Applying exponential models to structured prediction problems
CS 546 Final Paper

Haoyan Cai
hcai6@illinois.edu

Cassandra L. Jacobs
cljacob2@illinois.edu

Yang Song
ysong71@illinois.edu

Wenzhu Tong
wtong8@illinois.edu

Abstract

The exponential family of models is highly flexible, relying on features to predict labels based on their conditional likelihood. Among these models, conditional random fields have been successfully applied to sequence labeling, and logistic regression is a commonly employed all-purpose classifier. More specifically, our system used CRF-based models for entity extraction and logistic regression classifiers for the coreference resolution and relation identification tasks on the ACE2005 dataset. We found that the named entity recognition task had good performance with varying degrees of success based on different features and strategies. Coreference resolution was largely successful but significant improvements could be made by incorporating prior knowledge into the model about whether two entities are semantically related. Relations were first filtered and then classified into six categories. This generally had high recall but low precision, with word-level and syntactic features increasing performance.

1 Introduction

Exponential models allow for predictive models to include a number of different features at the same time when performing different NLP tasks. In the natural language proprecessing (NLP) community, conditional random fields (CRF) and logistic regression models have demonstrated great power (Martin and Jurafsky, 2000; Lafferty et al., 2001) in tasks such as sequence labeling and classification. In this paper, we discuss how we extract entities from text, resolve co-referring expressions, and identify relations between entities from the ACE2005 dataset (Doddington et al., 2004) as well as other knowledge bases. To be more specific, a linear-chain CRF model is utilized for entity extraction, and logistic regression models are applied to the coreference resolution and relation identification tasks. In each of the three sections of the paper, we examine the impact of different features, including word, string, syntactic, and external knowledge features including gazzeteers, Wikipedia, WordNet and Brown clusters. We also discuss our approaches to balancing the dataset and filtering irrelevant examples. Extensive experiments on ACE2005 dataset show the effectiveness of our proposed models on the three tasks.

2 Entity Extraction

We develop an entity identification method on top of Conditional Random Field (Lafferty et al., 2001), which is the state-of-the-art sequence labeling approach in NLP tasks due to its flexibility to incorporate various types of features. We are using the Conditional Random Field implementation of *python-crfsuite* [1], which uses L-BFGS training algorithm with Elastic Net($L_1 + L_2$) regularization for efficient training of a linear chain CRF model.

2.1 Features

In this section we introduce the features we used in NER model, including baseline word features, external knowledge like gazetteers and Brown clusters,

[1] https://github.com/tpeng/python-crfsuite

and non-local features such as context aggregation and two-stage predictions (Ratinov and Roth, 2009).

Baseline Features

Our baseline features include word form, affix, POS tags, capitalization, digits, and the features of the neighboring words within a window size of 2.

External Knowledge

Features from external knowledge are also incorporated in our NER model, including gazetteers, brown clusters, and phrase mining features, as described below.

Gazetteers

The gazetteer consists of 14 high-precision low-recall lists about common names, countries, etc, plus 16 gazetteers extracted from Wikipedia that cover over 1.5M entities (Ratinov and Roth, 2009). We use the Illinois NER tool to extract the 286 binary features, each representing whether the word matches a type in a gazetteer. If a word matches a gazetteer, then it is more likely to be within an entity mention.

Brown Clusters

Brown clusters (Brown et al., 1992) group similar words in the same cluster in an unsupervised manner, which helps to address the data sparsity problem. We use the results from Liang (2005), where different length of the cluster encoding represents different level of word abstraction (see (Ratinov and Roth, 2009) for more details).

Phrases

Recently a data-driven phrase mining algorithm, SegPhrase, was proposed to find high quality phrases from a large corpus (Liu et al., 2015). Since phrases are related to the boundary detection task in NER, we incorporate the results from SegPhrase as features in our model, where the discovered phrases are encoded in BIO or BILOU schema.

Non-Local Features

It is widely noted in NER community that identical tokens tend to have identical label assignments (Ratinov and Roth, 2009; Finkel et al., 2005). However, the tokens might be far from each other, making the inference intractable in a linear-chain CRF. Although there are some CRF models carefully designed to incorporate those constraints, the models are much more complicated and less efficient. Therefore we propose two families of non-local features as a tradeoff.

Context Aggregation

When the same word appears in several locations in the same document, we aggregate the context (i.e., tokens within window size 2 on both sides) across all the instances.

Two Stage Predictions

Inspired by Krishan and Manning (2006), we first apply a simple NER model (using baseline and local features in our system) to make predictions, and then append the predictions as features in the second stage of inference. Some instances of a word appear in easily identifiable contexts, and their labels could be strong signals for the other occurrences, improving accuracy.

2.2 Experiments

Our experiments are performed on the ACE (Automatic Content Extraction) 2005 **bn** and **nw** dataset [2]. The training and test dataset contains 196 of 215 files and 111 of 117 files respectively. In the training and testing datasets there are 12146 and 7774 head mentions respectively.

We studied the impact of the proposed features, as well as encoding schemas and mention types. We report entity-based precision, recall, and F1 score, where only an exact match of entity type and boundary is counted as correct. Our final system uses BILOU encoding on head mentions, and includes all local features plus two stage predictions, achieving an F1 score of 0.786.

2.2.1 Preliminaries

We first compare two encoding schemas, i.e., BIO and BILOU, and discuss whether to use the entire extent or just the head of the entities. Only baseline and local features are used in this experiment. As shown in Table 1, BILOU encoding significantly outperforms BIO. Performance on head mentions is much better because the mention extent might overlap with other mentions in ACE dataset, which

[2] https://piazza.com/class/ijolw690hly6jy?cid=79

Table 1: Encoding and Mention Type

	Precision	Recall	F1
BIO - head	0.759	0.710	0.734
BILOU - extent	0.705	0.567	0.629
BILOU - head	**0.812**	**0.760**	**0.785**

makes the encoding more chaotic. Therefore we use BILOU encoding on head mentions in our later experiments.

2.2.2 Local Features

Table 2 presents the impact of local features, including baseline and external knowledge features. The baseline features already achieve a reasonably good performance, with a F1 score of 0.746. For local features, gazetteer and Brown clusters are quite effective, since they effectively alleviate the problem of unseen words and thus significantly increase the recall. Phrase features also provide marginal improvement. Their combination results in a highest F1 score of 0.785, which indicates that the three sets of features are complimentary.

Table 2: Impact of Local Features. B stands for Baseline.

	Precision	Recall	F1
Baseline	0.800	0.700	0.746
B+gazetteer	**0.819**	0.742	0.779
B+brown	0.788	0.734	0.761
B+phrase	0.802	0.701	0.748
B+all	0.812	**0.769**	**0.785**

2.2.3 Non-Local Features

We then study how non-local features influence the NER performance in addition to the baseline and all local features, as illustrated in Table 3. It is surprising that the two approaches do not improve the local model much. Two stage predictions marginally improve the performance (F1 score 0.786), while context aggregation is even worse than just using local features. This might be because the linear-chain CRF model is not very capable to include non-local features.

Table 3: Impact of Non-Local Features. L stands for Local features (plus baseline)

	Precision	Recall	F1
Local	0.812	**0.769**	0.785
L+context	0.809	0.750	0.778
L+two-stage	**0.816**	0.759	**0.786**

3 Coreference Resolution

In this section we discuss how we identified whether two mentions are coreferent or not, which is formulated as a binary classification task, and is implemented using logistic regression provided by the Python package *scikit-learn* in conjunction with *pandas* (Pedregosa et al., 2011).

3.1 Data

The dataset used in training included all non-test documents from the newswire and broadcast news sets of the ACE05 dataset. In addition to these datasets, the other sections of ACE2005 were used, include weblogs, talk shows, forums, and other sources, to a total of 436 documents. Mentions in ACE2005 are grouped by document entities, with coreferent mentions belonging to the same entity. All entities were tagged using the Stanford Part-of-Speech tagger (Toutanova et al., 2003).

To develop balanced training and test sets, a balancing procedure was used. For each mention of an entity e in a document, add a link between all possible pairs of mentions, enforcing symmetry so that if e_1 and e_2 are linked that a link for e_2 and e_1 are not treated as different pairs. For each mention pair that is generated, also generate a random non-coreferent mention by selecting at random a mention from the next entity's list of mentions. This selection process generated 446,146 training items.

Test items were generated in the same way as the training set, resulting in 53,444 test items with 26,722 positive and negative examples. Test occurs only on the gold labels without mention detection.

3.2 Features

The entire entity mention extents are used to generate features without lemmatization. Extracted features are common to many other studies (Bengtsson and Roth, 2008; Luo, 2007; Finkel and Manning, 2008; Durrett and Klein, 2013), including whether the two terms are an exact string, number of overlapping words, pronoun status of both mentions, the mention types of the first and second mentions (name, noun, etc.), whether the mentions have the same mention type (e.g. both pronouns or proper nouns), number of shared modifiers, number of shared nouns, whether one is a substring of another like *Russia* and *Russian* (Culotta et al.,), and distance in characters between the two mentions. We also expanded the feature space using the dual representation of the features and compared this to the primal representation.

3.3 Experiment

We trained a binary classifier using the logistic regression using the above features with L_2 regularization.

Evaluation

We attempted to perform coreference classification on the named entities we identified but found that there was a significant drop in performance so here we only evaluate on gold entity mentions extracted from ACE. We compare the performance of regularized and unregularized dual models to the primal model. The evaluation does not include b^3 or MUC evaluation metrics (Cai and Strube, 2010), so we report only precision, recall, and F1 in Table 4.

Table 4: Coreference model performance as a function of feature representation and regularization constraints.

	Precision	Recall	F1
Primal	**0.71**	**0.70**	**0.70**
Dual	0.65	0.64	0.64
Dual (regularized)	0.58	0.57	0.55

Coreference classification is moderately successful, but performance is below that of many similar models that do not require parameters to conform to a probability distribution. This coupled with the lack of external knowledge likely contribute to the modest performance of the model.

4 Relation Extraction

The task of relation extraction is to identify whether relations exists between two entities, and what type of relation it is if there is an relation. We implemented the logistic regression model using Python package scikit-learn for this task.

4.1 Features

This section describes the features we used for relation extraction task. The selection of features was inspired by several prior studies (Chan and Roth, 2010; Kambhatla, 2004; Choi et al., 2006) including entity types, POS tags, and head words of two mentions, the distance between two mentions in the parse tree of a sentence, overlapping and inclusion of two mentions, and whether the head context of one mention is found in the wikipedia page of another entity. The same features are used in the binary filter and multi-class classifier.

4.2 Experiment

The relation extraction task is split into three stages: candidate generation, filtering and multi-class classification. In the candidate generation step, we extracted all entity pairs as candidates. This generates a highly imbalanced dataset. In second step, we applied a filter to filter out the entity pairs with no relations. Our goal for this step is to increase the recall as much as possible. We used two filters for this step. In the first stage, we collected all entity mentions in the training set that have relations. In the testing phase, if neither mention was captured by the filter, we predicted no relation. The second stage is a binary classifier using the result from the first filter. The third stage is a multi-class classifier to judge the six types of relation between two mentions. The training data set and test sets have over 30,000 and 20,000 entity mention pairs respectively. All models were trained using the scikit-learn logistic regression module.

Evaluation

The performance of the binary classifier is evaluated according to whether two mentions have a relation, and if so what kind. For the filter, the precision is 0.45, and the recall is 0.85. The average precision, recall and F1 for the 7 classes is reported in Table 5. In Table 6 we report the impact of features to the overall performance.

We noticed that if we filter out all no-relation entity pairs, the multi-class classifier with 6 classes does well using only the entity type features, with an F1 score around 0.8. This shows that the performance of the binary filter is crucial for this task.

5 Discussion

In this section we analyze our models and discuss potential strategies to improve the results.

Table 5: Relation extraction model performance for 6 types of relations.

	Precision	Recall	F1
No Relations	0.985	0.911	0.946
Part-Whole	0.392	0.672	0.495
PHYS	0.244	0.562	0.341
ARTS	0.343	0.700	0.461
ORG-AFF	0.558	0.781	0.651
GEN-AFF	0.318	0.529	0.398
PER-SOC	0.332	0.857	0.479

Table 6: Impact of features on model performance.

Features	Precision	Recall	F1
Entity type + BOW	0.27	0.54	0.32
+Parse tree	0.39	0.69	0.48
+POS	0.39	0.69	0.48
+Contains overlap	0.45	0.71	0.54
+Wikipedia	0.45	0.72	0.54

5.1 Entity Extraction

The model performance is satisfactory but not as good as expected. The most important reason is the lack of labeled data. On the training data F1 is as high as 0.97 using only use the baseline features, indicating that our model can capture the characteristics of the training data. However, the distribution of the test data may be very different from training data, so thus our model overfits on training data and does not generalize well to the test set.

Another weakness of our model is the straightforward way to incorporate non-local features. We does not enforce constraints (e.g., same token should have same label) in inference or training, but simply use the predictions as features. A better way to consider the long-range dependency could be utilizing the CRF model proposed in Finkel et al. (2005). Joint learning with coreference and relations could be another option (Singh et al., 2013).

5.2 Coreference Resolution

There are a number of reasons for the poor performance of the model. Part of this is due to the structure of the problem. Heads of the entities rather than the extents may be a better classification baseline because extents may introduce noise into the coreference task, such as when heads and extents overlap and are treated as coreferent (e.g. *Luis Bongoyan* and *Luis Bongoyan, Vice Mayor of Davao City*). In addition, the model was trained on a slightly broader domain than the test set (the non-newswire and non broadcast news sections of ACE2005), so this could have negatively impacted performance.

The model does not capture certain kinds coreferent mentions with hypernym and hyponym relations and completely misses initialisms and acronyms, such as *EU* for *European Union*. Number and gender agreement may also be useful for capturing coreferent pronouns. Syntactic distance, rather than word or character distance, may be an informative coreference feature. To address this we could count breaks provided by a shallow parser to measure the distance between entities.

External knowledge would also be useful, such as bag of words from Wikipedia pages, but the generality of this approach is less certain since many entities are not in formal ontologies.

It may also be useful to formulate the categorization task as a clustering problem (Bengtson and Roth, 2008; Finkel and Manning, 2008).

5.3 Relation Extraction

The overall performance of the relation extraction model is satisfactory, but there is still room for improvement. First, Wikipedia features did not improve the performance as expected (Chan and Roth, 2010). Perhaps using a simplified version of the database causes a significant drop in precision by restricting relevant links between two entities. Second, although the binary filter for identifying relations has a high recall, the low precision makes it difficult for the multi-class classifier to generate good predictions. More features on the mention and dependency level (Kambhatla, 2004), collocation information and coreference information (Chan and Roth, 2010) could be used to further improve the performance of binary classifier.

6 Conclusion

We learned valuable lessons and gained lots of practical experience when working out different strategies for three NLP tasks: named entity recognition, coreference resolution and relation extraction. Exponential models are flexible and can be applied to all three tasks. In the future, our models could potentially be extended to integrate more features and training paradigms, such as joint training of all three models.

References

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. *EMNLP*, pages 294–303.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 28–36.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. *EMNLP*, pages 431–439.

Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. *NAACL-HLT*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. *4th International conference on language resources and evaluation*, pages 1–4.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. *EMNLP*, pages 1971–1982.

Jenny Rose Finkel and Christopher D Manning. 2008. Enforcing transitivity in coreference resolution. *ACL*, pages 45–48.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *ACL*, pages 363–370.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *Proceedings of ACLDemo 2004 - Interactive poster and demonstration sessions*.

Vijay Krishnan and Christopher D Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. *Proceedings of the 21st International Conference on Computational Linguistics*, pages 1121–1128.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Citeseer.

Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744.

Xiaoqiang Luo. 2007. Coreference or not: A twin model for coreference resolution. *NAACL-HLT*, pages 73–80.

James H Martin and Daniel Jurafsky. 2000. *Speech and Language Processing*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *NAACL*, pages 173–180.