Learning and Inference in Entity and Relation Identification

John Wieting
University of Illinois-Urbana Champaign
wieting2@illinois.edu

Abstract

In this study, I examine several different approaches to identifying entities and relations in sentences. I compare three different strategies to learn entities and relations. The first uses just local classifiers, the second uses local classifiers with integer linear programming (ILP) inference, and the third uses inference based training (IBT) and evaluates using ILP inference. My experiments indicate that in solving this particular problem, IBT performs the others, followed by local classifiers with (ILP) inference, and lastly the local classifiers by themselves. However these differences are not as large as one might expect.

1 Introduction

Often times solving real world machine learning problems involves predicting structured outputs. These outputs often consist of many parts which are not independent. Thus it is often prudent to try and make use of the dependencies that exist between these parts when solving these problems. In this paper, I explore three of the fundamentally different approaches to solving structured prediction. The first of these serves primarily as a baseline. In this method, dubbed L, local classifiers are trained separately. Then the One Versus All approach was used and so for each type of label, a different classifier was trained. In order to predict the complete output structure, each variable was determined separately without any knowledge of the dependencies or constraints on the structure. The second approach was similar to the first in that local classifiers were trained separately. The difference in this approach occurs in the inference step, where integer linear programming (ILP) is used to predict the best global structure subject to some constraints thus incorporating knowledge of the global structure into the prediction. In the last approach, the classifiers are trained together using a method known as inference based training or IBT. In this approach, all classifiers are trained simultaneously and inference feedback is used during training to promote or demote those classifiers that made mistakes.

The context for exploring these different paradigms is the Natural Language Processing problem known as entity and relation identification. This is a key task that is useful in many NLP systems like question-answering and information extraction. This task is an example of a structured prediction problem as the goal is to predict output structures which consist of dependent variables that adhere to some constraints. In this particular problem, we are given a sentence as input with the segments containing the entities pre-labeled. In other words, we assume that the segmentation task has been solved. Then our task is to label the entities in the sentence from a list of possibilities and similarly we must also predict the labels of the relations that exist between each of the entities. For an example of the input and output of this task see Figure 1, which was adopted from (Roth and Yih, 2007). One should note that the relations that are to be predicted are not symmetric, implying that every two entities in a sentence must have two relations that must be labeled. More specifically, if e is the

number of entities in a sentence, then we must label $2\binom{e}{2}$ relations. Thus for large sentences, a brute first search in labeling quickly becomes intractable since we must make l^{e^2} decisions if each entity can be given l labels and a sentence has e entities. Thus an efficient inference method such as integer linear programming must be used. In Section 2 of this paper, more information is provided on integer linear programming and the three learning schemes described above. Section 3 discusses the experiments and presents the results that were obtained, most notable of which is that on this task, IBT outperforms L+I which outperforms I. The results of these experiments are explained and are compared to related work in Section 4.

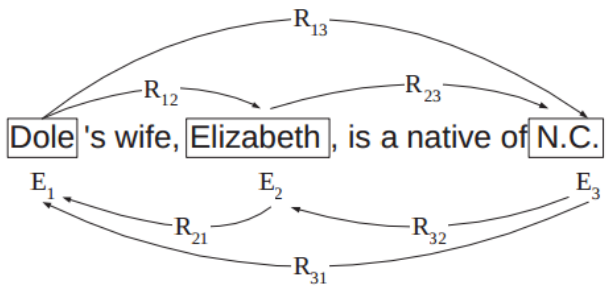

Figure 1. Entity-Relation Example.

2 Background

2.1 Integer Linear Programming

The method of inference used for the experiments in this paper is integer linear programming, similar to the approach in (Roth and Yih, 2007). In integer linear programming, we create an indicator variable for each possible assignment of every entity and relation in a sentence. Let E be the set of entities in an instance and L_E be the list of labels that entities can take. Similarly, let R be the set of relations in an instance and L_R be the list of the possible labels for relations. Then these indicator variables can be written for entities as $x_{e,le}$ where $e \in E$ and $le \in L_E$. For example, $x_{e_0,person}$ is an indicator variable that is 1 when the first entity in a sentence is a *person*. Similarly, indicator variables for relations can be defined as well as $x_{r,lr}$ where $r \in R$ and $lr \in L_E$. An example of this indicator variable could be $x_{r_{01},kill}$ which would be 1 if the relation between the first and second entity is *kill*. Using this formalism, the inference can be written as:

$$\max \sum_{e \in E} \sum_{le \in L_E} c_{e,l_e} x_{e,l_e} + \sum_{r \in R} \sum_{lr \in L_R} c_{r,l_r} x_{r,l_r}$$

subject to the following constraints:

$$\sum_{le \in L_E} x_{e,le} = 1 \; \forall e \in E$$

$$\sum_{lr \in L_R} x_{r,lr} = 1 \; \forall r \in R$$

$$2 \cdot x_{r_{ij},kill} \leq x_{e_i,person} + x_{e_j,person} \; \forall i,j \text{ and } i \neq j$$

$$2 \cdot x_{r_{ij},birthplace} \leq x_{e_i,person} + x_{e_j,location} \; \forall i,j \text{ and } i \neq j$$

$$x_{e,le} \in \{0,1\} \, \forall e \in E, \; \forall le \in L_E$$

$$x_{r,lr} \in \{0,1\} \, \forall r \in R, \; \forall lr \in L_R$$

In the equations above the $c_{i,j}$ correspond to the softmax output of the classifier that predicts label j on the entity or relation i. Upon examination the above equation is quite intuitive. We are trying to maximize the total confidence in our prediction in an effort to choose the most likely labeling for each component. The constraints are necessary in order to assure that the resulting structure is coherent. For instance if we are trying to label a relation *birthplace* we need to make sure that the first entity in this relation is a *person* and the second a *location*. The first constraint just assures that for each relation, the sum of its indicators sum to 1. In other words, each entity must belong to one and only one label. Similarly, the second constraint does the same for relations. The third and fourth constraints are the most interesting. In the particular invocation of the entity-relation identification problem in this paper, we are required to label the entities as *person*, *location*, or *unknown*; and the relations as *kill*, *birthplace*, and *other*. So the third constraint specifies that if we label a certain relation indicator variable as *kill* then the first and second entity must be labeled as *person*. The fourth constraint specifies that respective entities for the *birthplace* relation must be *person* and *location* respectively. These constraints are very intuitive and easy to verify. The

last two constraints just specify that the entity and relation indicator variables are binary.

Interestingly, if we changed the $c_{i,j}$ to be $logc_{i,j}$ this would be an approximation to the log-likelihood of the total structure. One might expect that this would focus the optimization towards the accuracy of the global structure from the accuracy of each of its components. This conjecture is explored in our experiments.

2.2 Algorithms

The first two paradigms, L and L+I, are fairly straightforward. In either case we train an Averaged Perceptron classifier for each label in an One Vs. All fashion. Sometimes in solving these types of problems, the classifiers have some interaction such as the output of one being used as a feature in another and then the output of that classifier is fed back into the original one. However, in these experiments these local classifiers are kept independent because one of our goals is to see how the performance in a structured problem such as entity and relation identification improves as these dependencies are included. In (Roth and Yih, 2007), this approach on this exact same problem is explored and the results indicate that pipelining these classifiers gives little to no improvement anyway. As has been mentioned before the way in which L and L+I differ is that L+I has an ILP inference step during evaluation. The last model, IBT, has more differences than these two models and in fact can be shown to be equivalent to Structured Perceptron (Collins, 2002). This algorithm is shown in Figure 2. To see that they are the same, one can notice that we can concatenate all of the weight vectors of the local classifiers together to obtain the global weight vector. The global feature vector is obtained for a given $\mathbf{x}$ and $\mathbf{y}$ by summing up the feature vectors for each component of the prediction and then concatenating them. So in this entity-relations identification task, we would would sum up the the features of every chosen entity and relation label and concatenate the resulting six vectors. Then if a mistake is made, we update the weight vector by adding the difference between the feature vector of the prediction and the gold structure. It is easy to see that this is the same as promoting the weights that should have been chosen and demoting those that should not have been chosen as long as the learning rate across all algorithms is uniform. Notice that this update rule is more conservative than that used in a One Vs. All scheme.

Collins Perceptron Algorithm
SETUP:
Input: Training Examples (x_i, y_i)
Output: Parameters $\mathbf{w}$
Repeat until convergence:
for each (x_i, y_i):
$y' = \text{argmax}_{y \in C(y^{n_y})} \mathbf{w} \cdot \phi(\mathbf{x}, \mathbf{y})$
if $y' \neq y$ then:
$\mathbf{w} = \mathbf{w} + \phi(\mathbf{x}, \mathbf{y}) - \phi(\mathbf{x}, \mathbf{y}')$

Figure 2. Collins Perceptron algorithm

3 Experiments

The data set used in this paper was the same data set used in (Roth and Yih, 2002). This is a rather small data set consisting of sentences from TREC documents which are mainly articles from newspapers such as the Wall Street Journal, Associated Press, etc. As input to the system, we were given a sentence and the segments containing the entities. This data set consisted of three different types of entities: *people*, *locations*, and *unknown*. There were also three different types of relations: *kill*, *birthplace*, and *other*. This data set consisted of 926 sentences where 245 contained the *kill* relation, 179 contained the *birthplace* relation, and 502 contained the *other* relation.

Each of the three paradigms: L, L+I, and IBT were evaluated on this data set. In each case, 5-fold cross validation was used to achieve accuracy and F1 scores which are included in the results below. These values were averaged over each of the 5 folds. The same random seed value was used when creating the folds in each paradigm so that as fair as possible of a comparison could be made.

Table 1 contains the accuracy and F1 scores for predicting entities. Similarly, Table 2 contains the accuracy and F1 scores for predicting relations. Lastly, Table 3 contains the average of the F1 scores over all 6 entities and relations and the global accuracy which measures the percentage of

complete sentence structures that were accurately predicted.

One additional experiment not included in these tables was to compare the results from using an objective function which uses the soft-max probabilities for each indicator variables against the results from using an objective function that used the log of these probabilities. The idea here is that that using the log may increase the global accuracy at the expense of the accuracy of each of the components. It turns out that, at least in this case, using logarithms in the objective function provides no change in L+I case and actually decreases both the average F1 and global accuracy in the IBT case to 0.877 and 0.592 respectively.

Algorithm	Average F1	Global Acc.
L	0.879	0.577
L+I	0.885	0.588
IBT	**0.888**	**0.598**

Table 3. Average F1 and Global Accuracy

4 Discussion

The results of these experiments compare well to previous work. This exact data set was used in (Roth and Yih, 2002). In this paper, the authors tried three approaches as well to modeling this problem. Their first and most basic approach used a learning system called SNoW (Roth, 1998). This system learned a network of linear functions using a winnow-like algorithm and then chose the labels based on which had the highest soft-max probabilities. This approach is very similar to the L approach in the system used in this paper except here, Averaged Perceptron was used. Their second model was a belief network. They constructed a network that represents the constraints between the relations and entities. Then they use the classifier from the basic approach to obtain the posterior probabilities. Inference on the resulting network was done using belief propagation in order to find the labeling that maximized the joint probability of all assignments for a particular instance. Their final approach, called omniscient, used the basic approach again except this time the true labels of the entities were given to the relation classifiers and the true labels of the relations are given to the entity classifiers. This is not a realistic scenario since the inputs to the classifiers must be predicted and are not given in this task, but this experiment is interesting as it gives some information as to how these classifiers are influenced by knowing these true labels. The results are shown in Table 5. Clearly the results of the experiments done in this paper are better - even just the L paradigm versus their belief network which makes use of the constraints of this problem. The reason for this is most likely feature choice, but also Averaged Perceptron may perform better on this task than the winnow-like algorithm in SNoW. The features included in their experiments were bigrams, trigrams, words, tags, ann words related to "kill" and "birth" from WordNet. Thus some of the best features from our models like Gazeteers and the distance between entities in a relation are missing in their models.

There are other works that try to solve the entity and relation identification problem using ILP for inference. These papers, (Roth and Yih, 2004) and (Roth and Yih, 2007), both use a different data set which consists of a larger number of sentences from TREC documents (1437 versus 926). They also try to predict labels for more entities (*person*, *location*, *organization*, *unknown*) and more relations (*located_in*, *work_for*, *orgBased_in*, *live_in*, *kill*). It is difficult to compare the results between our experiments and the works of these authors since the tasks are different. However, the results between our experiments and theirs are close enough that the comparison should be mentioned. The best F1 score for an entity that is achieved in their experiments is for *person* with an F1 of 0.904, and their best F1 score for a relation is for *kill* with an F1 score of 0.814. The results for our experiments beat these two entities and relations with scores of 0.918 and 0.818 respectively. It should be noted though that these comparisons are not on even ground as these works used a larger data set but had to predict more types of labels. This comparison is just meant to show that the system used in this paper has respectable results.

An ablation study was also done on the features used in these models. The features used in this study, influenced by (Roth and Small, 2008), are shown in Table 6 below. This table was constructed

Algorithm	Person				Location				Unknown			
	Acc	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
L	0.911	0.938	0.863	0.899	0.932	0.879	0.884	0.881	0.889	0.784	0.873	0.825
L+I	0.915	0.939	0.872	0.904	0.933	0.881	0.884	0.882	0.915	0.939	0.872	**0.904**
IBT	0.926	0.935	0.901	**0.918**	0.936	0.894	0.878	**0.886**	0.896	0.805	0.861	0.831

Table 1. Entity Prediction Results

Algorithm	Kill				Birthplace				Other			
	Acc	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
L	0.987	0.870	0.750	0.799	0.993	0.918	0.851	0.882	0.981	0.986	0.993	0.990
L+I	0.988	0.891	0.766	**0.818**	0.994	0.926	0.855	0.888	0.982	0.987	0.994	0.990
IBT	0.988	0.895	0.754	0.815	0.994	0.922	0.865	**0.890**	0.982	0.987	0.994	**0.991**

Table 2. Relation Prediction Results

Approach	Person			Location			Kill			Born-In		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Basic	0.870	0.921	**0.894**	0.811	0.832	0.820	0.786	0.438	0.550	0.729	0.690	0.705
BN	0.947	0.788	0.860	0.813	0.830	**0.821**	0.868	0.472	**0.607**	0.875	0.684	**0.766**
Omniscient	0.873	0.934	0.902	0.831	0.835	0.832	0.795	0.528	0.621	0.713	0.761	0.732

Table 4. Results using a Bayesian Network

by evaluating the features using the L+I paradigm with a fixed random seed over the cross validation. The first row represents the model using all features and then each row below shows the result when that particular feature is left out of the model. Clearly from the table, the Gazeteers are the most crucial as their absence leads to the largest decrease in performance. The numbers in italics in the table belong to features whose presence actually improved or caused no change in the model. This suggests that these features are irrelevant and in the perceptron algorithm irrelevant features decrease the margin and reduce generalization which likely explains the performance drop when these features are included. The second to last row in the table shows the results when all irrelevant features are removed. If you look closely, the accuracy and F1 score of the model are the same whether 2 Words After E is included or not. If we include this feature the results improve to be the best results of all as can be shown in the last row of the table. The poor performance of many of these ineffective features can likely be blamed on having too few examples. It is sensible to assume that with more data, many of these features will start to become more useful.

Feature	Average F1	Global Acc.
————	0.882	0.579
2 Words Before E	0.876	0.575
2 Words After E	*0.882*	*0.579*
2 Word Conj. Before E	*0.882*	*0.582*
2 Word Conj. After E	*0.882*	*0.587*
Gazeteers	**0.854**	**0.482**
2 POS Before E	*0.883*	*0.587*
2 POS After E	*0.883*	*0.587*
E Length	0.878	0.564
Words in E	0.862	0.507
Words in E Conj.	0.880	0.584
Dist. Between R	0.881	0.582
Words Between R	0.876	0.570
Wds. Fst. Lst.	*0.883*	*0.585*
Related Wds. R	*0.882*	*0.582*
Best (w/ Wds. After)	0.884	0.564
Best (w/o Wds. After)	**0.885**	**0.588**

Table 5. Ablation Study

Some of the features in the table above may need some explaining. The Gazeteer feature was calculated just by looking at each possible word sequence in an entity and checking if it was in a list of people's names or a list of locations. These lists were fairly small - less than 100 KB in size. We also tried including much larger lists obtained from Wikipedia that were on the order of

several MB, but the results did not improve at all. Another interesting feature is the Words Between R feature. This feature is just a bag of words type feature where each word between the two entities that are the arguments of a relation are included. Interestingly, this feature helps the model while a similar feature Related Wds. R hurts it. Related Wds. R is a feature that checks to see if any of the words between a relations's two argument entities is included in a list of synonyms of "born" and "kill" (including as well all forms of a word i.e. both assassinated and assassinate) which were obtained by using a thesaurus. When looking at the training data this makes sense as some words such as "fire" in the phrase "opened fire" are used several times in *kill* relations and these words are not included in the list of synonyms of "kill". Lastly the feature Wds. Fst. Lst. contained the words from the beginning of a sentence to the first entity of a relation if that relation contained the first entity of the sentence. It also contained the words from the last entity in the sentence to the end of the sentence if a relation contained the last entity. The idea behind this feature was to try and capture those relations whose keyword is not between its two entities. Often times this will occur in sentences where its included entities were either the first or last of the sentence.

One last interesting result of these experiments is the performance differences in the three paradigms L, L+I, and IBT. The results indicate the average F1 and global accuracy increase by about a half point and a point respectively as the model becomes more complex. In (Punyakanok et al., 2005), the authors claim and show that if the local classifiers are linearly separable, L+I outperforms IBT and if the task is globally separable, but not locally separable, IBT outperforms L+I, but only if there are a sufficient number of examples. The number of examples necessary is correlated with the degree of separability of the classifiers. The reason for this is that the global hypothesis space is much larger and so the global model is more expressive. However, a side effect of this is that its error bound is also larger since there are more hypotheses to choose from that will fit the training data. This is why it doesn't perform as well on easily separable data sets. This expressiveness helps though in the case when the data is not locally separable, but globally separable, since now an additional term, representing the expected error, must be added to the error bound of the local models that is not added in the global case. Thus if the global model sees enough examples it will start to separate the data in a way that the local models cannot and its performance will surpass that of the local models. Note that L+I was used instead of just L, since L+I should always outperform L with a sensible model.

In terms of our experiments, this suggests that the data is not linearly separable, which was already suspected by us before the experiments. However, it is reasonable to assume that if we had more examples than the 926 sentences in the data set, the performance of IBT will continue to increase and will do so at a faster rate than the L+I model. Interestingly though, (Punyakanok et al., 2005) also shows that as the the number of features increases so does the performance of the L and L+I models as these models become easier to learn. Thus if more expressive features can be uncovered, the local models could catch and surpass the IBT approach. Thus it seems that which model to use when solving a problem like entity-relation identification really depends on the number of training examples one has access to in addition to the expressiveness of one's features.

5 Conclusion

In this paper we studied the effectiveness of different paradigms for solving the entity and relation identification problem. These three approaches included learning just local classifiers using Averaged Perceptron (L), using local classifiers with integer linear programming inference (L+I), and using inference based training (IBT) where inference is done during training as well as testing. The results show that IBT outperforms L+I which outperforms L, although the performance differences between the methods is slight. As mentioned in (Punyakanok et al., 2005), this closeness in the results likely indicates that the local classifiers were not linearly separable as the global constraints and information did improve the results even with such a small data set. Also, comparing the results of

these experiments with related work illustrates, once again, the importance of choosing useful features. Lastly, these experiments indicate that respectable results can be achieved in this difficult task with rather simple models. This is important, as entity and relation identification is a key task in question-answering systems and information extraction systems.

Acknowledgments

Thanks Prof. Roth for your discussions on this project.

References

[Collins2002] M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

[Punyakanok et al.2005] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. 2005. Learning and inference over constrained output. In *IJCAI*, pages 1124–1129.

[Roth and Small2008] D. Roth and K. Small. 2008. Active learning for pipeline models. In *AAAI*, 7.

[Roth and Yih2002] D. Roth and W. Yih. 2002. Probabilistic reasoning for entity and relation recognition. In *COLING*, pages 835–841.

[Roth and Yih2004] D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *CoNLL*, pages 1–8. Association for Computational Linguistics.

[Roth and Yih2007] D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation.

[Roth1998] D. Roth. 1998. Learning to resolve natural language ambiguities: A unified approach. In *AAAI*, pages 806–813.