Joint Mention Extraction and Classification with Mention Hypergraphs

Wei Lu
Singapore University
of Technology and Design
luwei@sutd.edu.sg

Dan Roth
University of Illinois
at Urbana-Champaign
danr@illinois.edu

Abstract

We present a novel model for the task of joint mention extraction and classification. Unlike existing approaches, our model is able to effectively capture overlapping mentions with unbounded lengths. The model is highly scalable, with a time complexity that is linear in the number of words in the input sentence and linear in the number of possible mention classes. Our model can be extended to additionally capture mention heads explicitly in a joint manner under the same time complexity. We demonstrate the effectiveness of our model through extensive experiments on standard datasets.

1 Introduction

One of the essential goals in natural language processing (NLP) is the development of effective systems that can capture the underlying semantics conveyed by human languages. An important step towards such a goal is the development of practical systems that can efficiently extract useful shallow semantic information such as entities and at the same time identify their semantic classes (*e.g.*, person, organization, etc).

Such a task is often known as named entity recognition and classification (NERC), one of the standard tasks in information extraction (IE). While such a task focuses on the extraction and classification of entities in the texts which are named, recently researchers also showed interest in a closely related task – *mention extraction and classification/typing*. Unlike a named entity, a mention is typically defined as a reference to an entity in natural language text that can be either named, nominal or pronominal (Florian et al., 2004). The task of mention detection and tracking has received substantial attention, largely due to its important role in conducting several downstream tasks, such as *relation extraction* (Mintz et al., 2009), *entity linking* (Guo et al., 2013), and *coreference resolution* (Chang et al., 2013).

While most existing work on named entity recognition and mention extraction and classification have been effective, there remain several key limitations associated with existing models. In fact, one can view these problems as instances of the more general problem of *semantic tagging* – the task of assigning appropriate semantic tags to certain text spans for a given input sentence. Unlike part-of-speech (POS) tagging, which has been extensively studied in the past few decades by the community, such a semantic tagging task presents several additional new challenges. First, a mention can consist of multiple words, so its length can be arbitrarily long. Second, the mentions can overlap with one another. Popular models used for POS tagging, such as linear-chain conditional random fields (Lafferty et al., 2001) or semi-Markov conditional random fields (Sarawagi and Cohen, 2004) have difficulties coping with these issues. While approaches on addressing these issues exist, current algorithms typically suffer from high time complexity (Finkel and Manning, 2009) and are therefore difficult to scale to large datasets. On the other hand, the problem of designing efficient and scalable models for mention extraction and classification from natural language texts becomes increasingly important in this era where a large volume of textual data is becoming available on the Web every day – users need systems which are able to scale to extremely large datasets to support efficient semantic analysis for timely decison-making.

In this paper, we tackle the above-mentioned issue by introducing a novel model for joint mention extraction and classification. We make the following major contributions in this work:

- We propose a model that is able to effectively

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 857–867,
Lisbon, Portugal, 17-21 September 2015.

handle overlapping mentions with unbounded lengths.

- The learning and inference algorithms of our proposed model have a time complexity that is linear in the number of words in the input sentence and also linear in the number of possible semantic classes/types, making our model scalable to extremely large datasets.
- Our model can additionally capture mentions' head information in a joint manner under the same time complexity.

Our system and code are available for download from `http://statnlp.org/research/ie/`.

2 Related Work

Existing work has been largely focused on the task of named entity recognition and classification (NERC). The survey of (Nadeau and Sekine, 2007) is a comprehensive study of this topic.

Most prior work took a supervised learning approach. Zhou and Su (2002) presented a system for recognizing named entities using an HMM-based approach. Florian et al. (2003) presented a system for named entity recognition by combining different classifiers. McDonald and Pereira (2005) used conditional random fields for extracting gene and protein mentions from biomedical texts. Ratinov and Roth (2009) presented a systematic analysis over several issues related to the design of a named entity recognition and classification system where issues such as chunk representations and the choice of inference algorithms were discussed. Researchers also looked into semi-supervised and unsupervised approaches for such a task (Cucchiarelli and Velardi, 2001; Etzioni et al., 2005). Additional efforts on addressing the NERC problem under a multilingual or cross lingual setting also exist (Florian et al., 2004; Che et al., 2013; Wang et al., 2013).

As pointed out by Finkel and Manning (2009), named entities are often nested. This fact was often ignored by the community largely due to technical reasons. They therefore proposed to use a constituency parser with a $O(n^3)$ time complexity (n is the number of words in the input sentence) to handle nested entities, and showed its effectiveness across several datasets. Alex et al. (2007) also presented several approaches by building models on top of linear-chain conditional random fields for recognizing nested entities in biomedical texts. Hoffmann et al. (2011) looked into a separate issue, which is to identify overlapping *relations* amongst entities.

Named entity recognition and classification still remains a popular topic in the field of statistical natural language processing. Ritter et al. (2011) looked into recognizing entities from social media data that involves informal and potentially noisy texts. Pasupat and Liang (2014) looked into the issue of zero-shot entity extraction from Web pages with natural language queries where minimal supervision was used. Neelakantan and Collins (2014) looked into the problem of automatically constructing dictionaries with minimal supervision for improved named entity extraction. Li and Ji (2014) presented an approach to perform the task of extraction of mentions and their relations in a joint and incremental manner.

3 Approach

3.1 Mentions and Their Combinations

Typically, a mention that appears in a natural language sentence consists of a contiguous sequence of natural language words. Consider a sentence that consists of n words where each word is indexed with its position in the sentence. A mention m can be uniquely represented with a tuple $\langle b_m, e_m, \tau \rangle$, where b_m and e_m are the indices of the first and last word of the mention, respectively, and τ is its semantic class (type).

We can see that for a given sentence consisting of n words, there are altogether $tn(n+1)/2$ possible different mention candidates, where t is the total number of possible mention types. Now, for each such candidate in the given sentence, it can be either a mention, or not a mention. This leads to a total number of $2^{tn(n+1)/2}$ possible mention combinations. This number is prohibitively large even for small values of n and t, which prevents us from exhaustively enumerating all of them during learning and inference.

One approach to performing inference over such a large space is to introduce compact representations that are able to encode exponentially many mentions that would enable tractable inference algorithms to be employed. We discuss in the next section our novel mention hypergraph representation proposed for such a purpose.

3.2 Mention Hypergraphs

Central to our approach is the introduction of the novel *mention hypergraphs* that allow us to

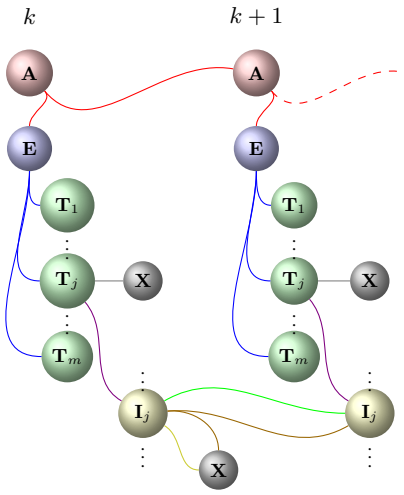

Figure 1: The (partial) hypergraph for representing all possible combinations of mention occurrences. Links that belong to the same hyperedge are highlighted with the same color, and different hyperedges are highlighted with different colors, *e.g.*, the green link that connects two **I** nodes forms a single hyperedge, while the two brown links that connect two **I** nodes and one **X** node form a separate single hyperedge.

compactly represent exponentially many possible combinations of potentially overlapping, length-unbounded mentions of different types.

A *hypergraph* is a generalization of a conventional graph, whose edges (*a.k.a. hyperedges*) can connect two or more nodes. In this work, we consider a special class of hypergraphs, where each hyperedge consists of a designated parent node and an *ordered* list of child nodes. Hypergraphs have also been used in other fields, such as syntactic parsing (Klein and Manning, 2001), semantic parsing (Lu, 2015) and machine translation (Cmejrek et al., 2013).

Our mention hypergraphs consist of five types of nodes which are used to compactly represent many mentions of different semantic types and boundaries, namely, **A** nodes, **E** nodes, **T** nodes, **I** nodes, and **X** nodes. A partial mention hypergraph is depicted in Figure 1. We describe the definition of each type of nodes next.

- **A nodes.** These nodes are used to sequentially arrange mentions with different left boundaries. Specifically, each **A** node at position k (the k-th word), or $\mathbf{A}^k$, is used to compactly represent all such mentions in the sentence whose left boundaries are *exactly at or strictly after* k.
- **E nodes.** The node $\mathbf{E}^k$ is used to compactly represent all possible mentions (possibly of length zero) whose left boundaries are *exactly at* the current position k.
- **T nodes.** The node $\mathbf{T}^k_j$ is used to compactly represent all mentions (possibly of length zero) whose left boundaries are *exactly at* position k, and *have the mention type* j.
- **I nodes.** The node $\mathbf{I}^k_j$ is used to compactly represent all *incomplete* mentions which contain the current word at position k as part of the mention, and have the mention type j.
- **X nodes.** These are the "terminal" nodes indicating the completion of a path. No additional node will be attached to such nodes as a child.

There are also various hyperedges that connect different nodes in the mention hypergraph. We use $\langle \alpha \leftarrow \beta_1, \ldots, \beta_n \rangle$ to denote a hyperedge which connects a parent node α and child nodes $\beta_1, \ldots, \beta_n$. Each hyperedge essentially provides one possible way of re-expressing the semantics conveyed by the parent node using the child nodes. For example, as shown in Figure 1, the hyperedge connecting the parent node $\mathbf{A}^k$ and the child nodes $\mathbf{E}^k, \mathbf{A}^{k+1}$ explains the fact that any mention covered by $\mathbf{A}^k$ either has a left boundary that is "*exactly at* k" ($\mathbf{E}^k$), or "*exactly at* or *strictly after* $k+1$" ($\mathbf{A}^{k+1}$).

Similarly, for each **I** node, there exist 3 hyperedges that connect it to other child nodes. The top hyperedge (in green) encodes the fact that the current word appears in the middle of a mention; the bottom hyperedge (in yellow) encodes the fact that the current word appears in a mention as the last word; the middle hyperedge (in brown) encodes the fact that both cases can occur at the same time (*i.e.*, the current word belongs to multiple overlapping mentions of the same type). We have the following theorem:

Theorem 3.1 *Any combination of mentions in a sentence can be represented with exactly one sub-hypergraph of the complete mention hypergraph.*

Proof For each mention, there exists a unique path in the mention hypergraph to represent it. For any combination of mentions, there exist unique paths in the mention hypergraph to represent such a combination. These paths altogether form a unique sub-hypergraph of the original hypergraph.

For example, consider the following sentence: "*he also talked with the egyptian president* ." This sentence contains three mentions. The first is "*he*" with type `PER`, the second is "*the egyptian president*" with type `PER`, and the third mention is "*egyptian*" with type `GPE`. Figure 2 gives the sub-hypergraph structure showing how these mentions

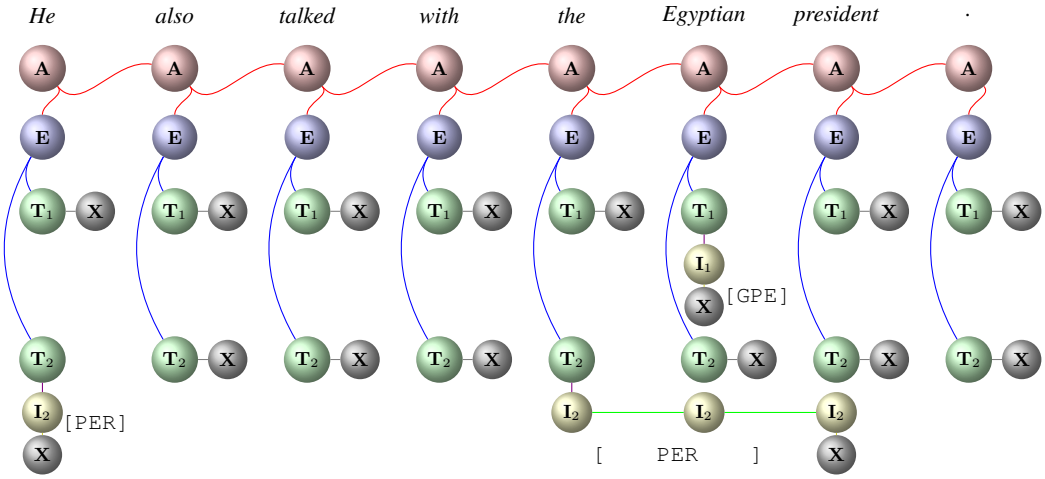

Figure 2: An example sub-hypergraph structure for jointly representing all the three mentions that appear in the sentence "*He also talked with the Egyptian president* ." For simplicity and the ease of illustration, we assume there are only two possible mention types: PER and GPE.

are jointly represented. The mention hypergraph defined over the input sentence contains exponentially many such sub-hypergraph structures.

We note that the converse of Theorem 3.1 is not true. In certain cases, it is possible for two different overlapping mention combinations to share the same mention hypergraph.

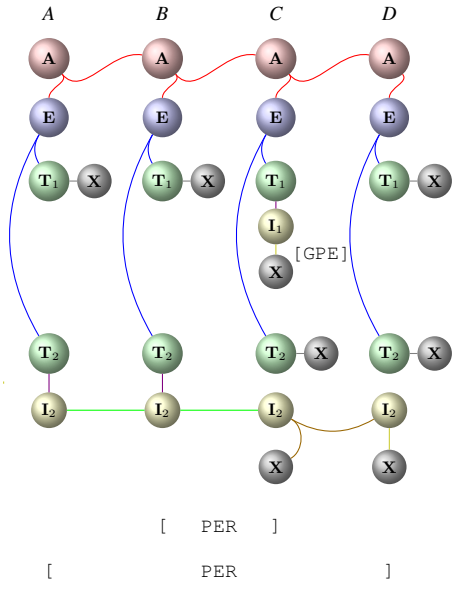

Figure 3: An example illustrating the converse of Theorem 3.1 is not true.

For example, consider a toy example sentence *A B C D* shown in Figure 3, both *B C* and *A B C D* are mentions of the same type PER (*i.e.* one is strictly contained by the other. We call such combinations type-I combinations). The above sub-hypergraph shows how to encode such a combination. However, if both *A B C* and *B C D* are mentions of the same type PER (*i.e.*, two mentions overlap but no one is contained by the other. We call such combinations type-II combinations), such a combination shares the same representation as the above sub-hypergraph. Note that such an ambiguity happens only when two overlapping mentions have the same type, and one mention is strictly contained by the other and their boundaries are all different. In practice, however, we found that in the two datasets that we used for evaluations, if two mentions overlap with one another, they almost always form a type-I combination, and type-II combinations are very rare. Empirically, as we will see later in our experiments, our model is effective in handling overlapping mentions.

3.3 Log-Linear Modeling

Following the conditional random fields (Lafferty et al., 2001), we adopted a log-linear approach for such a joint mention extraction and typing task. Specifically, for a given input sentence $\mathbf{x}$, the probability of predicting a possible output $\mathbf{y}$ (a mention sub-hypergraph that represents a particular combination of mentions) is given as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(\mathbf{w}^T\mathbf{f}(\mathbf{x},\mathbf{y}))}{\sum_{\mathbf{y}'}\exp(\mathbf{w}^T\mathbf{f}(\mathbf{x},\mathbf{y}'))} \quad (1)$$

where $\mathbf{f}(\mathbf{x},\mathbf{y})$ is the feature vector defined over the input-output pair $(\mathbf{x},\mathbf{y})$, and the weight vector $\mathbf{w}$ gives the parameters of the model.

Our objective is to minimize the regularized negative joint log-likelihood of the dataset:

$$\mathcal{L}(\mathbf{w}) = \sum_i \log \sum_{\mathbf{y}'} \exp(\mathbf{w}^T\mathbf{f}(\mathbf{x_i},\mathbf{y}')) - \sum_i \mathbf{w}^T\mathbf{f}(\mathbf{x_i},\mathbf{y_i}) + \lambda\mathbf{w}^T\mathbf{w} \quad (2)$$

where $(\mathbf{x_i}, \mathbf{y_i})$ refers to the i-th training instance, and the last term is a L_2 regularization term with λ being a positive scalar (fixed to 0.01 in this work).

The gradient of the above objective function is:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_k} = \sum_i \mathbf{E}_{p(\mathbf{y}'|\mathbf{x_i})}[f_k(\mathbf{x_i}, \mathbf{y}')] - \sum_i f_k(\mathbf{x_i}, \mathbf{y_i}) + 2\lambda w_k \quad (3)$$

where w_k is the weight of the k-th feature f_k.

We note that unlike many recent latent-variable approaches to structured prediction (Petrov and Klein, 2007; Blunsom et al., 2008), we are able to represent each of our outputs $\mathbf{y}$ with a single fully-observed structure. Thus, our objective function essentially defines a standard regularized softmax regression model, and is therefore convex (Boyd and Vandenberghe, 2004), where a global optimum can be found.

The objective function defined in Equation 2 can be optimized with standard gradient-based methods. We used L-BFGS (Liu and Nocedal, 1989) as our optimization method.

3.4 Algorithms

In order to solve the optimization problem described above, one needs to compute the values of the gradient scores in Equation 3. Computation of the second and third terms in this equation is straightforward. The first term in Equation 3 involves the computation of an expectation of feature values over all possible mention combinations for a given input sentence. Following classic dynamic programming algorithms used in graphical models, we develop analogous efficient dynamic programming algorithms that work on hypergraphs and generalize the conventional forward-backward/inside-outside algorithm to efficiently compute such values.

Time Complexity At each time step k, we need to compute scores for m $\mathbf{I}$ nodes, m $\mathbf{T}$ nodes, 1 $\mathbf{E}$ node, and 1 $\mathbf{A}$ node. Hence, the overall time complexity for our algorithm is in $O(mn)$ (assuming computation of the feature scores at each node involves a constant time), where m is the total number of possible mention types, and n is the total number of words in the given sentence. [1]

[1] Note that the time complexity for the linear chain CRF is in $O(m^2 n)$ due to their first-order assumption.

3.5 Features

The features that we use are inspired by the work of (Carreras et al., 2002). Specifically, we consider the following features defined over the inputs:

- Words (and POS tags, if available) that appear around the current word (with position information), with a window of size 3.
- Word n-grams (and POS n-grams, if available) that contain the current word (with position information), for $n = 2, 3, 4$.
- Bag of words around the current word, with a window of size 5.
- Word pattern features [2].

Note that these are the indicator functions defined over the inputs. The final set of features are defined over $(\mathbf{x}, \mathbf{y})$ tuples, which is obtained as a cross-product between the above indicator functions and the following indicator function:

- The type of the node (such as $\mathbf{T}$ or $\mathbf{I}$).

In addition, we also introduce the following feature defined over the output structure only:

- The number of such hyperedges that exactly connect one $\mathbf{T}$ node and one $\mathbf{I}$ node.

We call this feature *mention penalty*. This feature learns a global preference of the number of mentions that should appear in any input sentence.

3.6 Joint Modeling of Mention Heads

One additional assumption for the mention extraction and typing task is that each mention comes with a *head*. A head is strictly a substring of the mention and provides important information about the mention. It is possible to extend our model to support joint modeling of mention heads, while still maintaining the same time complexity.

Due to space limitations, we could only give a relatively brief description of this extension in this section. The idea is to replace the $\mathbf{I}$ nodes with three different types of nodes, namely $\mathbf{I}_j$–$\mathbf{B}$ nodes (used to represent words that appear within a mention of type j and *before* its head), $\mathbf{I}_j$–$\mathbf{W}$ nodes (used to represent words that appear *within* the head of a mention of type j), and $\mathbf{I}_j$–$\mathbf{A}$ nodes (used to represent words that appear within a mention of type j and *after* its head). The hyperedges also need to be established accordingly in order to properly model all possible mention and head

[2] *all-caps, all-digits, all-alphanumeric, contains-digits, contains-dots, contains-hyphen, initial-caps, lonely-initial, punctuaion-mark, roman-number, single-character, URL.*

	ACE2004			ACE2005		
	TRAIN	DEV	TEST	TRAIN	DEV	TEST
Documents	356	41	46	370	43	51
Sentences	6,799	829	879	7,336	958	1,047
with o.l. mentions	2,683 (39%)	293 (35%)	373 (42%)	2,683 (37%)	340 (35%)	330 (32%)
Mentions	22,207	2,511	3,031	24,687	3,217	3,027
length > 6	1,439 (6%)	179 (7%)	199 (7%)	1,343 (5%)	148 (5%)	160 (6%)
max length	57	35	43	49	30	27

Table 1: Corpora statistics for the ACE2004 and ACE2005 datasets.

	ACE2004						ACE2005					
	DEV			TEST			DEV			TEST		
	P	R	F	P	R	F	P	R	F	P	R	F
CRF (BIO)	69.6	42.8	53.0	70.0	40.3	51.2	69.8	45.3	55.0	67.6	43.7	53.1
CRF (BILOU)	70.7	42.6	53.1	71.8	40.8	52.1	71.1	45.5	55.5	69.5	44.5	54.2
CRF (CC)	77.9	49.1	60.2	78.4	46.4	58.3	76.8	52.0	62.0	74.8	49.1	59.3
Semi-CRF (c=6)	75.4	44.1	55.6	76.1	41.4	53.6	75.3	48.5	59.0	72.8	45.0	55.6
Semi-CRF (c=∞)	66.8	44.8	53.7	66.7	42.0	51.5	69.7	48.9	57.5	67.5	46.1	54.8
MH	**79.6**	50.0	61.4	**79.2**	46.8	58.9	**79.3**	50.6	61.8	**76.9**	47.7	58.9
MH (F)	70.0	**59.2**	**63.8**	70.0	**56.9**	**62.8**	67.5	**61.8**	**64.5**	66.3	**59.2**	**62.5**

Table 2: Results on ACE2004 and ACE2005. The last two rows give the results of this work.

combinations. Since in such a new hypergraph, at each time step, only a constant number (2) of additional nodes are involved, the time complexity for learning and inference with such a model remains the same, which is in $O(mn)$.

3.7 Optimization of F measure

One standard evaluation metric for named entity recognition is the F (F_1) measure. In our task, the F measure is defined as the harmonic mean of the precision (P) and recall (R) scores, where precision is the ratio between the number of correctly predicted mentions and the total number of predicted mentions, and recall is the ratio between the number of correctly predicted mentions and the total number of gold mentions. We will also adopt these metrics in our evaluations later. Unfortunately, the model only optimizes its objective function defined in Equation 2, which is the negative (regularized) joint log-likelihood. Previous work showed it was possible to optimize the F measure in a log-linear model (Suzuki et al., 2006). Culotta and McCallum (2004) also proposed a method for optimizing information extraction performance based on confidence estimation. Their work is based on linear-chain CRF and estimate the confidence of extracted fields based on marginal probabilities. The technique is not directly applicable to our task where a hypergraph representation is used to encode overlapping mentions. In this work, we used a very simple and intuitive technique for optimizing the F measure. The idea is to further tune the weight of a single parameter – *mention penalty* based on the development set, after the training process completes. This is based on the observation that by increasing the value of the mention penalty, we are essentially forcing our model to predict more mentions. Therefore the recall is a monotonic function with respect to the mention penalty. Based on this fact, we use a simple search algorithm with a fixed step size (we set it to 0.01) to determine the optimal value of the modified mention penalty so that the F measure of the development set is optimized.

4 Experiments

In this section, we present empirical evaluations. Our main experiments were conducted on the standard ACE2004 and ACE2005 datasets which contain overlapping mentions. Two additional experiments on the GENIA and CONLL2003 dataset were also conducted .

4.1 Results on ACE

Our primary experiments were conducted based on the English portion of the ACE2004 dataset[3] and the ACE2005 dataset[4]. Following previous work, for ACE2004, we considered all documents from *arabic_treebank, bnews, chinese_treebank*, and *nwire*, and for ACE2005, we considered all documents from *bc, bn, nw*, and *wl* . We randomly split the documents for each dataset into three portions: 80% for training, 10% for development, and the remaining 10% for evaluations. The statistics of the datasets are summarized in Table 1[5]. We

[3]https://catalog.ldc.upenn.edu/LDC2005T09

[4]https://catalog.ldc.upenn.edu/LDC2006T06

[5]Exact train/dev/test splits information can be found on http://statnlp.org/research/ie/.

can observe that overlapping mentions are common – over 30% of the sentences contain overlapping mentions (see row 3 of the table). Mentions can also be very long – over 5% of the mentions consist of more than 6 words, and the longest mention consists of 57 words.

We compared our system's performance with those of several baseline approaches. We first built two simple baseline approaches based on sequence labelling models using the conditional random fields (CRFs). Such approaches can not handle overlapping mentions. To train such models, whenever two mentions overlap with one another in the training set, we remove the mention that is shorter in length. Following (Ratinov and Roth, 2009), we considered the BIO (Begin, Inside, Outside) approach and the BILOU (Begin, Inside, Last, Outside, Unit) approach for designing the output labels. Results show the BILOU approach yields better results. Similar observations were reported in Ratinov and Roth (2009).

In the work of (Alex et al., 2007), the authors proposed several approaches for building models to handle nested named entities in biomedical texts. Their best results were obtained from a *cascaded* approach where they built one model for each named entity class. Outputs from one model can then served as the inputs to the next model for predicting the named entity class of a different type. One fundamental limitation of such an approach is that it being unable to handle overlapping mentions of the same type. Nevertheless, this approach worked very well on both datasets. The results are shown in the row of "CRF (CC)".[6]

Another class of models that is often used in information extraction are the semi-Markov conditional random fields (semi-CRFs) (Sarawagi and Cohen, 2004). Semi-CRF models are able to capture the non-Markovian properties of mentions. However, they are unable to handle nested or overlapping mentions. We thus used the same method as discussed above to exclude certain mentions for training. Such semi-CRF models typically assume there is a length restriction for the mentions – each mention can consist of up to c words – in order to scale linearly. When such a restriction is lifted, the time complexity of such models becomes quadratic in the number of words in the input sentence. We train two models: one with a length restriction, where $c = 6$, and the other without a length restriction ($c = \infty$). For features defined over the inputs, besides the Markovian features described in Sec 3.5, we also used the surface forms of complete mention spans as features. The results of these two models are reported in the fourth and fifth row of Table 2, respectively. Interestingly, imposing the length restriction appears to be helpful for precision, and as a result it makes a positive contribution towards the final F measure.

[6]For all such linear chain CRF-related experiments, we used the CRF++ toolkit (https://code.google.com/p/crfpp/) with L-BFGS, which gives us the most competitive results over several different CRF implementations (see: http://www.chokkan.org/software/crfsuite/benchmark.html).

	7 TYPES		14 TYPES		28 TYPES	
	#f	w/s	#f	w/s	#f	w/s
CRF	3.6M	1219	13.6M	305	51.9M	76
MH	4.2M	**1532**	8.4M	**733**	16.9M	**430**

Table 3: The decoding time and the number of features change as we increase the number of possible types. (#f: number of features created (in millions). w/s: number of words processed per second.) Experiments are conducted on the ACE 2004 dataset.

Our basic model (**MH**: *mention hypergraph*) that optimizes the negative joint log likelihood is able to obtain the best precision across these two datasets. When the model is further augmented with the F measure optimization step described in Sec 3.7 (**MH** (F)) it consistently yields the best results in terms of both recall score and F measure across these two datasets.

4.1.1 Running Time

We also conducted controlled experiments to report the actual execution time of our model and make a comparison with the linear-chain CRF model (BILOU approach). The experiments are all conducted on the ACE2004 dataset on the same machine. To make a proper comparison here, we implemented the linear-chain CRF model using Java (the same language is used when implementing our model), and employed the same data structures for creating features as well as the same learning and inference routines used by our mention hypergraph model.

To understand how the features and speed change as we increase the number of mention types (*i.e.,* semantic types), we also conducted experiments where we increase the number of possible mention types. Specifically, we created subtypes from each original type annotated in the dataset. For example, we randomly replaced the type "GPE" by sub-types "GPE1" or "GPE2" in the dataset. This gave us 14 different mention types. Similarly, we could randomly replace the type "GPE" by sub-types "GPE1" – "GPE4", re-

	ACE2004						ACE2005					
	DEV			TEST			DEV			TEST		
	P	R	F	P	R	F	P	R	F	P	R	F
CRF (CC-S)	57.0	35.9	44.1	52.7	31.2	39.2	56.5	38.3	45.6	54.2	35.5	42.9
CRF (CC-F)	51.5	32.5	39.9	47.4	28.0	35.2	53.3	36.1	43.1	51.3	33.6	40.6
CRF (CC-L)	64.4	40.6	49.9	61.6	36.4	45.8	66.6	45.1	53.8	65.3	42.8	51.7
CRF (CC-CC)	63.6	40.5	49.5	60.8	36.1	45.3	65.9	44.8	53.3	64.2	42.0	50.7
MH (L)	66.7	41.9	51.5	64.7	38.2	48.1	70.5	45.0	55.0	69.2	43.0	53.0
MH (Joint)	**78.6**	47.4	59.1	**79.0**	44.1	56.6	**79.2**	48.7	60.3	**70.1**	45.1	54.8
MH (Joint F)	73.9	**52.4**	**61.3**	74.4	**50.0**	**59.8**	70.2	**57.4**	**63.2**	63.4	**53.8**	**58.3**

Table 4: Results on joint mention boundary, type, and head prediction on ACE2004 and ACE2005.

sulting in 28 different mention types in total. Our purpose of doing so is to understand how the models behave when the number of possible mention types becomes large. We found that training on the entire training set of ACE2004 using the linear-chain CRF model with a large number of mention types was very expensive due to the extremely large number of features involved. We instead trained the models on the development set and presented decoding time on the test set.

Table 3 shows the results. We empirically captured the relationship between the speed of each system (average number of words processed per second) and the number of mention types. Specifically, we found that as we linearly increased the number of mention types, for the linear-chain CRF model, the number of features grew quadratically and the speed dropped quadratically, whereas for our model, the number of features grew linearly and the speed dropped linearly. This indicates that our model is more scalable to large, practical datasets with a large number of fine-grained mention types.

4.1.2 Joint Modeling of Heads

We also conducted experiments on these two datasets for the task of joint modeling of mention boundaries, types and heads. We used the same training and tuning methodology for optimizing the F measure. In such experiments, we adopted a very strict evaluation criterion: a predicted mention is regarded as correct iff and only if its boundaries, type and head all exactly match those of the gold standard.

We compared our system's results with those of several baseline approaches based on CRF where the cascaded BILOU approach described above was always used. Specifically, we considered approaches that always regarded the complete span (CC-S), the first word (CC-F), and the last word (CC-L) as the predicted mention's head, respectively. We also considered a cascaded approach (CC-CC) where we first predicted mentions, and then predicted their heads by following a similar approach used for predicting overlapping mentions discussed above. The first four rows of Table 4 give the results of these baseline approaches. We can observe that always predicting the last word as the head gives the best performance. Inspired by this, we performed a simple approach by training a model presented in the previous section without considering head information. When making predictions, we always regarded the last word of each predicted mention as its head. The results for such an approach are given in the fifth row of Table 4. The sixth row shows the results obtained by optimizing our model's objective function. The last row gives the results obtained by tuning the mention penalty based on the development set. As seen, our joint models significantly outperformed all those baseline approaches. We are not aware of any prior work in the literature that performs joint modeling of mention boundaries, types, and heads.

4.2 Additional Experiments

We also additionally evaluated on the GENIA dataset (v3.02) whose focus was on biomedical related named entity recognition and classification, where the entities may overlap with one another. Furthermore, to see how our model works on datasets where mentions do not overlap with one another, we also conducted evaluations on the standard CONLL2003 NER dataset.

4.2.1 Results on GENIA

We followed the description of Finkel and Manning (2009) to set up our experiments on the GENIA dataset. Specifically, we used the first 90% of the sentences as the training data and the remaining 10% as the evaluation data. We also adhered to the paper's prescription of collapsing all *DNA* subtypes into *DNA*; *RNA* subtypes into *RNA*; and all *protein* subtypes into *protein*. We kept *cell line* and *cell type*, and removed all other entities.

To optimize the F measure, we further split the

	P	R	F
Semi-CRF	76.2	61.7	68.2
F & M (2009)	75.4	65.9	70.3
MH (F)	72.5	65.2	68.7

Table 5: Results on the GENIA dataset

training set into two portions. We trained a model using the first 90% of the training data, and used the remaining 10% for development. For features, no POS and no bag-of-words features are used.

We compared our model's performance with that of a model based on a constituency parser proposed by (Finkel and Manning, 2009), as well as the semi-CRF model reported there. The results are shown in Table 5. Our model yields a better F measure than the semi-CRF model, but gives a lower performance than the model of (Finkel and Manning, 2009). We note that, however, these results are not directly comparable. Specifically, both of these two previous models relied on an additional 200 million words from PubMed abstracts to learn word clusters as additional features, which we do not have access to.

One distinctive advantage of our model is the efficiency and scalability. The model of (Finkel and Manning, 2009) had a time complexity that is cubic in the number of words in the input sentence. In contrast, our model scales linearly as the length of the input sentence increases. [7]

4.2.2 Results on CONLL2003

To understand how well our model works on datasets where mentions or entities do not overlap with one another, we conducted additional experiments on the standard dataset used in the CONLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003), where the named entities strictly do not overlap with one another. We compared our system's performance against that of a baseline version of the state-of-the-art Illinois NER system (Ratinov and Roth, 2009). Their system performed sequential prediction over the input words and adopted the BILOU approach. Their full model also incorporates external knowledge resources (*e.g.*, gazetteers and word class).

In order to make a proper comparison with the baseline version of their model, besides the general features we mentioned earlier, we also followed (Ratinov and Roth, 2009) in incorporating word's prefixes and suffixes (of length up to 5) as features, and normalized words referring to months, dates and numbers. Table 6 shows that our system gives an F measure that is comparable to that of the baseline version of their system, where no external resources are used.

	DEV			TEST		
	P	R	F	P	R	F
Illinois (b)	-	-	89.3	-	-	83.7
MH	94.7	83.0	88.5	91.4	76.5	83.3
MH (F)	91.4	86.7	89.2	87.3	80.7	83.8

Table 6: Results on the CONLL2003. Illinois (b): baseline version of (Ratinov and Roth, 2009).

This additional experiment showed that while our model is designed for handling more realistic scenarios where mentions can overlap, it yields a performance competitive to a state-of-the-art system which only handles datasets with non-overlapping mentions.

5 Conclusions

In this work, we have introduced a novel model for the task of joint modeling of mention boundaries, types, as well as their heads. Unlike many previous research efforts for mention extraction and classification, our novel *mention hypergraph* representations for compactly representing exponentially many possible mentions enables a mention's boundaries, type and head information to be jointly learned in a single framework. The model scales linearly with respect to the number of words in the input sentence, and performs exact learning where a unique global optimum can be found. Empirically, we have demonstrated the effectiveness of such a model across several standard datasets.

Future work include explorations of efficient algorithms for other information extraction tasks, such as joint mention and relation extraction (Li and Ji, 2014) and event extraction (Li et al., 2013). Our system and code can be downloaded from `http://statnlp.org/research/ie/`.

Acknowledgements

We would like to thank Kian Ming A. Chai, Hai Leong Chieu and the three anonymous reviewers for their comments on this work. This work is supported by Temasek Lab of Singapore University of Technology and Design project IGDSS1403011 and IGDST1403013, and is partly supported by DARPA (under agreement number FA8750-13-2-0008).

[7] In our experiments, for this dataset our model tagged over 5,000 words/second. In (Finkel and Manning, 2009), the authors mentioned that their model tagged about 38 words/second, and the semi-CRF model tagged about 45 words/second. However, we note these numbers are not directly comparable due to the advancement of CPU speed.

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *BioNLP*, pages 65–72. Association for Computational Linguistics.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A Discriminative Latent Variable Model for Statistical Machine Translation. In *ACL*, pages 200–208.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Xavier Carreras, Lluis Marquez, and Lluís Padró. 2002. Named entity extraction using adaboost. *CONLL*, pages 167–170.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A Constrained Latent Variable Model for Coreference Resolution. In *EMNLP*, pages 601–612.

Wanxiang Che, Mengqiu Wang, Christopher D Manning, and Ting Liu. 2013. Named Entity Recognition with Bilingual Constraints. In *NAACL-HLT*, pages 52–62.

Martin Cmejrek, Haitao Mi, and Bowen Zhou. 2013. Flexible and efficient hypergraph interactions for joint hierarchical and forest-to-string decoding. In *EMNLP*, pages 545–555.

Alessandro Cucchiarelli and Paola Velardi. 2001. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131.

Aron Culotta and Andrew McCallum. 2004. Confidence estimation for information extraction. In *HLT-NAACL*, pages 109–112.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *EMNLP*, pages 141–150. Association for Computational Linguistics.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *CONLL*, pages 168–171. Association for Computational Linguistics.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, H Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *HLT-NAACL*, pages 1–8. DTIC Document.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *NAACL-HLT*, pages 1020–1030.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2001. Parsing and hypergraphs. In *IWPT*, pages 123–134.

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

Qi Li and Heng Ji. 2014. Incremental Joint Extraction of Entity Mentions and Relations. In *ACL*, pages 402–412.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *ACL*, pages 73–82.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Wei Lu. 2015. Constrained semantic forests for improved discriminative semantic parsing. In *ACL*.

Ryan McDonald and Fernando Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *BMC bioinformatics*, 6(S6).

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP*, pages 1003–1011. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Arvind Neelakantan and Michael Collins. 2014. Learning Dictionaries for Named Entity Recognition using Minimal Supervision. In *EACL*, pages 452–461.

Panupong Pasupat and Percy Liang. 2014. Zero-shot Entity Extraction from Web Pages. In *ACL*, pages 391–401.

Slav Petrov and Dan Klein. 2007. Discriminative log-linear grammars with latent variables. In *NIPS*, pages 1153–1160.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CONLL*, pages 147–155. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP*, pages 1524–1534. Association for Computational Linguistics.

Sunita Sarawagi and William W Cohen. 2004. Semi-markov conditional random fields for information extraction. In *NIPS*, pages 1185–1192.

Jun Suzuki, Erik McDermott, and Hideki Isozaki. 2006. Training conditional random fields with multivariate evaluation measures. In *COLING/AC*, pages 217–224.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CONLL*, pages 142–147. Association for Computational Linguistics.

Mengqiu Wang, Wanxiang Che, and Christopher D Manning. 2013. Joint Word Alignment and Bilingual Named Entity Recognition Using Dual Decomposition. In *ACL*, pages 1073–1082.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *ACL*, pages 473–480.