Using Gazetteers in Discriminative Information Extraction

Andrew Smith
Division of Informatics
University of Edinburgh
United Kingdom
a.p.smith-2@sms.ed.ac.uk

Miles Osborne
Division of Informatics
University of Edinburgh
United Kingdom
miles@inf.ed.ac.uk

Abstract

Much work on information extraction has successfully used gazetteers to recognise uncommon entities that cannot be reliably identified from local context alone. Approaches to such tasks often involve the use of maximum entropy-style models, where gazetteers usually appear as highly informative features in the model. Although such features can improve model accuracy, they can also introduce hidden negative effects. In this paper we describe and analyse these effects and suggest ways in which they may be overcome. In particular, we show that by quarantining gazetteer features and training them in a separate model, then decoding using a logarithmic opinion pool (Smith et al., 2005), we may achieve much higher accuracy. Finally, we suggest ways in which other features with gazetteer feature-like behaviour may be identified.

1 Introduction

In recent years discriminative probabilistic models have been successfully applied to a number of information extraction tasks in natural language processing (NLP), such as named entity recognition (NER) (McCallum and Li, 2003), noun phrase chunking (Sha and Pereira, 2003) and information extraction from research papers (Peng and McCallum, 2004). Discriminative models offer a significant advantage over their generative counterparts by allowing the specification of powerful, possibly non-independent features which would be difficult to tractably encode in a generative model.

In a task such as NER, one sometimes encounters an entity which is difficult to identify using local contextual cues alone because the entity has not be seen before. In these cases, a **gazetteer** or dictionary of possible entity identifiers is often useful. Such identifiers could be names of people, places, companies or other organisations. Using gazetteers one may define additional features in the model that represent the dependencies between a word's NER label and its presence in a particular gazetteer. Such gazetteer features are often highly informative, and their inclusion in the model should in principle result in higher model accuracy. However, these features can also introduce hidden negative effects taking the form of labelling errors that the model makes at places where a model without the gazetteer features would have labelled correctly. Consequently, ensuring optimal usage of gazetteers can be difficult.

In this paper we describe and analyse the labelling errors made by a model, and show that they generally result from the model's over-dependence on the gazetteer features for making labelling decisions. By including gazetteer features in the model we may, in some cases, transfer too much explanatory dependency to the gazetteer features from the non-gazetteer features. In order to avoid this problem, a more careful treatment of these features is required during training. We demonstrate that a traditional regularisation approach, where different features are regularised to different degrees, does not offer a sat-

Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X),
pages 133–140, New York City, June 2006. ©2006 Association for Computational Linguistics

isfactory solution. Instead, we show that by training gazetteer features in a separate model to the other features, and decoding using a **logarithmic opinion pool** (LOP) (Smith et al., 2005), much greater accuracy can be obtained. Finally, we identify other features with gazetteer feature-like properties and show that similar results may be obtained using our method with these features.

We take as our model a linear chain conditional random field (CRF), and apply it to NER in English.

2 Conditional Random Fields

A linear chain conditional random field (CRF) (Lafferty et al., 2001) defines the conditional probability of a label sequence **s** given an observed sequence **o** via:

$$p(\mathbf{s} \mid \mathbf{o}) = \frac{1}{Z(\mathbf{o})} \exp\left(\sum_{t=1}^{T+1} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t) \right) \quad (1)$$

where T is the length of both sequences, λ_k are parameters of the model and $Z(\mathbf{o})$ is a partition function that ensures that (1) represents a probability distribution. The functions f_k are feature functions representing the occurrence of different events in the sequences **s** and **o**.

The parameters λ_k can be estimated by maximising the conditional log-likelihood of a set of labelled training sequences. At the maximum likelihood solution the model satisfies a set of feature constraints, whereby the expected count of each feature under the model is equal to its empirical count on the training data:

$$E_{\tilde{p}(\mathbf{o},\mathbf{s})}[f_k] - E_{p(\mathbf{s}|\mathbf{o})}[f_k] = 0,\ \forall k$$

In general this cannot be solved for the λ_k in closed form, so numerical optimisation must be used. For our experiments we use the limited memory variable metric (LMVM) (Sha and Pereira, 2003) routine, which has become the standard algorithm for CRF training with a likelihood-based objective function.

To avoid overfitting, a prior distribution over the model parameters is typically used. A common example of this is the Gaussian prior. Use of a prior involves adding extra terms to the objective and its derivative. In the case of a Gaussian prior, these additional terms involve the mean and variance of the distribution.

3 Previous Use of Gazetteers

Gazetteers have been widely used in a variety of information extraction systems, including both rule-based systems and statistical models. In addition to lists of people names, locations, etc., recent work in the biomedical domain has utilised gazetteers of biological and genetic entities such as gene names (Finkel et al., 2005; McDonald and Pereira, 2005). In general gazetteers are thought to provide a useful source of external knowledge that is helpful when an entity cannot be identified from knowledge contained solely within the data set used for training. However, some research has questioned the usefulness of gazetteers (Krupka and Hausman, 1998). Other work has supported the use of gazetteers in general but has found that lists of only moderate size are sufficient to provide most of the benefit (Mikheev et al., 1999). Therefore, to date the effective use of gazetteers for information extraction has in general been regarded as a "black art". In this paper we explain some of the likely reasons for these findings, and propose ways to more effectively handle gazetteers when they are used by maxent-style models.

In work developed independently and in parallel to the work presented here, Sutton et al. (2006) identify general problems with gazetteer features and propose a solution similar to ours. They present results on NP-chunking in addition to NER, and provide a slightly more general approach. By contrast, we motivate the problem more thoroughly through analysis of the actual errors observed and through consideration of the success of other candidate solutions, such as traditional regularisation over feature subsets.

4 Our Experiments

In this section we describe our experimental setup, and provide results for the baseline models.

4.1 Task and Dataset

Named entity recognition (NER) involves the identification of the location and type of pre-defined entities within a sentence. The CRF is presented with a set of sentences and must label each word so as to indicate whether the word appears outside an entity, at the beginning of an entity of a certain type or within the continuation of an entity of a certain type.

Our results are reported on the CoNLL-2003 shared task English dataset (Sang and Meulder, 2003). For this dataset the entity types are: persons (PER), locations (LOC), organisations (ORG) and miscellaneous (MISC). The training set consists of $14,987$ sentences and $204,567$ tokens, the development set consists of $3,466$ sentences and $51,578$ tokens and the test set consists of $3,684$ sentences and $46,666$ tokens.

4.2 Gazetteers

We employ a total of seven gazetteers for our experiments. These cover names of people, places and organisations. Specifically, we have gazetteers containing surnames ($88,799$ entries), female first names ($4,275$ entries), male first names ($1,219$ entries), names of places ($27,635$ entries), names of companies ($20,638$ and $279,195$ entries) and names of other organisations (425 entries).

4.3 Feature set

Our experiments are centred around two CRF models, one with and one without gazetteer features. The model *without* gazetteer features, which we call **standard**, comprises features defined in a window of five words around the current word. These include features encoding *n*-grams of words and POS tags, and features encoding orthographic properties of the current word. The orthographic features are based on those found in (Curran and Clark, 2003). Examples include whether the current word is capitalised, is an initial, contains a digit, contains punctuation, etc. In total there are $450,345$ features in the **standard** model.

We call the second model, *with* gazetteer features, **standard+g**. This includes all the features contained in the **standard** model as well as $8,329$ gazetteer features. Our gazetteer features are a typical way to represent gazetteer information in maxent-style models. They are divided into two categories: *unlexicalised* and *lexicalised*. The unlexicalised features model the dependency between a word's presence in a gazetteer and its NER label, irrespective of the word's identity. The lexicalised features, on the other hand, include the word's identity and so provide more refined word-specific modelling of the gazetteer-NER label dependency.[1] There are 35 unlexicalised gazetteer features and $8,294$ lexicalised gazetteer features, giving a total of $458,675$ features in the **standard+g** model.

Model	Development		Test	
	Unreg.	Reg.	Unreg.	Reg.
standard	88.21	89.86	81.60	83.97
standard+g	89.19	90.40	83.10	84.70

Table 1: Model F scores

		standard+g	
		✓	✗
standard	✓	44,945	160
	✗	228	1,333

Table 2: Test set errors

4.4 Baseline Results

Table 1 gives F scores for the **standard** and **standard+g** models. Development set scores are included for completeness, and are referred to later in the paper. We show results for both unregularised and regularised models. The regularised models are trained with a zero-mean Gaussian prior, with the variance set using the development data.

We see that, as expected, the presence of the gazetteer features allows **standard+g** to outperform **standard**, for both the unregularised and regularised models. To test significance, we use McNemar's matched-pairs test (Gillick and Cox, 1989) on pointwise labelling errors. In each case, the **standard+g** model outperforms the **standard** model at a significance level of $p < 0.02$. However, these results camouflage the fact that the gazetteer features introduce some negative effects, which we explore in the next section. As such, the real benefit of including the gazetteer features in **standard+g** is not fully realised.

5 Problems with Gazetteer Features

We identify problems with the use of gazetteer features by considering test set labelling errors for both **standard** and **standard+g**. We use regularised models here as an illustration. Table 2 shows the

[1] Many gazetteer entries involve strings of words where the individual words in the string do not appear in the gazetteer in isolation. For this reason the lexicalised gazetteer features are *not* simply determined by the word identity features.

number of sites (a site being a particular word at a particular position in a sentence) where labellings have improved, worsened or remained unchanged with respect to the gold-standard labelling with the addition of the gazetteer features. For example, the value in the top-left cell is the number of sites where both the **standard** and **standard+g** label words correctly.

The most interesting cell in the table is the top-right one, which represents sites where **standard** is correctly labelling words but, with the addition of the gazetteer features, **standard+g** mislabels them. At these sites, the addition of the gazetteer features actually worsens things. How well, then, could the **standard+g** model do if it could somehow reduce the number of errors in the top-right cell? In fact, if it had correctly labelled those sites, a significantly higher test set F score of 90.36% would have been obtained. This potential upside suggests much could be gained from investigating ways of correcting the errors in the top-right cell. It is not clear whether there exists any approach that could correct *all* the errors in the top-right cell while simultaneously maintaining the state in the other cells, but approaches that are able to correct at least some of the errors should prove worthwhile.

On inspection of the sites where errors in the top-right cell occur, we observe that some of the errors occur in sequences where no words are in any gazetteer, so no gazetteer features are active for any possible labelling of these sequences. In other cases, the errors occur at sites where some of the gazetteer features appear to have dictated the label, but have made an incorrect decision. As a result of these observations, we classify the errors from the top-right cell of Table 2 into two types: *type A* and *type B*.

5.1 Type A Errors

We call type A errors those errors that occur at sites where gazetteer features seem to have been *directly* responsible for the mislabelling. In these cases the gazetteer features effectively "over-rule" the other features in the model causing a mislabelling where the **standard** model, without the gazetteer features, correctly labels the word.

An example of a type A error is given in the sentence extract below:

```
about/O Healy/I-LOC
```

This is the labelling given by **standard+g**. The correct label for `Healy` here is `I-PER`. The **standard** model is able to decode this correctly as `Healy` appears in the training data with the `I-PER` label. The reason for the mislabelling by the **standard+g** model is that `Healy` appears in both the gazetteer of place names and the gazetteer of person surnames. The feature encoding the gazetteer of place names with the `I-LOC` label has a λ value of 4.20, while the feature encoding the gazetteer of surnames with the `I-PER` label has a λ value of 1.96, and the feature encoding the word `Healy` with the `I-PER` label has a λ value of 0.25. Although other features both at the word `Healy` and at other sites in the sentence contribute to the labelling of `Healy`, the influence of the first feature above dominates. So in this case the addition of the gazetteer features has confused things.

5.2 Type B Errors

We call type B errors those errors that occur at sites where the gazetteer features seem to have been only *indirectly* responsible for the mislabelling. In these cases the mislabelling appears to be more attributable to the non-gazetteer features, which are in some sense less expressive after being trained with the gazetteer features. Consequently, they are less able to decode words that they could previously label correctly.

An example of a type B error is given in the sentence extract below:

```
Chanderpaul/O was/O
```

This is the labelling given by **standard+g**. The correct labelling, given by **standard**, is `I-PER` for `Chanderpaul`. In this case no words in the sentence (including the part not shown) are present in any of the gazetteers so no gazetteer features are active for any labelling of the sentence. Consequently, the gazetteer features do not contribute at all to the labelling decision. Non-gazetteer features in **standard+g** are, however, unable to find the correct labelling for `Chanderpaul` when they previously could in the **standard** model.

For both type A and type B errors it is clear that the gazetteer features in **standard+g** are in some sense too “powerful” while the non-gazetteers features have become too “weak”. The question, then, is: can we train all the features in the model in a more sophisticated way so as to correct for these effects?

6 Feature Dependent Regularisation

One interpretation of the findings of our error analysis above is that the addition of the gazetteer features to the model is having an implicit over-regularising effect on the other features. Therefore, is it possible to adjust for this effect through more careful explicit regularisation using a prior? Can we directly regularise the gazetteer features more heavily and the non-gazetteer features less? We investigate this possibility in this section.

The **standard+g** model is regularised by fitting a single Gaussian variance hyperparameter across all features. The optimal value for this single hyperparameter is 45. We now relax this single constraint by allocating a separate variance hyperparameter to different feature subsets, one for the gazetteer features (σ_{gaz}) and one for the non-gazetteer features ($\sigma_{non\text{-}gaz}$). The hope is that the differing subsets of features are best regularised using different prior hyperparameters. This is a natural approach within most standardly formulated priors for log-linear models. Clearly, by doing this we increase the search space significantly. In order to make the search manageable, we constrain ourselves to three scenarios: (1) Hold $\sigma_{non\text{-}gaz}$ at 45, and regularise the gazetteer features a little more by reducing σ_{gaz}. (2) Hold σ_{gaz} at 45, and regularise the non-gazetteer features a little less by increasing $\sigma_{non\text{-}gaz}$. (3) Simultaneously regularise the gazetteer features a little more than at the single variance optimum, and regularise the non-gazetteer features a little less.

Table 3 gives representative development set F scores for each of these three scenarios, with each scenario separated by a horizontal dividing line. We see that in general the results do not differ significantly from that of the single variance optimum. We conjecture that the reason for this is that the regularising effect of the gazetteer features on the non-gazetteer features is due to relatively subtle interactions during training that relate to the dependencies the features encode and how these dependencies overlap. Regularising different feature subsets by different amounts with a Gaussian prior does not directly address these interactions but instead just rather crudely penalises the magnitude of the parameter values of different feature sets to different degrees. Indeed this is true for any standardly formulated prior. It seems therefore that any solution to the regularising problem should come through more explicit restricting or removing of the interactions between gazetteer and non-gazetteer features during training.

σ_{gaz}	$\sigma_{non-gaz}$	F score
42	45	90.40
40	45	90.30
45	46	90.39
45	50	90.38
44.8	45.2	90.41
43	47	90.35

Table 3: FDR development set F scores

7 Combining Separately Trained Models

We may remove interactions between gazetteer and non-gazetteer features entirely by quarantining the gazetteer features and training them in a separate model. This allows the non-gazetteer features to be protected from the over-regularising effect of the gazetteer features. In order to decode taking advantage of the information contained in both models, we must combine the models in some way. To do this we use a **logarithmic opinion pool** (LOP) (Smith et al., 2005). This is similar to a mixture model, but uses a weighted multiplicative combination of models rather than a weighted additive combination. Given models p_α and per-model weights w_α, the LOP distribution is defined by:

$$p_{\text{LOP}}(\mathbf{s}\,|\,\mathbf{o}) = \frac{1}{Z_{\text{LOP}}(\mathbf{o})}\prod_{\alpha}[p_\alpha(\mathbf{s}\,|\,\mathbf{o})]^{w_\alpha} \tag{2}$$

with $w_\alpha \geq 0$ and $\sum_\alpha w_\alpha = 1$, and where $Z_{\text{LOP}}(\mathbf{o})$ is a normalising function. The weight w_α encodes the dependence of the LOP on model α. In the case of a CRF, the LOP itself is a CRF and so decoding is no more complex than for standard CRF decoding.

In order to use a LOP for decoding we must set the weights w_α in the weighted product. In (Smith et

Feature Subset	Feature Type
s1	simple structural features
s2	advanced structural features
n	*n*-grams of words and POS tags
o	simple orthographic features
a	advanced orthographic features
g	gazetteer features

Table 4: **standard+g** feature subsets

al., 2005) a procedure is described whereby the (normalised) weights are explicitly trained. In this paper, however, we only construct LOPs consisting of two models in each case, one model with gazetteer features and one without. We therefore do not require the weight training procedure as we can easily fit the two weights (only one of which is free) using the development set.

To construct models for the gazetteer and non-gazetteer features we first partition the feature set of the **standard+g** model into the subsets outlined in Table 4. The *simple structural features* model label-label and label-word dependencies, while the *advanced structural features* include these features as well as those modelling label-label-word conjunctions. The *simple orthographic features* measure properties of a word such as capitalisation, presence of a digit, etc., while the *advanced orthographic properties* model the occurrence of prefixes and suffixes of varying length.

We create and train different models for the gazetteer features by adding different feature subsets to the gazetteer features. We regularise these models in the usual way using a Gaussian prior. In each case we then combine these models with the **standard** model and decode under a LOP.

Table 5 gives results for LOP decoding for the different model pairs. Results for the **standard+g** model are included in the first row for comparison. For each LOP the hyphen separates the two models comprising the LOP. So, for example, in the second row of the table we combine the gazetteer features with simple structural features in a model, train and decode with the **standard** model using a LOP. The simple structural features are included so as to provide some basic support to the gazetteer features.

We see from Table 5 that the first two LOPs significantly outperform the regularised **standard+g** model (at a significance level of $p < 0.01$, on both the test and development sets). By training the gazetteer features separately we have avoided their over-regularising effect on the non-gazetteer features. This relies on training the gazetteer features with a relatively small set of other features. This is illustrated as we read down the table, below the top two rows. As more features are added to the model containing the gazetteer features we obtain decreasing test set F scores because the advantage created from separate training of the features is increasingly lost.

LOP	Dev Set	Test Set
standard+g	90.40	84.70
s1g-standard	**91.34**	**85.98**
s2g-standard	91.32	85.59
s2ng-standard	90.66	84.59
s2nog-standard	90.47	84.92
s2noag-standard	90.56	84.78

Table 5: Reg. LOP F scores

LOP	LOP Weights
s1g-standard	[0.39, 0.61]
s2g-standard	[0.29, 0.71]
s2ng-standard	[0.43, 0.57]
s2nog-standard	[0.33, 0.67]
s2noag-standard	[0.39, 0.61]

Table 6: Reg. LOP weights

Table 6 gives the corresponding weights for the LOPs in Table 5, which are set using the development data. We see that in every case the LOP allocates a smaller weight to the gazetteer features model than the non-gazetteer features model and in doing so restricts the influence that the gazetteer features have in the LOP's labelling decisions.

Table 7, similar to Table 2 earlier, shows test set labelling errors for the **standard** model and one of the LOPs. We take the **s2g-standard** LOP here for illustration. We see from the table that the number of errors in the top-right cell shows a reduction of 29% over the corresponding value in Table 2. We have therefore reduced the number errors of the type we were targeting with our approach. The approach has also had the effect of reducing the number of errors in the bottom-right cell, which further improves model accuracy.

All the LOPs in Table 5 contain regularised mod-

		s2g-standard LOP	
		✓	✗
standard	✓	44,991	114
	✗	305	1,256

Table 7: Test set errors

LOP	Dev Set	Test Set
s1g-standard	**90.58**	**84.87**
s2g-standard	90.70	84.28
s2ng-standard	89.70	84.01
s2nog-standard	89.48	83.99
s2noag-standard	89.40	83.70

Table 8: Unreg. LOP F scores

els. Table 8 gives test set F scores for the corresponding LOPs constructed from unregularised models. As we would expect, the scores are lower than those in Table 5. However, it is interesting to note that the **s1g-standard** LOP still outperforms the *regularised* **standard+g** model.

In summary, by training the gazetteer features and non-gazetteer features in separate models and decoding using a LOP, we are able to overcome the problems described in earlier sections and can achieve much higher accuracy. This shows that successfully deploying gazetteer features within maxent-style models should involve careful consideration of restrictions on how features interact with each other, rather than simply considering the absolute values of feature parameters.

8 Gazetteer-Like Features

So far our discussion has focused on gazetteer features. However, we would expect that the problems we have described and dealt with in the last section also occur with other types of features that have similar properties to gazetteer features. By applying similar treatment to these features during training we may be able harness their usefulness to a greater degree than is currently the case when training in a single model. So how can we identify these features?

The task of identifying the optimal partitioning for creation of models in the previous section is in general a hard problem as it relies on clustering the features based on their explanatory power relative to all other clusters. It may be possible, however, to devise some heuristics that approximately correspond to the salient properties of gazetteer features (with respect to the clustering) and which can then be used to identify other features that have these properties. In this section we consider three such heuristics. All of these heuristics are motivated by the observation that gazetteer features are both highly discriminative and generally very sparse.

Family Singleton Features We define a feature *family* as a set of features that have the same conjunction of predicates defined on the observations. Hence they differ from each other only in the NER label that they encode. *Family singleton features* are features that have a count of 1 in the training data when all other members of that feature family have zero counts. These features have a flavour of gazetteer features in that they represent the fact that the conjunction of observation predicates they encode is highly predictive of the corresponding NER label, and that they are also very sparse.

Family *n*-ton Features These are features that have a count of *n* (greater than 1) in the training data when all other members of that feature family have zero counts. They are similar to family singleton features, but exhibit gazetteer-like properties less and less as the value of *n* is increased because a larger value of *n* represents less sparsity.

Loner Features These are features which occur with a low mean number of other features in the training data. They are similar to gazetteer features in that, at the points where they occur, they are in some sense being relied upon more than most features to explain the data. To create loner feature sets we rank all features in the **standard+g** model based on the mean number of other features they are observed with in the training data, then we take subsets of increasing size. We present results for subsets of size 500, 1000, 5000 and 10000.

For each of these categories of features we add simple structural features (the **s1** set from earlier), to provide basic structural support, and then train a regularised model. We also train a regularised model consisting of all features in **standard+g** except the features from the category in question. We decode these model pairs under a LOP as described earlier.

Table 9 gives test set F scores for LOPs created from each of the categories of features above

LOP	Test Set
FSF	85.79
FnF	84.78
LF 500	85.80
LF 1000	85.70
LF 5000	85.77
LF 10000	85.62

Table 9: Reg. LOP F scores

(with abbreviated names derived from the category names). The results show that for the *family singleton features* and each of the *loner feature* sets we obtain LOPs that significantly outperform the regularised **standard+g** model ($p < 0.0002$ in every case). The *family n-ton features'* LOP does not do as well, but that is probably due to the fact that some of the features in this set have a large value of *n* and so behave much less like gazetteer features.

In summary, we obtain the same pattern of results using our quarantined training and LOP decoding method with these categories of features that we do with the gazetteer features. We conclude that the problems with gazetteer features that we have identified in this paper are exhibited by general discriminative features with gazetteer feature-like properties, and our method is also successful with these more general features. Clearly, the heuristics that we have devised in this section are very simple, and it is likely that with more careful engineering better feature partitions can be found.

9 Conclusion and future work

In this paper we have identified and analysed negative effects that can be introduced to maxent-style models by the inclusion of highly discriminative gazetteer features. We have shown that such effects manifest themselves through errors that generally result from the model's over-dependence on the gazetteer features for decision making. To overcome this problem a more careful treatment of these features is required during training. We have proposed a solution that involves quarantining the features and training them separately to the other features in the model, then decoding the separate models with a logarithmic opinion pool. In fact, the LOP provides a natural way to handle the problem, with different constituent models for the different feature types. The method leads to much greater accuracy, and allows the power of gazetteer features to be more effectively harnessed. Finally, we have identified other feature sets with gazetteer feature-like properties and shown that similar results may be obtained using our method with these feature sets.

In this paper we defined intuitively-motivated feature partitions (gazetteer feature-based or otherwise) using heuristics. In future work we will focus on automatically determining such partitions.

References

James Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proc. CoNLL-2003*.

Jenny Finkel, Shipra Dingare, Christopher D. Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the boundaries: gene and protein identification in biomedical text. *BMC Bioinformatics*, (6).

L. Gillick and Stephen Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 532–535.

George R. Krupka and Kevin Hausman. 1998. Isoquest Inc: Description of the NetOwl (TM) extractor system as used for MUC-7. In *Proc. MUC-7*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML 2001*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. CoNLL-2003*.

Ryan McDonald and Fernando Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, (6).

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers.

Fuchun Peng and Andrew McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Proc. HLT-NAACL 2004*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. CoNLL-2003*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. HLT-NAACL 2003*.

Andrew Smith, Trevor Cohn, and Miles Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *Proc. ACL 2005*.

Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing weight undertraining in struxctured discriminative learning. In *Proc. HLT/NAACL 2006*.