Incremental Joint Extraction of Entity Mentions and Relations

Qi Li **Heng Ji**
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
{liq7,jih}@rpi.edu

Abstract

We present an incremental joint framework to simultaneously extract entity mentions and relations using structured perceptron with efficient beam-search. A segment-based decoder based on the idea of semi-Markov chain is adopted to the new framework as opposed to traditional token-based tagging. In addition, by virtue of the inexact search, we developed a number of new and effective global features as soft constraints to capture the interdependency among entity mentions and relations. Experiments on Automatic Content Extraction (ACE)[1] corpora demonstrate that our joint model significantly outperforms a strong pipelined baseline, which attains better performance than the best-reported end-to-end system.

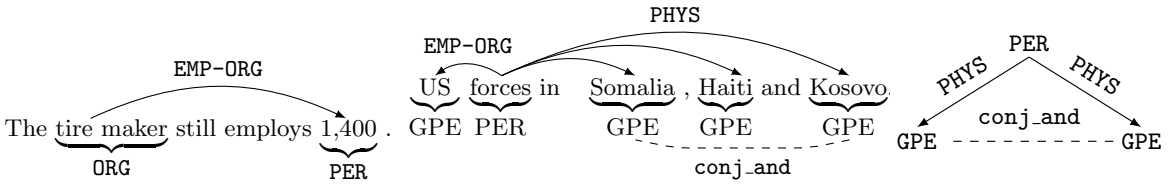
1 Introduction

The goal of end-to-end entity mention and relation extraction is to discover relational structures of entity mentions from unstructured texts. This problem has been artificially broken down into several components such as entity mention boundary identification, entity type classification and relation extraction. Although adopting such a pipelined approach would make a system comparatively easy to assemble, it has some limitations: First, it prohibits the interactions between components. Errors in the upstream components are propagated to the downstream components without any feedback. Second, it over-simplifies the problem as multiple local classification steps without modeling long-distance and cross-task dependencies. By contrast, we re-formulate this task as a structured prediction problem to reveal the linguistic and logical properties of the hidden structures. For example, in Figure 1, the output structure of each sentence can be interpreted as a graph in which entity mentions are nodes and relations are directed arcs with relation types. By jointly predicting the structures, we aim to address the aforementioned limitations by capturing: (i) The interactions between two tasks. For example, in Figure 1a, although it may be difficult for a mention extractor to predict "1,400" as a Person (PER) mention, the context word "employs" between "tire maker" and "1,400" strongly indicates an Employment-Organization (EMP-ORG) relation which must involve a PER mention. (ii) The global features of the hidden structure. Various entity mentions and relations share linguistic and logical constraints. For example, we can use the triangle feature in Figure 1b to ensure that the relations between "forces", and each of the entity mentions "Somalia$_{\text{GPE}}$", "Haiti$_{\text{GPE}}$" and "Kosovo$_{\text{GPE}}$", are of the same type (Physical (PHYS), in this case).

Following the above intuitions, we introduce a joint framework based on structured perceptron (Collins, 2002; Collins and Roark, 2004) with beam-search to extract entity mentions and relations simultaneously. With the benefit of inexact search, we are also able to use arbitrary global features with low cost. The underlying learning algorithm has been successfully applied to some other Natural Language Processing (NLP) tasks. Our task differs from dependency parsing (such as (Huang and Sagae, 2010)) in that relation structures are more flexible, where each node can have arbitrary relation arcs. Our previous work (Li et al., 2013) used perceptron model with token-based tagging to jointly extract event triggers and arguments. By contrast, we aim to address a more challenging task: identifying mention boundaries and types together with relations, which raises the issue that assignments for the same sentence with different mention boundaries are difficult to syn-

[1] http://www.itl.nist.gov/iad/mig//tests/ace

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 402–412,
Baltimore, Maryland, USA, June 23-25 2014.



(a) Interactions between Two Tasks (b) Example of Global Feature

Figure 1: End-to-End Entity Mention and Relation Extraction.

chronize during search. To tackle this problem, we adopt a segment-based decoding algorithm derived from (Sarawagi and Cohen, 2004; Zhang and Clark, 2008) based on the idea of semi-Markov chain (a.k.a, multiple-beam search algorithm).

Most previous attempts on joint inference of entity mentions and relations (such as (Roth and Yih, 2004; Roth and Yih, 2007)) assumed that entity mention boundaries were given, and the classifiers of mentions and relations are separately learned. As a key difference, we incrementally extract entity mentions together with relations using a single model. The main contributions of this paper are as follows:

1. This is the first work to incrementally predict entity mentions and relations using a single joint model (Section 3).
2. Predicting mention boundaries in the joint framework raises the challenge of synchronizing different assignments in the same beam. We solve this problem by detecting entity mentions on segment-level instead of traditional token-based approaches (Section 3.1.1).
3. We design a set of novel global features based on soft constraints over the entire output graph structure with low cost (Section 4).

Experimental results show that the proposed framework achieves better performance than pipelined approaches, and global features provide further significant gains.

2 Background

2.1 Task Definition

The entity mention extraction and relation extraction tasks we are addressing are those of the Automatic Content Extraction (ACE) program[2]. ACE defined 7 main entity types including Person (PER), Organization (ORG), Geographical Entities (GPE), Location (LOC), Facility (FAC), Weapon (WEA) and Vehicle (VEH). The goal of relation extraction[3] is to extract semantic relations of the targeted types between a pair of entity mentions which appear in the same sentence. ACE'04 defined 7 main relation types: Physical (PHYS), Person-Social (PER-SOC), Employment-Organization (EMP-ORG), Agent-Artifact (ART), PER/ORG Affiliation (Other-AFF), GPE-Affiliation (GPE-AFF) and Discourse (DISC). ACE'05 kept PER-SOC, ART and GPE-AFF, split PHYS into PHYS and a new relation type Part-Whole, removed DISC, and merged EMP-ORG and Other-AFF into EMP-ORG.

Throughout this paper, we use $\perp$ to denote non-entity or non-relation classes. We consider relation asymmetric. The same relation type with opposite directions is considered to be two classes, which we refer to as *directed relation types*.

Most previous research on relation extraction assumed that entity mentions were given In this work we aim to address the problem of end-to-end entity mention and relation extraction from raw texts.

2.2 Baseline System

In order to develop a baseline system representing state-of-the-art pipelined approaches, we trained a linear-chain Conditional Random Fields model (Lafferty et al., 2001) for entity mention extraction and a Maximum Entropy model for relation extraction.

Entity Mention Extraction Model We re-cast the problem of entity mention extraction as a sequential token tagging task as in the state-of-the-art system (Florian et al., 2006). We applied the BILOU scheme, where each tag means a token is the **B**eginning, **I**nside, **L**ast, **O**utside, and **U**nit of an entity mention, respectively. Most of our features are similar to the work of (Florian et al.,

[2]http://www.nist.gov/speech/tests/ace

[3]Throughout this paper we refer to relation mention as relation since we do not consider relation mention coreference.

2004; Florian et al., 2006) except that we do not have their gazetteers and outputs from other mention detection systems as features. Our additional features are as follows:

- Governor word of the current token based on dependency parsing (Marneffe et al., 2006).
- Prefix of each word in Brown clusters learned from TDT5 corpus (Sun et al., 2011).

Relation Extraction Model Given a sentence with entity mention annotations, the goal of baseline relation extraction is to classify each mention pair into one of the pre-defined relation types with direction or $\perp$ (non-relation). Most of our relation extraction features are based on the previous work of (Zhou et al., 2005) and (Kambhatla, 2004). We designed the following additional features:

- The label sequence of phrases covering the two mentions. For example, for the sentence in Figure 1a, the sequence is "NP VP NP". We also augment it by head words of each phrase.
- Four syntactico - semantic patterns described in (Chan and Roth, 2010).
- We replicated each lexical feature by replacing each word with its Brown cluster.

3 Algorithm

3.1 The Model

Our goal is to predict the hidden structure of each sentence based on arbitrary features and constraints. Let $x \in \mathcal{X}$ be an input sentence, $y' \in \mathcal{Y}$ be a candidate structure, and $\mathbf{f}(x, y')$ be the feature vector that characterizes the entire structure. We use the following linear model to predict the most probable structure $\hat{y}$ for x:

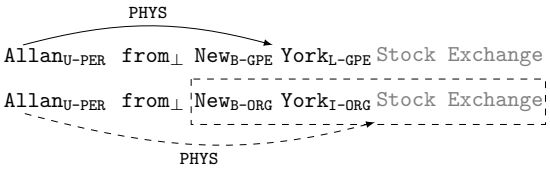
$$\hat{y} = \underset{y' \in \mathcal{Y}(x)}{\operatorname{argmax}} \mathbf{f}(x, y') \cdot \mathbf{w} \quad (1)$$

where the score of each candidate assignment is defined as the inner product of the feature vector $\mathbf{f}(x, y')$ and feature weights $\mathbf{w}$.

Since the structures contain both entity mentions relations, and we also aim to exploit global features. There does not exist a polynomial-time algorithm to find the best structure. In practice we apply beam-search to expand partial configurations for the input sentence incrementally to find the structure with the highest score.

3.1.1 Joint Decoding Algorithm

One main challenge to search for entity mentions and relations incrementally is the alignment of different assignments. Assignments for the same sentence can have different numbers of entity mentions and relation arcs. The entity mention extraction task is often re-cast as a token-level sequential labeling problem with BIO or BILOU scheme (Ratinov and Roth, 2009; Florian et al., 2006). A naive solution to our task is to adopt this strategy by treating each token as a state. However, different assignments for the same sentence can have various mention boundaries. It is unfair to compare the model scores of a partial mention and a complete mention. It is also difficult to synchronize the search process of relations. For example, consider the two hypotheses ending at "York" for the same sentence:



The model would bias towards the incorrect assignment "New/B-GPE York/L-GPE" since it can have more informative features as a complete mention (e.g., a binary feature indicating if the entire mention appears in a GPE gazetter). Furthermore, the predictions of the two PHYS relations cannot be synchronized since "New/B-FAC York/I-FAC" is not yet a complete mention.

To tackle these problems, we employ the idea of semi-Markov chain (Sarawagi and Cohen, 2004), in which each state corresponds to a segment of the input sequence. They presented a variant of Viterbi algorithm for exact inference in semi-Markov chain. We relax the max operation by beam-search, resulting in a segment-based decoder similar to the multiple-beam algorithm in (Zhang and Clark, 2008). Let $\hat{d}$ be the upper bound of entity mention length. The k-*best* partial assignments ending at the i-th token can be calculated as:

$$B[i] = \underset{y' \in \{y_{[1..i]} | y_{[1:i-d]} \in B[i-d],\, d=1...\hat{d}\}}{\text{k-}BEST} \mathbf{f}(x, y') \cdot \mathbf{w}$$

where $y_{[1:i-d]}$ stands for a partial configuration ending at the $(i\text{-}d)$-th token, and $y_{[i-d+1,i]}$ corresponds to the structure of a new segment (i.e., subsequence of x) $x_{[i-d+1,i]}$. Our joint decoding algorithm is shown in Figure 2. For each token index i, it maintains a beam for the partial assignments whose last segments end at the i-th token. There are two types of actions during the search:

Input: input sentence $x = (x_1, x_2, ..., x_m)$.
k: beam size.
$\mathcal{T} \cup \{\perp\}$: entity mention type alphabet.
$\mathcal{R} \cup \{\perp\}$: directed relation type alphabet.[4]
d_t: max length of type-t segment, $t \in \mathcal{T} \cup \{\perp\}$.
Output: best configuration $\hat{y}$ for x

```
1  initialize m empty beams B[1..m]
2  for i ← 1...m do
3    for t ∈ T ∪ {⊥} do
4      for d ← 1...d_t, y' ∈ B[i − d] do
5        k ← i − d + 1
6        B[i] ← B[i] ∪ APPEND(y', t, k, i)
7    B[i] ← k-BEST(B[i])
8    for j ← (i − 1)...1 do
9      buf ← ∅
10     for y' ∈ B[i] do
11       if HASPAIR(y', i, j) then
12         for r ∈ R ∪ {⊥} do
13           buf ← buf ∪ LINK(y', r, i, j)
14       else
15         buf ← buf ∪ {y'}
16     B[i] ← k-BEST(buf)
17 return B[m][0]
```

Figure 2: Joint Decoding for Entity Mentions and Relations. *HASPAIR*(y', i, j) checks if there are two entity mentions in y' that end at token x_i and token x_j, respectively. *APPEND*(y', t, k, i) appends y' with a type-t segment spanning from x_k to x_i. Similarly *LINK*(y', r, i, j) augments y' by assigning a directed relation r to the pair of entity mentions ending at x_i and x_j respectively.

1. *APPEND* (Lines 3-7). First, the algorithm enumerates all possible segments (i.e., subsequences) of x ending at the current token with various entity types. A special type of segment is a single token with non-entity label ($\perp$). Each segment is then appended to existing partial assignments in one of the previous beams to form new assignments. Finally the top k results are recorded in the current beam.
2. *LINK* (Lines 8-16). After each step of *APPEND*, the algorithm looks backward to link the newly identified entity mentions and previous ones (if any) with relation arcs. At the j-th sub-step, it only considers the previous mention ending at the j-th previous token. Therefore different configurations are guaranteed to have the same number of sub-steps. Finally, all assignments are re-ranked with new relation information.

[4]The same relation type with opposite directions is considered to be two classes in $\mathcal{R}$.

There are m *APPEND* actions, each is followed by at most $(i-1)$ *LINK* actions (line 8). Therefore the worst-case time complexity is $O(\hat{d} \cdot k \cdot m^2)$, where $\hat{d}$ is the upper bound of segment length.

3.1.2 Example Demonstration

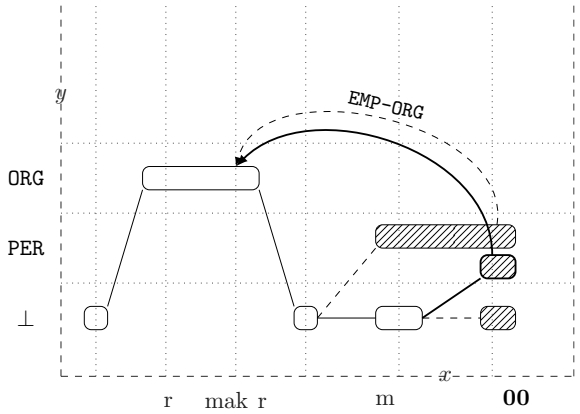


Figure 3: Example of decoding steps. x-axis and y-axis represent the input sentence and entity types, respectively. The rectangles denote segments with entity types, among which the shaded ones are three competing hypotheses ending at "1,400". The solid lines and arrows indicate correct *APPEND* and *LINK* actions respectively, while the dashed indicate incorrect actions.

Here we demonstrate a simple but concrete example by considering again the sentence described in Figure 1a. Suppose we are at the token "1,400". At this point we can propose multiple entity mentions with various lengths. Assuming "1,400$_{/\texttt{PER}}$", "1,400$_{/\perp}$" and "(employs 1,400)$_{/\texttt{PER}}$" are possible assignments, the algorithm appends these new segments to the partial assignments in the beams of the tokens "employs" and "still", respectively. Figure 3 illustrates this process. For simplicity, only a small part of the search space is presented. The algorithm then links the newly identified mentions to the previous ones in the same configuration. In this example, the only previous mention is "(tire maker)$_{/\texttt{ORG}}$". Finally, "1,400$_{/\texttt{PER}}$" will be preferred by the model since there are more indicative context features for EMP−ORG relation between "(tire maker)$_{/\texttt{PER}}$" and "1,400$_{/\texttt{PER}}$".

3.2 Structured-Perceptron Learning

To estimate the feature weights, we use structured perceptron (Collins, 2002), an extension of the standard perceptron for structured prediction, as the learning framework. Huang et al. (2012) proved the convergency of structured perceptron when inexact search is applied with violation-fixing update methods such as early-update (Collins and Roark, 2004). Since we use beam-search in this work, we apply early-update. In addition, we use averaged parameters to reduce overfitting as in (Collins, 2002).

Figure 4 shows the pseudocode for structured perceptron training with early-update. Here *BEAMSEARCH* is identical to the decoding algorithm described in Figure 2 except that if y', the prefix of the gold standard y, falls out of the beam after each execution of the k-*BEST* function (line 7 and 16), then the top assignment z and y' are returned for parameter update. It is worth noting that this can only happen if the gold-standard has a segment ending at the current token. For instance, in the example of Figure 1a, $B[2]$ cannot trigger any early-update since the gold standard does not contain any segment ending at the second token.

Input: training set $\mathcal{D} = \{(x^{(j)}, y^{(j)})\}_{i=1}^{N}$, maximum iteration number T
Output: model parameters $\mathbf{w}$
1 initialize $\mathbf{w} \leftarrow \mathbf{0}$
2 **for** $t \leftarrow 1...T$ **do**
3 **foreach** $(x, y) \in \mathcal{D}$ **do**
4 $(x, y', z) \leftarrow$ *BEAMSEARCH* $(x, y, \mathbf{w})$
5 **if** $z \neq y$ **then**
6 $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{f}(x, y') - \mathbf{f}(x, z)$
7 return $\mathbf{w}$

Figure 4: Perceptron algorithm with beam-search and early-update. y' is the prefix of the gold-standard and z is the top assignment.

3.3 Entity Type Constraints

Entity type constraints have been shown effective in predicting relations (Roth and Yih, 2007; Chan and Roth, 2010). We automatically collect a mapping table of permissible entity types for each relation type from our training data. Instead of applying the constraints in post-processing inference, we prune the branches that violate the type constraints during search. This type of pruning can reduce search space as well as make the input for parameter update less noisy. In our experiments, only 7 relation mentions (0.5%) in the dev set and 5 relation mentions (0.3%) in the test set violate the constraints collected from the training data.

4 Features

An advantage of our framework is that we can easily exploit arbitrary features across the two tasks. This section describes the local features (Section 4.1) and global features (Section 4.2) we developed in this work.

4.1 Local Features

We design segment-based features to directly evaluate the properties of an entity mention instead of the individual tokens it contains. Let $\hat{y}$ be a predicted structure of a sentence x. The entity segments of $\hat{y}$ can be expressed as a list of triples $(e_1, ..., e_m)$, where each segment $e_i = \langle u_i, v_i, t_i \rangle$ is a triple of start index u_i, end index v_i, and entity type t_i. The following is an example of segment-based feature:

$$f_{001}(x, \hat{y}, i) = \begin{cases} 1 & \text{if } x_{[\hat{y}.u_i, \hat{y}.v_i]} = \text{ tire maker} \\ & \hat{y}.t_{(i-1)}, \hat{y}.t_i = \perp, \texttt{ORG} \\ 0 & \text{otherwise} \end{cases}$$

This feature is triggered if the labels of the $(i-1)$-th and the i-th segments are "$\perp$, ORG", and the text of the i-th segment is "tire maker". Our segment-based features are described as follows:

Gazetteer features Entity type of each segment based on matching a number of gazetteers including persons, countries, cities and organizations.

Case features Whether a segment's words are initial-capitalized, all lower cased, or mixture.

Contextual features Unigrams and bigrams of the text and part-of-speech tags in a segment's contextual window of size 2.

Parsing-based features Features derived from constituent parsing trees, including (a) the phrase type of the lowest common ancestor of the tokens contained in the segment, (b) the depth of the lowest common ancestor, (c) a binary feature indicating if the segment is a base phrase or a suffix of a base phrase, and (d) the head words of the segment and its neighbor phrases.

In addition, we convert each triple $\langle u_i, v_i, t_i \rangle$ to BILOU tags for the tokens it contains to implement token-based features. The token-based men-

tion features and local relation features are identical to those of our pipelined system (Section 2.2).

4.2 Global Entity Mention Features

By virtue of the efficient inexact search, we are able to use arbitrary features from the entire structure of $\hat{y}$ to capture long-distance dependencies. The following features between related entity mentions are extracted once a new segment is appended during decoding.

Coreference consistency Coreferential entity mentions should be assigned the same entity type. We determine high-recall coreference links between two segments in the same sentence using some simple heuristic rules:

- Two segments exactly or partially string match.
- A pronoun (e.g., "*their*","*it*") refers to previous entity mentions. For example, in "***they** have no insurance on **their** cars*", "*they*" and "*their*" should have the same entity type.
- A relative pronoun (e.g., "*which*","*that*", and "*who*") refers to the noun phrase it modifies in the parsing tree. For example, in "*the starting kicker is **nikita kargalskiy**, **who** may be 5,000 miles from his hometown*", "*nikita kargalskiy*" and "*who*" should both be labeled as persons.

Then we encode a global feature to check whether two coreferential segments share the same entity type. This feature is particularly effective for pronouns because their contexts alone are often not informative.

Neighbor coherence Neighboring entity mentions tend to have coherent entity types. For example, in "***Barbara Starr** was reporting from the **Pentagon***", "*Barbara Starr*" and "*Pentagon*" are connected by a dependency link *prep_from* and thus they are unlikely to be a pair of PER mentions. Two types of neighbor are considered: (i) the first entity mention before the current segment, and (ii) the segment which is connected by a single word or a dependency link with the current segment. We take the entity types of the two segments and the linkage together as a global feature. For instance, "PER *prep_from* PER" is a feature for the above example when "*Barbara Starr*" and "*Pentagon*" are both labeled as PER mentions.

Part-of-whole consistency If an entity mention is semantically part of another mention (connected by a *prep_of* dependency link), they should be assigned the same entity type. For example, in "***some** of Iraq's **exiles***", "*some*" and "*exiles*" are both PER mentions; in "***one** of the town's two meat-packing **plants***", "*one*" and "*plants*" are both FAC mentions; in "*the **rest** of **America***", "*rest*" and "*America*" are both GPE mentions.

4.3 Global Relation Features

Relation arcs can also share inter-dependencies or obey soft constraints. We extract the following relation-centric global features when a new relation hypothesis is made during decoding.

Role coherence If an entity mention is involved in multiple relations with the same type, then its roles should be coherent. For example, a PER mention is unlikely to have more than one employer. However, a GPE mention can be a physical location for multiple entity mentions. We combine the relation type and the entity mention's argument roles as a global feature, as shown in Figure 5a.

Triangle constraint Multiple entity mentions are unlikely to be fully connected with the same relation type. We use a negative feature to penalize any configuration that contains this type of structure. An example is shown in Figure 5b.

Inter-dependent compatibility If two entity mentions are connected by a dependency link, they tend to have compatible relations with other entities. For example, in Figure 5c, the *conj_and* dependency link between "*Somalia*" and "*Kosovo*" indicates they may share the same relation type with the third entity mention "*forces*".

Neighbor coherence Similar to the entity mention neighbor coherence feature, we also combine the types of two neighbor relations in the same sentence as a bigram feature.

5 Experiments

5.1 Data and Scoring Metric

Most previous work on ACE relation extraction has reported results on ACE'04 data set. As we will show later in our experiments, ACE'05 made significant improvement on both relation type definition and annotation quality. Therefore we present the overall performance on ACE'05 data. We removed two small subsets in informal genres - *cts* and *un*, and then randomly split the remaining 511 documents into 3 parts: 351 for training, 80 for development, and the rest 80 for blind test. In order to compare with state-of-the-art we also performed the same 5-fold cross-validation on *bnews* and *nwire* subsets of ACE'04 corpus as in previous work. The statistics of these data sets

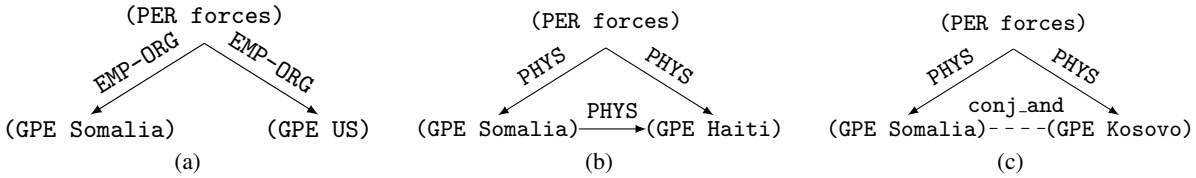
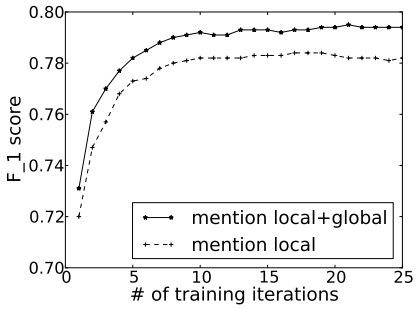
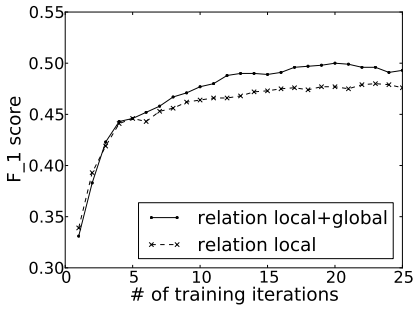


Figure 5: Examples of Global Relation Features.



(a) Entity Mention Performance



(b) Relation Performance

Figure 6: Learning Curves on Development Set.

are summarized in Table 1. We ran the Stanford CoreNLP toolkit[5] to automatically recover the true cases for lowercased documents.

Data Set		# sentences	# mentions	# relations
ACE'05	Train	7,273	26,470	4,779
	Dev	1,765	6,421	1,179
	Test	1,535	5,476	1,147
ACE'04		6,789	22,740	4,368

Table 1: Data Sets.

We use the standard F_1 measure to evaluate the performance of entity mention extraction and relation extraction. An entity mention is considered correct if its entity type is correct and the offsets of its mention head are correct. A relation mention is considered correct if its relation type is correct, and the head offsets of two entity mention arguments are both correct. As in Chan and Roth (2011), we excluded the DISC relation type, and removed relations in the system output which are implicitly correct via coreference links for fair comparison. Furthermore, we combine these two criteria to evaluate the performance of end-to-end entity mention and relation extraction.

[5]http://nlp.stanford.edu/software/corenlp.shtml

5.2 Development Results

In general a larger beam size can yield better performance but increase training and decoding time. As a tradeoff, we set the beam size as 8 throughout the experiments. Figure 6 shows the learning curves on the development set, and compares the performance with and without global features. From these figures we can clearly see that global features consistently improve the extraction performance of both tasks. We set the number of training iterations as 22 based on these curves.

5.3 Overall Performance

Table 2 shows the overall performance of various methods on the ACE'05 test data. We compare our proposed method (Joint w/ Global) with the pipelined system (Pipeline), the joint model with only local features (Joint w/ Local), and two human annotators who annotated 73 documents in ACE'05 corpus.

We can see that our approach significantly outperforms the pipelined approach for both tasks. As a real example, for the partial sentence "*a marcher from Florida*" from the test data, the pipelined approach failed to identify "*marcher*" as a PER mention, and thus missed the GEN-AFF relation between "*marcher*" and "*Florida*". Our joint model correctly identified the entity mentions and their relation. Figure 7 shows the details when the joint model is applied to this sentence. At the token "*marcher*", the top hypothesis in the beam is "$\langle \perp, \perp \rangle$", while the correct one is ranked second best. After the decoder processes the token "*Florida*", the correct hypothesis is promoted to the top in the beam by the *Neighbor Coherence* features for PER-GPE pair. Furthermore, after

Model	Entity Mention (%)			Relation (%)			Entity Mention + Relation (%)		
Score	P	R	F_1	P	R	F_1	P	R	F_1
Pipeline	83.2	73.6	78.1	67.5	39.4	49.8	65.1	38.1	48.0
Joint w/ Local	84.5	76.0	80.0	68.4	40.1	50.6	65.3	38.3	48.3
Joint w/ Global	85.2	76.9	**80.8**	68.9	41.9	**52.1**	65.4	39.8	**49.5**
Annotator 1	91.8	89.9	90.9	71.9	69.0	70.4	69.5	66.7	68.1
Annotator 2	88.7	88.3	88.5	65.2	63.6	64.4	61.8	60.2	61.0
Inter-Agreement	85.8	87.3	86.5	55.4	54.7	55.0	52.3	51.6	51.9

Table 2: Overall performance on ACE'05 corpus.

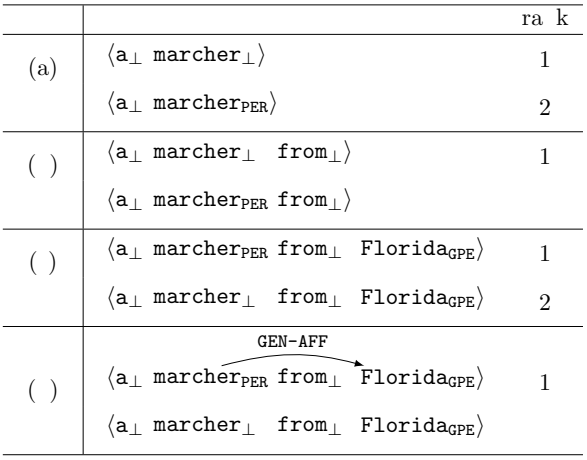


Figure 7: Two competing hypotheses for "*a marcher from Florida*" during decoding.

linking the two mentions by GEN-AFF relation, the ranking of the incorrect hypothesis "$\langle \perp, \perp \rangle$" is dropped to the 4-th place in the beam, resulting in a large margin from the correct hypothesis.

The human F_1 score on end-to-end relation extraction is only about 70%, which indicates it is a very challenging task. Furthermore, the F_1 score of the inter-annotator agreement is 51.9%, which is only 2.4% above that of our proposed method.

Compared to human annotators, the bottleneck of automatic approaches is the low recall of relation extraction. Among the 631 remaining missing relations, 318 (50.3%) of them were caused by missing entity mention arguments. A lot of nominal mention heads rarely appear in the training data, such as persons ("supremo", "shepherd", "oligarchs", "rich"), geo-political entity mentions ("stateside"), facilities ("roadblocks", "cells"), weapons ("sim lant", "nukes") and vehicles ("prams"). In addition, relations are often implicitly expressed in a variety of forms. Some examples are as follows:

- "*Rice has been chosen by President Bush to become the new Secretary of State*" indicates "Rice" has a PER-SOC relation with "Bush".
- "*U.S. troops are now knocking on the door of Baghdad*" indicates "troops" has a PHYS relation with "Baghdad".
- "*Russia and France sent planes to Baghdad*" indicates "Russia" and "France" are involved in an ART relation with "planes" as owners.

In addition to contextual features, deeper semantic knowledge is required to capture such implicit semantic relations.

5.4 Comparison with State-of-the-art

Table 3 compares the performance on ACE'04 corpus. For entity mention extraction, our joint model achieved 79.7% on 5-fold cross-validation, which is comparable with the best F_1 score 79.2% reported by (Florian et al., 2006) on single-fold. However, Florian et al. (2006) used some gazetteers and the output of other Information Extraction (IE) models as additional features, which provided significant gains ((Florian et al., 2004)). Since these gazetteers, additional data sets and external IE models are all not publicly available, it is not fair to directly compare our joint model with their results.

For end-to-end entity mention and relation extraction, both the joint approach and the pipelined baseline outperform the best results reported by (Chan and Roth, 2011) under the same setting.

6 Related Work

Entity mention extraction (e.g., (Florian et al., 2004; Florian et al., 2006; Florian et al., 2010; Zitouni and Florian, 2008; Ohta et al., 2012)) and relation extraction (e.g., (Reichartz et al., 2009; Sun et al., 2011; Jiang and Zhai, 2007; Bunescu and Mooney, 2005; Zhao and Grishman, 2005; Culotta and Sorensen, 2004; Zhou et al., 2007; Qian and Zhou, 2010; Qian et al., 2008; Chan and Roth, 2011; Plank and Moschitti, 2013)) have drawn much attention in recent years but were

Model	Entity Mention (%)			Relation (%)			Entity Mention + Relation (%)		
Score	P	R	F_1	P	R	F_1	P	R	F_1
Chan and Roth (2011)		-		42.9	38.9	40.8		-	
Pipeline	81.5	74.1	77.6	62.5	36.4	46.0	58.4	33.9	42.9
Joint w/ Local	82.7	75.2	78.8	64.2	37.0	46.9	60.3	34.8	44.1
Joint w/ Global	83.5	76.2	**79.7**	64.7	38.5	**48.3**	60.8	36.1	**45.3**

Table 3: 5-fold cross-validation on ACE'04 corpus. Bolded scores indicate highly statistical significant improvement as measured by paired t-test ($p < 0.01$)

usually studied separately. Most relation extraction work assumed that entity mention boundaries and/or types were given. Chan and Roth (2011) reported the best results using predicted entity mentions.

Some previous work used relations and entity mentions to enhance each other in joint inference frameworks, including re-ranking (Ji and Grishman, 2005), Integer Linear Programming (ILP) (Roth and Yih, 2004; Roth and Yih, 2007; Yang and Cardie, 2013), and Card-pyramid Parsing (Kate and Mooney, 2010). All these work noted the advantage of exploiting cross-component interactions and richer knowledge. However, they relied on models separately learned for each subtask. As a key difference, our approach jointly extracts entity mentions and relations using a single model, in which arbitrary soft constraints can be easily incorporated. Some other work applied probabilistic graphical models for joint extraction (e.g., (Singh et al., 2013; Yu and Lam, 2010)). By contrast, our work employs an efficient joint search algorithm without modeling joint distribution over numerous variables, therefore it is more flexible and computationally simpler. In addition, (Singh et al., 2013) used gold-standard mention boundaries.

Our previous work (Li et al., 2013) used structured perceptron with token-based decoder to jointly predict event triggers and arguments based on the assumption that entity mentions and other argument candidates are given as part of the input. In this paper, we solve a more challenging problem: take raw texts as input and identify the boundaries, types of entity mentions and relations all together in a single model. Sarawagi and Cohen (2004) proposed a segment-based CRFs model for name tagging. Zhang and Clark (2008) used a segment-based decoder for word segmentation and pos tagging. We extended the similar idea to our end-to-end task by incrementally predicting relations along with entity mention segments.

7 Conclusions and Future Work

In this paper we introduced a new architecture for more powerful end-to-end entity mention and relation extraction. For the first time, we addressed this challenging task by an incremental beam-search algorithm in conjunction with structured perceptron. While detecting mention boundaries jointly with other components raises the challenge of synchronizing multiple assignments in the same beam, a simple yet effective segment-based decoder is adopted to solve this problem. More importantly, we exploited a set of global features based on linguistic and logical properties of the two tasks to predict more coherent structures. Experiments demonstrated our approach significantly outperformed pipelined approaches for both tasks and dramatically advanced state-of-the-art.

In future work, we plan to explore more soft and hard constraints to reduce search space as well as improve accuracy. In addition, we aim to incorporate other IE components such as event extraction into the joint model.

Acknowledgments

We thank the three anonymous reviewers for their insightful comments. This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), U.S. NSF CAREER Award under Grant IIS-0953149, U.S. DARPA Award No. FA8750-13-2-0041 in the Deep Exploration and Filtering of Text (DEFT) Program, IBM Faculty Award, Google Research Award and RPI faculty start-up grant. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proc. HLT/EMNLP*, pages 724–731.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proc. COLING*, pages 152–160.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proc. ACL*, pages 551–560.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proc. ACL*, pages 111–118.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*, pages 1–8.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. ACL*, pages 423–429.

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

Radu Florian, Hongyan Jing, Nanda Kambhatla, and Imed Zitouni. 2006. Factorizing complex models: A case study in mention detection. In *Proc. ACL*.

Radu Florian, John F. Pitrelli, Salim Roukos, and Imed Zitouni. 2010. Improving mention detection robustness to noisy input. In *Proc. EMNLP*, pages 335–345.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *ACL*, pages 1077–1086.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proc. HLT-NAACL*, pages 142–151.

Heng Ji and Ralph Grishman. 2005. Improving name tagging by reference resolution and relation detection. In *Proc. ACL*, pages 411–418.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proc. HLT-NAACL*.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proc. ACL*, pages 178–181.

Rohit J. Kate and Raymond Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proc. ACL*, pages 203–212.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proc. ACL*, pages 73–82.

Marie-Catherine De Marneffe, Bill Maccartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. LREC*, pages 449,454.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proc. ACL Workshop on Detecting Structure in Scholarly Discourse*, pages 27–36.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proc. ACL*, pages 1498–1507.

Longhua Qian and Guodong Zhou. 2010. Clustering-based stratified seed sampling for semi-supervised relation classification. In *Proc. EMNLP*, pages 346–355.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proc. COLING*, pages 697–704.

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

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2009. Composite kernels for relation extraction. In *Proc. ACL-IJCNLP (Short Papers)*, pages 365–368.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proc. CoNLL*.

Dan Roth and Wen-tau Yih. 2007. Global inference for entity and relation identification via a lin- ear programming formulation. In *Introduction to Statistical Relational Learning*. MIT.

Sunita Sarawagi and William W. Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Proc. NIPS*.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proc. CIKM Workshop on Automated Knowledge Base Construction*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *Proc. ACL*, pages 521–529.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proc. ACL*, pages 1640–1649.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proc. COLING (Posters)*, pages 1399–1407.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proc. ACL*, pages 1147–1157.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proc. ACL*, pages 419–426.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proc. ACL*, pages 427–434.

Guodong Zhou, Min Zhang, Dong-Hong Ji, and Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proc. EMNLP-CoNLL*, pages 728–736.

Imed Zitouni and Radu Florian. 2008. Mention detection crossing the language barrier. In *Proc. EMNLP*, pages 600–609.