Nested Named Entity Recognition

Jenny Rose Finkel and Christopher D. Manning
Computer Science Department
Stanford University
Stanford, CA 94305
{jrfinkel|manning}@cs.stanford.edu

Abstract

Many named entities contain other named entities inside them. Despite this fact, the field of named entity recognition has almost entirely ignored nested named entity recognition, but due to technological, rather than ideological reasons. In this paper, we present a new technique for recognizing nested named entities, by using a discriminative constituency parser. To train the model, we transform each sentence into a tree, with constituents for each named entity (and no other syntactic structure). We present results on both newspaper and biomedical corpora which contain nested named entities. In three out of four sets of experiments, our model outperforms a standard semi-CRF on the more traditional top-level entities. At the same time, we improve the overall F-score by up to 30% over the flat model, which is unable to recover any nested entities.

1 Introduction

Named entity recognition is the task of finding entities, such as people and organizations, in text. Frequently, entities are nested within each other, such as *Bank of China* and *University of Washington*, both *organizations* with nested *locations*. Nested entities are also common in biomedical data, where different biological entities of interest are often composed of one another. In the GENIA corpus (Ohta et al., 2002), which is labeled with entity types such as *protein* and *DNA*, roughly 17% of entities are embedded within another entity. In the AnCora corpus of Spanish and Catalan newspaper text (Martí et al., 2007), nearly half of the entities are embedded. However, work on named entity recognition (NER) has almost entirely ignored nested entities and instead chosen to focus on the outermost entities.

We believe this has largely been for practical, not ideological, reasons. Most corpus designers have chosen to skirt the issue entirely, and have annotated only the topmost entities. The widely used CoNLL (Sang and Meulder, 2003), MUC-6, and MUC-7 NER corpora, composed of American and British newswire, are all flatly annotated. The GENIA corpus contains nested entities, but the JNLPBA 2004 shared task (Collier et al., 2004), which utilized the corpus, removed all embedded entities for the evaluation. To our knowledge, the only shared task which has included nested entities is the SemEval 2007 Task 9 (Márquez et al., 2007b), which used a subset of the AnCora corpus. However, in that task all entities corresponded to particular parts of speech or noun phrases in the provided syntactic structure, and no participant directly addressed the nested nature of the data.

Another reason for the lack of focus on nested NER is technological. The NER task arose in the context of the MUC workshops, as small chunks which could be identified by finite state models or gazetteers. This then led to the widespread use of sequence models, first hidden Markov models, then conditional Markov models (Borthwick, 1999), and, more recently, linear chain conditional random fields (CRFs) (Lafferty et al., 2001). All of these models suffer from an inability to model nested entities.

In this paper we present a novel solution to the problem of nested named entity recognition. Our model explicitly represents the nested structure, allowing entities to be influenced not just by the labels of the words surrounding them, as in a CRF, but also by the entities contained in them, and in which they are contained. We represent each sentence as a parse tree, with the words as leaves, and with phrases corresponding to each entity (and a node which joins the entire sentence). Our trees look just like syntactic constituency trees, such as those in the Penn TreeBank (Marcus et al., 1993),

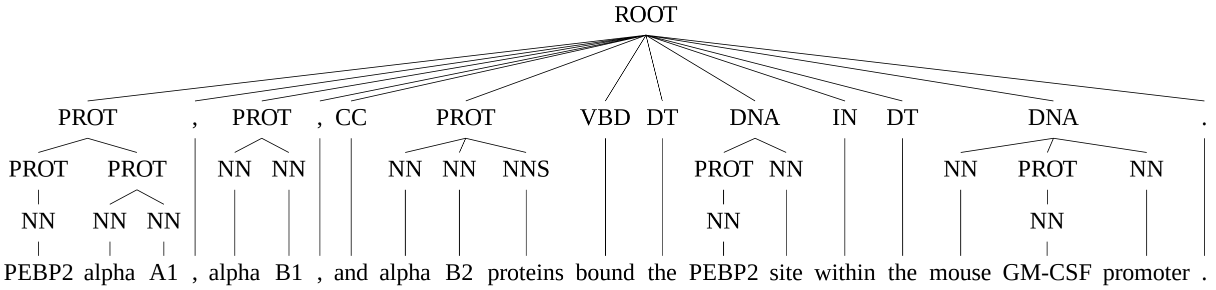
ROOT
PROT , PROT , CC PROT VBD DT DNA IN DT DNA .
PROT PROT NN NN NN NN NNS PROT NN NN PROT NN
NN NN NN NN NN
PEBP2 alpha A1 , alpha B1 , and alpha B2 proteins bound the PEBP2 site within the mouse GM-CSF promoter .

Figure 1: An example of our tree representation over nested named entities. The sentence is from the GENIA corpus. *PROT* is short for *PROTEIN*.

but they tend to be much flatter. This model allows us to include parts of speech in the tree, and therefore to jointly model the named entities and the part of speech tags. Once we have converted our sentences into parse trees, we train a discriminative constituency parser similar to that of (Finkel et al., 2008). We found that on top-level entities, our model does just as well as more conventional methods. When evaluating on *all* entities our model does well, with F-scores ranging from slightly worse than performance on top-level only, to substantially better than top-level only.

2 Related Work

There is a large body of work on named entity recognition, but very little of it addresses nested entities. Early work on the GENIA corpus (Kazama et al., 2002; Tsuruoka and Tsujii, 2003) only worked on the innermost entities. This was soon followed by several attempts at nested NER in GENIA (Shen et al., 2003; Zhang et al., 2004; Zhou et al., 2004) which built hidden Markov models over the innermost named entities, and then used a rule-based post-processing step to identify the named entities containing the innermost entities. Zhou (2006) used a more elaborate model for the innermost entities, but then used the same rule-based post-processing method on the output to identify non-innermost entities. Gu (2006) focused only on proteins and DNA, by building separate binary SVM classifiers for innermost and outermost entities for those two classes.

Several techniques for nested NER in GENIA where presented in (Alex et al., 2007). Their first approach was to layer CRFs, using the output of one as the input to the next. For inside-out layering, the first CRF would identify the innermost entities, the next layer would be over the words and the innermost entities to identify second-level entities, etc. For outside-in layering the first CRF would identify outermost entities, and then successive CRFs would identify increasingly nested entities. They also tried a cascaded approach, with separate CRFs for each entity type. The CRFs would be applied in a specified order, and then each CRF could utilize features derived from the output of previously applied CRFs. This technique has the problem that it cannot identify nested entities of the same type; this happens frequently in the data, such as the nested *proteins* at the beginning of the sentence in Figure 1. They also tried a joint labeling approach, where they trained a single CRF, but the label set was significantly expanded so that a single label would include all of the entities for a particular word. Their best results where from the cascaded approach.

Byrne (2007) took a different approach, on historical archive text. She modified the data by concatenating adjacent tokens (up to length six) into potential entities, and then labeled each concatenated string using the C&C tagger (Curran and Clark, 1999). When labeling a string, the "previous" string was the one-token-shorter string containing all but the last token of the current string. For single tokens the "previous" token was the longest concatenation starting one token earlier.

SemEval 2007 Task 9 (Márquez et al., 2007b) included a nested NER component, as well as noun sense disambiguation and semantic role labeling. However, the parts of speech and syntactic tree were given as part of the input, and named entities were specified as corresponding to noun phrases in the tree, or particular parts of speech. This restriction substantially changes the task. Two groups participated in the shared task, but only one (Márquez et al., 2007a) worked on the named entity component. They used a multi-label AdaBoost.MH algorithm, over phrases in the

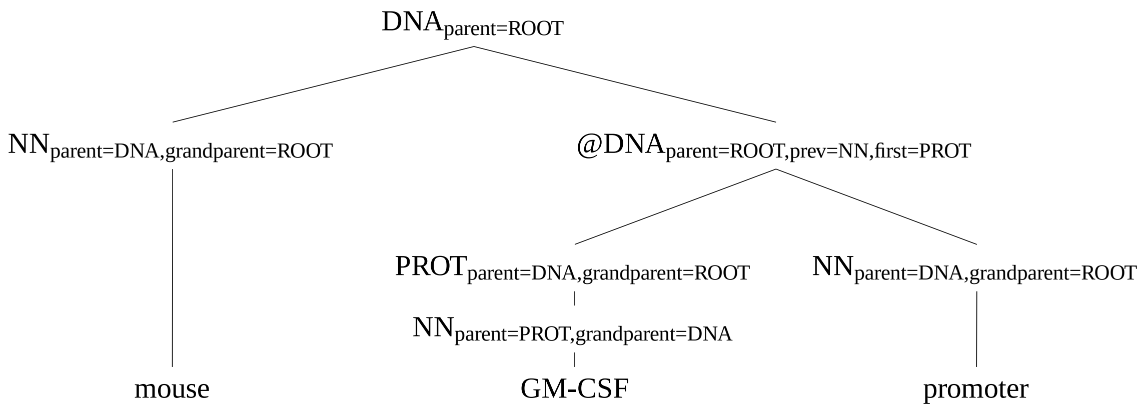
DNA parent=ROOT
NN parent=DNA,grandparent=ROOT
@DNA parent=ROOT,prev=NN,first=PROT
PROT parent=DNA,grandparent=ROOT
NN parent=DNA,grandparent=ROOT
NN parent=PROT,grandparent=DNA
mouse
GM-CSF
promoter

Figure 2: An example of a subtree after it has been annotated and binarized. Features are computed over this representation. An @ indicates a chart parser active state (incomplete constituent).

parse tree which, based on their labels, could potentially be entities.

Finally, McDonald et al. (2005) presented a technique for labeling potentially overlapping segments of text, based on a large margin, multilabel classification algorithm. Their method could be used for nested named entity recognition, but the experiments they performed were on joint (flat) NER and noun phrase chunking.

3 Nested Named Entity Recognition as Parsing

Our model is quite simple – we represent each sentence as a constituency tree, with each named entity corresponding to a phrase in the tree, along with a root node which connects the entire sentence. No additional syntactic structure is represented. We also model the parts of speech as preterminals, and the words themselves as the leaves. See Figure 1 for an example of a named entity tree. Each node is then annotated with both its parent and grandparent labels, which allows the model to learn how entities nest. We binarize our trees in a right-branching manner, and then build features over the labels, unary rules, and binary rules. We also use first-order horizontal Markovization, which allows us to retain some information about the previous node in the binarized rule. See Figure 2 for an example of an annotated and binarized subtree. Once each sentence has been converted into a tree, we train a discriminative constituency parser, based on (Finkel et al., 2008).

It is worth noting that if you use our model on data which does not have any nested entities, then it is precisely equivalent to a semi-CRF (Sarawagi and Cohen, 2004; Andrew, 2006), but with no length restriction on entities. Like a semi-CRF, we are able to define features over entire entities of arbitrary length, instead of just over a small, fixed window of words like a regular linear chain CRF.

We model part of speech tags jointly with the named entities, though the model also works without them. We determine the possible part of speech tags based on distributional similarity clusters. We used Alexander Clarke's software,[1] based on (Clark, 2003), to cluster the words, and then allow each word to be labeled with any part of speech tag seen in the data with any other word in the same cluster. Because the parts of speech are annotated with the parent (and grandparent) labels, they determine what, if any, entity types a word can be labeled with. Many words, such as verbs, cannot be labeled with any entities. We also limit our grammar based on the rules observed in the data. The rules whose children include part of speech tags restrict the possible pairs of adjacent tags. Interestingly, the restrictions imposed by this joint modeling (both observed word/tag pairs and observed rules) actually result in much faster inference (and therefore faster train and test times) than a model over named entities alone. This is different from most work on joint modeling of multiple levels of annotation, which usually results in significantly slower inference.

3.1 Discriminative Constituency Parsing

We train our nested NER model using the same technique as the discriminatively trained, conditional random field-based, CRF-CFG parser of (Finkel et al., 2008). The parser is similar to a

[1] http://www.cs.rhul.ac.uk/home/alexc/RHUL/Downloads.html

Local Features		**Pairwise Features**
label$_i$	distsim$_i$ + distsim$_{i-1}$ + label$_i$	label$_{i-1}$ + label$_i$
word$_i$ + label$_i$	shape$_i$ + shape$_{i+1}$ + label$_i$	word$_i$ + label$_{i-1}$ + label$_i$
word$_{i-1}$ + label$_i$	shape$_{i-1}$ + shape$_i$ + label$_i$	word$_{i-1}$ + label$_{i-1}$ + label$_i$
word$_{i+1}$ + label$_i$	word$_{i-1}$ + shape$_i$ + label$_i$	word$_{i+1}$ + label$_{i-1}$ + label$_i$
distsim$_i$ + label$_i$	shape$_i$ + word$_{i+1}$ + label$_i$	distsim$_i$ + label$_{i-1}$ + label$_i$
distsim$_{i-1}$ + label$_i$	words in a 5 word window	distsim$_{i-1}$ + label$_{i-1}$ + label$_i$
distsim$_{i+1}$ + label$_i$	prefixes up to length 6	distsim$_{i+1}$ + label$_{i-1}$ + label$_i$
shape$_i$ + label$_i$	suffixes up to length 6	distsim$_{i-1}$ + distsim$_i$ + label$_{i-1}$ + label$_i$
shape$_{i-1}$ + label$_i$		shape$_i$ + label$_{i-1}$ + label$_i$
shape$_{i+1}$ + label$_i$		shape$_{i-1}$ + label$_{i-1}$ + label$_i$
		shape$_{i+1}$ + label$_{i-1}$ + label$_i$
		shape$_{i-1}$ + shape$_i$ + label$_{i-1}$ + label$_i$
		shape$_{i-1}$ + shape$_{i+1}$ + label$_{i-1}$ + label$_i$

Table 1: The local and pairwise NER features used in all of our experiments. Consult the text for a full description of all features, which includes feature classes not in this table.

chart-based PCFG parser, except that instead of putting probabilities over rules, it puts *clique potentials* over local subtrees. These unnormalized potentials know what span (and split) the rule is over, and arbitrary features can be defined over the local subtree, the span/split and the words of the sentence. The inside-outside algorithm is run over the clique potentials to produce the partial derivatives and normalizing constant which are necessary for optimizing the log likelihood. Optimization is done by stochastic gradient descent.

The only real drawback to our model is runtime. The algorithm is $O(n^3)$ in sentence length. Training on all of GENIA took approximately 23 hours for the nested model and 16 hours for the semi-CRF. A semi-CRF *with* an entity length restriction, or a regular CRF, would both have been faster. At runtime, the nested model for GENIA tagged about 38 words per second, while the semi-CRF tagged 45 words per second. For comparison, a first-order linear chain CRF trained with similar features on the same data can tag about 4,000 words per second.

4 Features

When designing features, we first made ones similar to the features typically designed for a first-order CRF, and then added features which are not possible in a CRF, but are possible in our enhanced representation. This includes features over entire entities, features which directly model nested entities, and joint features over entities and parts of speech. When features are computed over each label, unary rule, and binary rule, the feature function is aware of the rule span and split.

Each word is labeled with its distributional similarity cluster (*distsim*), and a string indicating orthographic information (*shape*) (Finkel et al., 2005). Subscripts represent word position in the sentence. In addition to those below, we include features for each fully annotated label and rule.

Local named entity features. Local named entity features are over the label for a single word. They are equivalent to the local features in a linear chain CRF. However, unlike in a linear chain CRF, if a word belongs to multiple entities then the local features are computed for each entity. Local features are also computed for words not contained in any entity. Local features are in Table 1.

Pairwise named entity features. Pairwise features are over the labels for adjacent words, and are equivalent to the edge features in a linear chain CRF. They can occur when pairs of words have the same label, or over entity boundaries where the words have different labels. Like with the local features, if a pair of words are contained in, or straddle the border of, multiple entities, then the features are repeated for each. The pairwise features we use are shown in Table 1.

Embedded named entity features. Embedded named entity features occur in binary rules where one entity is the child of another entity. For our embedded features, we replicated the pairwise features, except that the embedded named entity was treated as one of the words, where the “word” (and other annotations) were indicative of the type of entity, and not the actual string that is the entity. For instance, in the subtree in Figure 2, we would compute *word*$_i$+*label*$_{i-1}$+*label*$_i$ as *PROT-DNA-DNA* for $i = 18$ (the index of the word *GM-CSF*). The normal pairwise feature at the same po-

GENIA – Testing on All Entities

		Nested NER Model			**Semi-CRF Model**		
	# Test	(train on all entities)			(train on top-level entities)		
	Entities	Precision	Recall	F_1	Precision	Recall	F_1
Protein	3034	79.04	69.22	73.80	78.63	64.04	70.59
DNA	1222	69.61	61.29	65.19	71.62	57.61	63.85
RNA	103	86.08	66.02	74.73	79.27	63.11	70.27
Cell Line	444	73.82	56.53	64.03	76.59	59.68	67.09
Cell Type	599	68.77	65.44	67.07	72.12	59.60	65.27
Overall	5402	75.39	**65.90**	**70.33**	**76.17**	61.72	68.19

Table 2: Named entity results on GENIA, evaluating on all entities.

GENIA – Testing on Top-level Entities Only

		Nested NER Model			**Semi-CRF Model**		
	# Test	(train on all entities)			(train on top-level entities)		
	Entities	Precision	Recall	F_1	Precision	Recall	F_1
Protein	2592	78.24	72.42	75.22	76.16	72.61	74.34
DNA	1129	70.40	64.66	67.41	71.21	62.00	66.29
RNA	103	86.08	66.02	74.73	79.27	63.11	70.27
Cell Line	420	75.54	58.81	66.13	76.59	63.10	69.19
Cell Type	537	69.36	70.39	69.87	71.11	65.55	68.22
Overall	4781	**75.22**	**69.02**	**71.99**	74.57	68.27	71.28

Table 3: Named entity results on GENIA, evaluating on only top-level entities.

sition would be *GM-CSF-DNA-DNA*.

Whole entity features. We had four whole entity features: the entire phrase; the preceding and following word; the preceding and following distributional similarity tags; and the preceding distributional similarity tag with the following word.

Local part of speech features. We used the same POS features as (Finkel et al., 2008).

Joint named entity and part of speech features. For the joint features we replicated the POS features, but included the parent of the POS, which either is the innermost entity type, or would indicate that the word is not in any entities.

5 Experiments

We performed two sets of experiments, the first set over biomedical data, and the second over Spanish and Catalan newspaper text. We designed our experiments to show that our model works just as well on outermost entities, the typical NER task, and also works well on nested entities.

5.1 GENIA Experiments

5.1.1 Data

We performed experiments on the GENIA v.3.02 corpus (Ohta et al., 2002). This corpus contains 2000 Medline abstracts (≈500k words), annotated with 36 different kinds of biological entities, and with parts of speech. Previous NER work using this corpus has employed 10-fold cross-validation for evaluation. We wanted to explore different model variations (e.g., level of Markovization, and different sets of distributional similarity clusterings) and feature sets, so we needed to set aside a development set. We split the data by putting the first 90% of sentences into the training set, and the remaining 10% into the test set. This is the exact same split used to evaluate part of speech tagging in (Tsuruoka et al., 2005). For development we used the first half of the data to train, and the next quarter of the data to test.[2] We made the same modifications to the label set as the organizers of the JNLPBA 2004 shared task (Collier et al., 2004). They collapsed all *DNA* subtypes into *DNA*; all *RNA* subtypes into *RNA*; all *protein* subtypes into *protein*; kept *cell line* and *cell type*; and removed all other entities. However, they also removed all embedded entities, while we kept them.

As discussed in Section 3, we annotated each word with a distributional similarity cluster. We used 200 clusters, trained using 200 million words from PubMed abstracts. During development, we found that fewer clusters resulted in slower infer-

[2] This split may seem strange: we had originally intended a 50/25/25 train/dev/test split, until we found the previously used 90/10 split.

JNLPBA 2004 – Testing on Top-level Entities Only

		Nested NER Model			**Semi-CRF Model**			**Zhou & Su (2004)**		
	# Test	(train on all entities)			(train on top-level entities)					
	Entities	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Protein	4944	66.98	74.58	70.57	68.15	62.68	65.30	69.01	79.24	73.77
DNA	1030	62.96	66.50	64.68	65.45	52.23	58.10	66.84	73.11	69.83
RNA	115	63.06	60.87	61.95	64.55	61.74	63.11	64.66	63.56	64.10
Cell line	487	49.92	60.78	54.81	49.61	52.16	50.85	53.85	65.80	59.23
Cell type	1858	75.12	65.34	69.89	73.29	55.81	63.37	78.06	72.41	75.13
Overall	8434	66.78	70.57	68.62	67.50	59.27	63.12	**69.42**	**75.99**	**72.55**

Table 4: Named entity results on the JNLPBA 2004 shared task data. Zhou and Su (2004) was the best system at the shared task, and is still state-of-the-art on the dataset.

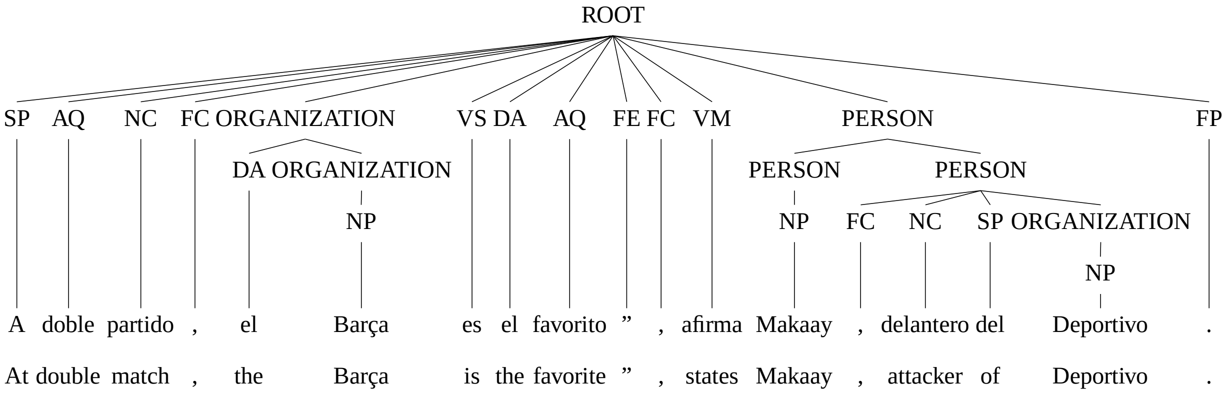
ROOT
SP
AQ
NC
FC
ORGANIZATION
VS
DA
AQ
FE
FC
VM
PERSON
FP
DA
ORGANIZATION
PERSON
PERSON
NP
NP
FC
NC
SP
ORGANIZATION
NP
A doble partido , el Barça es el favorito " , afirma Makaay , delantero del Deportivo .
At double match , the Barça is the favorite " , states Makaay , attacker of Deportivo .

Figure 3: An example sentence from the AnCora corpus, along with its English translation.

ence with no improvement in performance.

5.1.2 Experimental Setup

We ran several sets of experiments, varying between all entities, or just top-level entities, for training and testing. As discussed in Section 3, if we train on just top-level entities then the model is equivalent to a semi-CRF. Semi-CRFs are state-of-the-art and provide a good baseline for performance on just the top-level entities. Semi-CRFs are strictly better than regular, linear chain CRFs, because they can use all of the features and strucutre of a linear chain CRF, but also utilize whole-entity features (Andrew, 2006). We also evaluated the semi-CRF model on all entities. This may seem like an unfair evaluation, because the semi-CRF has no way of recovering the nested entities, but we wanted to illustrate just how much information is lost when using a flat representation.

5.1.3 Results

Our named entity results when evaluating on all entities are shown in Table 2 and when evaluating on only top-level entities are shown in Table 3. Our nested model outperforms the flat semi-CRF on both top-level entities and all entities.

While not our main focus, we also evaluated our models on parts of speech. The model trained on just top level entities achieved POS accuracy of 97.37%, and the one trained on all entities achieved 97.25% accuracy. The GENIA tagger (Tsuruoka et al., 2005) achieves 98.49% accuracy using the same train/test split.

5.1.4 Additional JNLPBA 2004 Experiments

Because we could not compare our results on the NER portion of the GENIA corpus with any other work, we also evaluated on the JNLPBA corpus. This corpus was used in a shared task for the BioNLP workshop at Coling in 2004 (Collier et al., 2004). They used the entire GENIA corpus for training, and modified the label set as discussed in Section 5.1.1. They also removed all embedded entities, and kept only the top-level ones. They then annotated new data for the test set. This dataset has no nested entities, but because the training data is GENIA we can still train our model on the data annotated with nested entities, and then evaluate on their test data by ignoring all embedded entities found by our named entity recognizer.

AnCora Spanish – Testing on All Entities

		Nested NER Model			**Semi-CRF Model**		
	# Test	(train on all entities)			(train on top-level entities)		
	Entities	Precision	Recall	F$_1$	Precision	Recall	F$_1$
Person	1778	65.29	78.91	71.45	75.10	32.73	45.59
Organization	2137	86.43	56.90	68.62	47.02	26.20	33.65
Location	1050	78.66	46.00	58.05	84.94	13.43	23.19
Date	568	87.13	83.45	85.25	79.43	29.23	42.73
Number	991	81.51	80.52	81.02	66.27	28.15	39.52
Other	512	17.90	64.65	28.04	10.77	16.60	13.07
Overall	7036	**62.38**	**66.87**	**64.55**	51.06	25.77	34.25

Table 5: Named entity results on the Spanish portion of AnCora, evaluating on all entities.

AnCora Spanish – Testing on Top-level Entities Only

		Nested NER Model			**Semi-CRF Model**		
	# Test	(train on all entities)			(train on top-level entities)		
	Entities	Precision	Recall	F$_1$	Precision	Recall	F$_1$
Person	1050	57.42	66.67	61.70	71.23	52.57	60.49
Organization	1060	77.38	40.66	53.31	44.33	49.81	46.91
Location	279	72.49	36.04	48.15	79.52	24.40	37.34
Date	290	72.29	57.59	64.11	71.77	51.72	60.12
Number	519	57.17	49.90	53.29	54.87	44.51	49.15
Other	541	11.30	38.35	17.46	9.51	26.88	14.04
Overall	3739	**50.57**	**49.72**	**50.14**	46.07	44.61	45.76

Table 6: Named entity results on the Spanish portion of AnCora, evaluating on only top-level entities.

This experiment allows us to show that our named entity recognizer works well on top-level entities, by comparing it with prior work. Our model also produces part of speech tags, but the test data is not annotated with POS tags, so we cannot show POS tagging results on this dataset.

One difficulty we had with the JNLPBA experiments was with tokenization. The version of GENIA distributed for the shared task is tokenized differently from the original GENIA corpus, but we needed to train on the original corpus as it is the only version with nested entities. We tried our best to retokenize the original corpus to match the distributed data, but did not have complete success. It is worth noting that the data is actually tokenized in a manner which allows a small amount of "cheating." Normally, hyphenated words, such as *LPS-induced*, are tokenized as one word. However, if the portion of the word before the hyphen is in an entity, and the part after is not, such as *BCR-induced*, then the word is split into two tokens: *BCR* and *-induced*. Therefore, when a word starts with a hyphen it is a strong indicator that the prior word and it span the right boundary of an entity. Because the train and test data for the shared task do not contain nested entities, fewer words are split in this manner than in the original data. We did not intentionally exploit this fact in our feature design, but it is probable that some of our orthographic features "learned" this fact anyway. This probably harmed our results overall, because some hyphenated words, which straddled boundaries in nested entities and would have been split in the original corpus (and were split in our training data), were not split in the test data, prohibiting our model from properly identifying them.

For this experiment, we retrained our model on the entire, retokenized, GENIA corpus. We also retrained the distributional similarity model on the retokenized data. Once again, we trained one model on the nested data, and one on just the top-level entities, so that we can compare performance of both models on the top-level entities. Our full results are shown in Table 4, along with the current state-of-the-art (Zhou and Su, 2004). Besides the tokenization issues harming our performance, Zhou and Su (2004) also employed clever post-processing to improve their results.

5.2 AnCora Experiments

5.2.1 Data

We performed experiments on the NER portion of AnCora (Martí et al., 2007). This corpus has Spanish and Catalan portions, and we evaluated on both. The data is also annotated with parts of speech, parse trees, semantic roles and word

AnCora Catalan – Testing on All Entities

	# Test	Nested NER Model (train all entities)			Semi-CRF Model (train top-level entities only)		
	Entities	Precision	Recall	F_1	Precision	Recall	F_1
Person	1303	89.01	50.35	64.31	70.08	46.20	55.69
Organization	1781	68.95	83.77	75.64	65.32	41.77	50.96
Location	1282	76.78	72.46	74.56	75.49	36.04	48.79
Date	606	84.27	81.35	82.79	70.87	38.94	50.27
Number	1128	86.55	83.87	85.19	75.74	38.74	51.26
Other	596	85.48	8.89	16.11	64.91	6.21	11.33
Overall	6696	**78.09**	**68.23**	**72.83**	70.39	37.60	49.02

Table 7: Named entity results on the Catalan portion of AnCora, evaluating on all entities.

AnCora Catalan – Testing on Top-level Entities Only

	# Test	Nested NER Model (train all entities)			Semi-CRF Model (train top-level entities only)		
	Entities	Precision	Recall	F_1	Precision	Recall	F_1
Person	801	67.44	47.32	55.61	62.63	67.17	64.82
Organization	899	52.21	74.86	61.52	57.68	73.08	64.47
Location	659	54.86	67.68	60.60	62.42	57.97	60.11
Date	296	62.54	66.55	64.48	59.46	66.89	62.96
Number	528	62.35	70.27	66.07	63.08	68.94	65.88
Other	342	49.12	8.19	14.04	45.61	7.60	13.03
Overall	3525	57.67	59.40	58.52	**60.53**	**61.42**	**60.97**

Table 8: Named entity results on the Catalan portion of AnCora, evaluating on only top-level entities.

senses. The corpus annotators made a distinction between *strong* and *weak* entities. They define *strong* named entities as “a word, a number, a date, or a string of words that refer to a single individual entity in the real world.” If a strong NE contains multiple words, it is collapsed into a single token. *Weak* named entities, “consist of a noun phrase, being it simple or complex” and must contain a *strong* entity. Figure 3 shows an example from the corpus with both strong and weak entities. The entity types present are *person*, *location*, *organization*, *date*, *number*, and *other*. Weak entities are very prevalent; 47.1% of entities are embedded.

For Spanish, files starting with 7–9 were the test set, 5–6 were the development test set, and the remainder were the development train set. For Catalan, files starting with 8–9 were the test set, 6–7 were the development test set, and the remainder were the development train set. For both, the development train and test sets were combined to form the final train set. We removed sentences longer than 80 words. Spanish has 15,591 training sentences, and Catalan has 14,906.

5.2.2 Experimental Setup

The parts of speech provided in the data include detailed morphological information, using a similar annotation scheme to the Prague TreeBank (Hana and Hanová, 2002). There are around 250 possible tags, and experiments on the development data with the full tagset where unsuccessful. We removed all but the first two characters of each POS tag, resulting in a set of 57 tags which more closely resembles that of the Penn TreeBank (Marcus et al., 1993). All reported results use our modified version of the POS tag set.

We took only the words as input, none of the extra annotations. For both languages we trained a 200 cluster distributional similarity model over the words in the corpus. We performed the same set of experiments on AnCora as we did on GENIA.

5.2.3 Results and Discussion

The full results for Spanish when testing on all entities are shown in Table 5, and for only top-level entities are shown in Table 6. For part of speech tagging, the nested model achieved 95.93% accuracy, compared with 95.60% for the flatly trained model. The full results for Catalan when testing on all entities are shown in Table 7, and for only top-level entities are shown in Table 8. POS tagging results were even closer on Catalan: 96.62% for the nested model, and 96.59% for the flat model.

It is not surprising that the models trained on all entities do significantly better than the flatly trained models when testing on all entities. The story is a little less clear when testing on just top-level entities. In this case, the nested model does 4.38% better than the flat model on the Spanish data, but 2.45% worse on the Catalan data. The overall picture is the same as for GENIA: modeling the nested entities does not, on average, reduce performance on the top-level entities, but a nested entity model does substantially better when evaluated on all entities.

6 Conclusions

We presented a discriminative parsing-based method for nested named entity recognition, which does well on both top-level and nested entities. The only real drawback to our method is that it is slower than common flat techniques. While most NER corpus designers have defenestrated embedded entities, we hope that in the future this will not continue, as large amounts of information are lost due to this design decision.

Acknowledgements

Thanks to Mihai Surdeanu for help with the AnCora data. The first author was supported by a Stanford Graduate Fellowship. This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *BioNLP Workshop at ACL 2007*, pages 65–72.

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University.

Kate Byrne. 2007. Nested named entity recognition in historical archive text. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 589–596.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth Annual Meeting of the European Association for Computational Linguistics (EACL)*, pages 59–66.

Nigel Collier, J. Kim, Y. Tateisi, T. Ohta, and Y. Tsuruoka, editors. 2004. *Proceedings of the International Joint Workshop on NLP in Biomedicine and its Applications*.

J. R. Curran and S. Clark. 1999. Language independent NER using a maximum entropy tagger. In *CoNLL 1999*, pages 164–167.

Jenny Finkel, Shipra Dingare, Christopher Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the boundaries: Gene and protein identification in biomedical text. In *BMC Bioinformatics 6 (Suppl. 1)*.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based conditional random field parsing. In *ACL/HLT-2008*.

Baohua Gu. 2006. Recognizing nested named entities in GENIA corpus. In *BioNLP Workshop at HLT-NAACL 2006*, pages 112–113.

Jiří Hana and Hana Hanová. 2002. Manual for morphological annotation. Technical Report TR-2002-14, UK MFF CKL.

Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. 2002. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain (ACL 2002)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

L. Márquez, L. Padrè, M. Surdeanu, and L. Villarejo. 2007a. UPC: Experiments with joint learning within semeval task 9. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*.

L. Márquez, L. Villarejo, M.A. Martí, and M. Taulè. 2007b. Semeval-2007 task 09: Multilevel semantic annotation of Catalan and Spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*.

M.A. Martí, M. Taulè, M. Bertran, and L. Márquez. 2007. Ancora: Multilingual and multilevel annotated corpora. MS, Universitat de Barcelona.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 987–994.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86.

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

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*. Association for Computational Linguistics (ACL 2003).

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2003. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL-03 Workshop on Natural Language Processing in Biomedicine*, pages 41–48.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746*, pages 382–392.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411–422.

GuoDong Zhou and Jian Su. 2004. Exploring deep knowledge resources in biomedical name recognition. In *Joint Workshop on Natural Language Processing in Biomedicine and Its Applications at Coling 2004*.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.

Guodong Zhou. 2006. Recognizing names in biomedical texts using mutual information independence model and SVM plus sigmoid. *International Journal of Medical Informatics*, 75:456–467.