are not present in test data, then model does not perform well. Hence, it is important not to assign too much weight to any single feature. Regularisation is used to spread the weights evenly between the features. It is one of the techniques to avoid over-fitting by penalizing log-likelihood with some prior distribution over the parameters. [12] has done an empirical study on different priors viz. Gaussian, Exponential and Hyperbolic- L_1 prior. A prior distribution encodes prior knowledge about the nature of different models. However, prior knowledge can be difficult to encode reliably and the optimal choice of prior family may vary from task to task. Also, regularisation term depends on the frequency of features. [12] explores feature-dependent variance for the Gaussian prior.

[9] suggested a new parameter-free approach called “logarithmic opinion pool (LOP)”. Results show that a LOP of CRF can outperform a standard unregularised CRF and attain a performance level close to that of regularised CRF.

Apart from regularisation, [?] has introduced new algorithm to avoid feature over-weight by Minimax Problem Formulation. It assumes that from single sequence at-most K features get deleted. Given a labeled example (x_i, y_i) ($i = 1, 2 \dots n$) with input feature vectors $x_i \in R^d$ and class labels $y_i \in \pm 1$ (a special case of segmentation where we assigned single label to the sequence. In Minimax Problem Formulation, the objective function is to minimize the hinge loss: $\sum_i [1 - y_i w \cdot x_i]$. We wish to develop a classifier which minimizes the worst case hinge loss when up-to K features may be deleted from each data vector. In the worst case, hinge loss for a particular example i is given by

$$h(w, y_i, x_i) = \max[1 - y_i w \cdot (x_i * (1 - \alpha_i))] \quad (14)$$

where, $\alpha_i \in \{0, 1\}$ and $\sum_j \alpha_{ij} = K$. Worst case hinge loss over the entire training set is $\sum_i h(w, y_i, x_i)$. Simplification of the above expressions and some variants are explained in [?].

Technique suggested by [?] may not be directly applicable to our problem. If we train our model by deleting only those features which are not relevant in test sequences then it may work well.

5.4 Exploiting Local Regularities

[10] discusses the special case of our problem where both training and test data are from the same domain but from different sources. It assumes that same kind of information available on web from different sources have different structures. For example all publications on Prof. Sunita’s web-page have conference names italicized and author names appearing just after title. It is very difficult to model such features, since such properties are confined to particular “locales” and one cannot determine them at training time. In [10] they build a model which exploits such local regularities in data and improve text segmentation.

Their approach is to use features specific to the problem domain which capture global regularities, and add features capable of modeling locale specific regularities in data. Their model is a two-stage model, which at first stage ignores the locality, takes a unified view of data in the locales and learns a global model, say G_1 . This model uses standard global features specific to the domain. Using this global model as a reference, then augment the data with locale specific statistics, which basically reflect the opinion of G_1 on each locale. At second stage, Locale Aware Learning Model say G_2 is trained using both relativised and global features to capture the local regularities in addition to global regularities.

Only limitation with the approach is, during second stage, where Locale Aware Learning Model only adds features which are locale specific since it assumes test data is from the same domain but different sources. In our case, we can extend this model by deleting those features which are not relevant in test domain.

5.5 Other Approaches

One approach is to play with the features learnt during training. First find the features which are relevant for the test sequences. Either change the weights of those features which are relevant or remove the features which are irrelevant. Three steps suggested below need to be done to follow this approach:

1. Domain checking

If test data is from the same domain then same model (same set of features and weights)can be used, otherwise follow step 2 and 3.

2. Feature selection

Select features which are relevant to the test domain. We performed some experiments to select relevant features discussed in Section 6. Currently most of the techniques depend on the expected value of the features on test data. There is a technique which uses features weights too.

3. Re-learn model

After selecting features, there are different techniques to re-learn the model. One is to change the weights of selected features appropriately. Another technique is to re-build the model using only those selected features.

Above technique will also face the problem of identifying domain. But the problem in this case is not that severe because if same domain gets identified as a different one, then also it wont affect the accuracy because all features become relevant in second step.

6 Work Done

We first generated all the features relevant to training set and learnt the model. During testing we tried to find out only those features which were relevant and re-learned the model using only limited features. We followed different approaches to find relevant features in test data. One approach was to find the expected value of each feature on test data and the features having expected value greater than zero were only considered. Other technique was to find expected value of each feature on test and train data set and normalize the difference by the maximum expected value among them. If the value comes out to be greater than some threshold value then remove that feature. Different results with different threshold values are shown in table 1 and 2. Sometimes it may happen that the difference comes to be very less and feature is not that much important. One solution is to multiply the difference with weights of that feature and if that value comes to be greater than some threshold value then remove that feature. Results of all above techniques are shown in table 1 and 2.

7 Experiments and Conclusion

Experiments were performed using *article* as a training data set and the same data set for testing with all words capitalized. Another experiment was performed using *article* dataset with all words capitalised as training and *article* dataset as testing. Both the experiments used 39 sequences for training and testing. We measured *Precision*, *Recall* and *F1-measure* for the label *author name* in both the experiment. Table 1 and 2 contains results of the former and latter approach respectively. First result in both the tables uses normal CRF without eliminating any features. Second result is obtained by removing those features whose expected value on test set was 0. Third, forth and fifth results are with our second approach, where the average difference thresholds were 0.9, 0.8, and 0.7 respectively. Last result is from the final approach in which we multiplied difference of expected value by weight of that feature. Threshold value used in final approach was 0.9.

<i>Methods</i>	<i>Token Level</i>			<i>Span Level</i>		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure
Standard CRF	95.92	39.66	56.12	75.00	27.00	39.71
testExpected > 0	99.24	54.85	70.65	87.27	48.00	61.94
avgDifference \geq 0.9	78.60	94.51	85.82	63.11	77.00	69.37
avgDifference \geq 0.8	78.20	95.36	85.93	67.21	82.00	73.87
avgDifference \geq 0.7	68.18	56.96	62.07	2.86	1.00	1.48
weightDifference \geq 0.9	95.83	97.05	96.44	86.87	86.00	86.43

Table 1: Training : Article , Testing : Article-Capitalized

Two types of accuracy can be measured. One is at token level and other at span level. We calculate precision, recall and F1-measure for both the types. Token level accuracy means to match each estimated label of token with the exact label, while span level accuracy match each segment. Hence span level accuracy is always less than or equal to token level accuracy. From the results it is clear that the standard CRF performs worse when training and testing data are not from the same distribution. Overall, the last method performs well where we multiply the weights of feature with the difference to filter the feature.

<i>Methods</i>	<i>Token Level</i>			<i>Span Level</i>		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure
Standard CRF	97.22	73.84	83.93	84.72	61.00	70.93
testExpected > 0	97.76	55.27	70.62	85.71	48.00	61.54
avgDifference \geq 0.9	97.81	94.09	95.91	89.58	86.00	87.76
avgDifference \geq 0.8	91.94	96.20	94.02	83.81	88.00	85.85
avgDifference \geq 0.7	90.76	95.36	93.00	82.86	87.00	84.88
weightDifference \geq 0.9	97.38	94.09	95.71	87.76	86.00	86.87

Table 2: Training : Article-Capitalized , Testing : Article

References

- [1] Hanna M. Wallach. Conditional random fields an introduction. Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.
- [2] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-2001)*, Williams, MA, 2001.
- [3] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, 2003.
- [4] Freitag McCallum, A. and Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning. ICML-2000*, 2000.
- [5] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *NIPS*, 2004.
- [6] Chi-Hoon Lee Feng Jiao, Shaojun Wang and Dale Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling.
- [7] Wei Li and Andrew McCallum. A note on semi-supervised learning using markov random fields. 2004.
- [8] Ming Fai Wong Benjamin Taskar and Daphne Koller. Learning on the test data leveraging unseen features. ICML, 2003.
- [9] Charles Sutton and Andrew McCallum. Composition of conditional random fields for transfer learning. In *Empirical Methods in Natural Language Processing*, 2005.
- [10] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics annual meeting*, pages 329 – 336, 2004.
- [11] Andrew Smith and Miles Osborne. Regularization techniques for conditional random fields: Parameterised versus parameter-free.
- [12] Amir Globerson and Sam Rowels. Nightmare at test time: Robust learning by feature deletion.
- [13] Utkarsh Jain. Exploiting local regularities in text segmentation using conditional random fields. 2006.