Constructing Information Networks Using One Single Model

Qi Li[†] **Heng Ji**[†] **Yu Hong**[†‡] **Sujian Li**[¶]
[†]Computer Science Department, Rensselaer Polytechnic Institute, USA
[‡]School of Computer Science and Technology, Soochow University, China
[¶]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[†]{liq7,hongy2,jih}@rpi.edu, [¶]lisujian@pku.edu.cn

Abstract

In this paper, we propose a new framework that unifies the output of three information extraction (IE) tasks - entity mentions, relations and events as an information network representation, and extracts all of them using one single joint model based on structured prediction. This novel formulation allows different parts of the information network fully interact with each other. For example, many relations can now be considered as the resultant states of events. Our approach achieves substantial improvements over traditional pipelined approaches, and significantly advances state-of-the-art end-to-end event argument extraction.

1 Introduction

Information extraction (IE) aims to discover entity mentions, relations and events from unstructured texts, and these three subtasks are closely interdependent: entity mentions are core components of relations and events, and the extraction of relations and events can help to accurately recognize entity mentions. In addition, the theory of eventualities (Dölling, 2011) suggested that relations can be viewed as states that events start from and result in. Therefore, it is intuitive but challenging to extract all of them simultaneously in a single model. Some recent research attempted to jointly model multiple IE subtasks (e.g., (Roth and Yih, 2007; Riedel and McCallum, 2011; Yang and Cardie, 2013; Riedel et al., 2009; Singh et al., 2013; Li et al., 2013; Li and Ji, 2014)). For example, Roth and Yih (2007) conducted joint inference over entity mentions and relations; Our previous work jointly extracted event triggers and arguments (Li et al., 2013), and entity mentions and relations (Li and Ji, 2014). However, a single model that can extract all of them has never been studied so far.

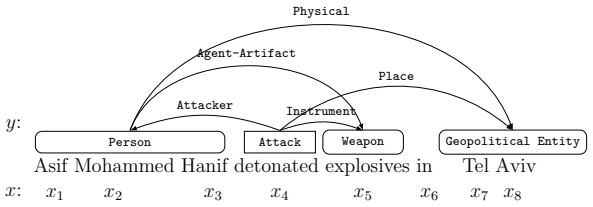


Figure 1: Information Network Representation. Information nodes are denoted by rectangles. Arrows represent information arcs.

For the first time, we uniformly represent the IE output from each sentence as an *information network*, where entity mentions and event triggers are nodes, relations and event-argument links are arcs. We apply a structured perceptron framework with a segment-based beam-search algorithm to construct the *information networks* (Collins, 2002; Li et al., 2013; Li and Ji, 2014). In addition to the perceptron update, we also apply k-best MIRA (McDonald et al., 2005), which refines the perceptron update in three aspects: it is flexible in using various loss functions, it is a large-margin approach, and it can use mulitple candidate structures to tune feature weights.

In an *information network*, we can capture the interactions among multiple nodes by learning joint features during training. In addition to the cross-component dependencies studied in (Li et al., 2013; Li and Ji, 2014), we are able to capture interactions between relations and events. For example, in Figure 1, if we know that the *Person* mention "*Asif Mohammed Hanif*" is an *Attacker* of the *Attack* event triggered by "*detonated*", and the *Weapon* mention "*explosives*" is an *Instrument*, we can infer that there exists an *Agent-Artifact* relation between them. Similarly we can infer the *Physical* relation between "*Asif Mohammed Hanif*" and "*Tel Aviv*".

However, in practice many useful interactions are missing during testing because of the data spar-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1846–1851,
October 25-29, 2014, Doha, Qatar.

sity problem of event triggers. We observe that 21.5% of event triggers appear fewer than twice in the ACE'05[1] training data. By using only lexical and syntactic features we are not able to discover the corresponding nodes and their connections. To tackle this problem, we use FrameNet (Baker and Sato, 2003) to generalize event triggers so that semantically similar triggers are clustered in the same frame.

The following sections will elaborate the detailed implementation of our new framework.

2 Approach

We uniformly represent the IE output from each sentence as an *information network* $y = (V, E)$. Each node $v_i \in V$ is represented as a triple $\langle u_i, v_i, t_i \rangle$ of start index u_i, end index v_i, and node type t_i. A node can be an entity mention or an event trigger. A particular type of node is $\perp$ (neither entity mention nor event trigger), whose maximal length is always 1. Similarly, each information arc $e_j \in E$ is represented as $\langle u_j, v_j, r_j \rangle$, where u_j and v_j are the end offsets of the nodes, and r_j is the arc type. For instance, in Figure 1, the event trigger "*detonated*" is represented as $\langle 4, 4, \texttt{Attack} \rangle$, the entity mention "*Asif Mohammed Hanif*" is represented as $\langle 1, 3, \texttt{Person} \rangle$, and their argument arc is $\langle 4, 3, \texttt{Attacker} \rangle$. Our goal is to extract the whole information network y for a given sentence x.

2.1 Decoding Algorithm

Our joint decoding algorithm is based on extending the segment-based algorithm described in our previous work (Li and Ji, 2014). Let $x = (x_1, ..., x_m)$ be the input sentence. The decoder performs two types of actions at each token x_i from left to right:

- NODEACTION(i, j): appends a new node $\langle j, i, t \rangle$ ending at the i-th token, where $i - d_t < j \leq i$, and d_t is the maximal length of type-t nodes in training data.
- ARCACTION(i, j): for each $j < i$, incrementally creates a new arc between the nodes ending at the j-th and i-th tokens respectively: $\langle i, j, r \rangle$.

After each action, the top-k hypotheses are selected according to their features $\mathbf{f}(x, y')$ and weights $\mathbf{w}$:

$$\underset{y' \in buffer}{\text{best}_\text{k}} \ \mathbf{f}(x, y') \cdot \mathbf{w}$$

Since a relation can only occur between a pair of entity mentions, an argument arc can only occur between an entity mention and an event trigger, and each edge must obey certain entity type constraints, during the search we prune invalid ARCACTIONs by checking the types of the nodes ending at the j-th and the i-th tokens. Finally, the top hypothesis in the beam is returned as the final prediction. The upper-bound time complexity of the decoding algorithm is $O(d \cdot b \cdot m^2)$, where d is the maximum size of nodes, b is the beam size, and m is the sentence length. The actual execution time is much shorter, especially when entity type constraints are applied.

2.2 Parameter Estimation

For each training instance (x, y), the structured perceptron algorithm seeks the assignment with the highest model score:

$$z = \underset{y' \in \mathcal{Y}(x)}{\text{argmax}} \, \mathbf{f}(x, y') \cdot \mathbf{w}$$

and then updates the feature weights by using:

$$\mathbf{w}^{new} = \mathbf{w} + \mathbf{f}(x, y) - \mathbf{f}(x, z)$$

We relax the exact inference problem by the aforementioned beam-search procedure. The standard perceptron will cause invalid updates because of inexact search. Therefore we apply early-update (Collins and Roark, 2004), an instance of violation-fixing methods (Huang et al., 2012). In the rest of this paper, we override y and z to denote prefixes of structures.

In addition to the simple perceptron update, we also apply k-best MIRA (McDonald et al., 2005), an online large-margin learning algorithm. During each update, it keeps the norm of the change to feature weights $\mathbf{w}$ as small as possible, and forces the margin between y and the k-best candidate z greater or equal to their loss $L(y, z)$. It is formulated as a quadratic programming problem:

$$\begin{aligned} &\min \|\mathbf{w}^{new} - \mathbf{w}\| \\ &\text{s.t. } \mathbf{w}^{new}\mathbf{f}(x, y) - \mathbf{w}^{new}\mathbf{f}(x, z) \geq L(y, z) \\ &\forall z \in best_k(x, \mathbf{w}) \end{aligned}$$

We employ the following three loss functions for comparison:

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

Freq.	Relation Type	Event Type	Arg-1	Arg-2	Example
159	Physical	Transport	Artifact	Destination	He$_{(arg-1)}$ was escorted$_{(trigger)}$ into Iraq$_{(arg-2)}$.
46	Physical	Attack	Target	Place	Many people$_{(arg-1)}$ were in the cafe$_{(arg-2)}$ during the blast$_{(trigger)}$.
42	Agent-Artifact	Attack	Attacker	Instrument	Terrorists$_{(arg-1)}$ might use$_{(trigger)}$ the devices$_{(arg-2)}$ as weapons.
41	Physical	Transport	Artifact	Origin	The truck$_{(arg-1)}$ was carrying$_{(trigger)}$ Syrians fleeing the war in Iraq$_{(arg-2)}$.
33	Physical	Meet	Entity	Place	They$_{(arg-1)}$ have reunited$_{(trigger)}$ with their friends in Norfolk$_{(arg-2)}$.
32	Physical	Die	Victim	Place	Two Marines$_{(arg-1)}$ were killed$_{(trigger)}$ in the fighting in Kut$_{(arg-2)}$.
28	Physical	Attack	Attacker	Place	Protesters$_{(arg-1)}$ have been clashing$_{(trigger)}$ with police in Tehran$_{(arg-2)}$.
26	ORG-Affiliation	End-Position	Person	Entity	NBC$_{(arg-2)}$ is terminating$_{(trigger)}$ freelance reporter Peter Arnett$_{(arg-1)}$.

Table 1: Frequent overlapping relation and event types in the training set.

- The first one is F_1 loss:

$$L_1(y, z) = 1 - \frac{2 \cdot |y \cap z|}{|y| + |z|}$$

When counting the numbers, we treat each node and arc as a single unit. For example, in Figure 1, $|y| = 6$.

- The second one is 0-1 loss:

$$L_2(y, z) = \begin{cases} 1 & y \neq z \\ 0 & y = z \end{cases}$$

It does not discriminate the extent to which z deviates from y.

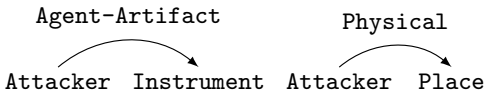
- The third loss function counts the difference between y and z:

$$L_3(y, z) = |y| + |z| - 2 \cdot |y \cap z|$$

Similar to F_1 loss function, it penalizes both missing and false-positive units. The difference is that it is sensitive to the size of y and z.

2.3 Joint Relation-Event Features

By extracting three core IE components in a joint search space, we can utilize joint features over multiple components in addition to factorized features in pipelined approaches. In addition to the features as described in (Li et al., 2013; Li and Ji, 2014), we can make use of joint features between relations and events, given the fact that relations are often ending or starting states of events (Dölling, 2011). Table 1 shows the most frequent overlapping relation and event types in our training data. In each partial structure y' during the search, if both arguments of a relation participate in an event, we compose the corresponding argument roles and relation type as a joint feature for y'. For example, for the structure in Figure 1, we obtain the following joint relation-event features:



Split	Sentences	Mentions	Relations	Triggers	Arguments
Train	7.2k	25.7k	4.8k	2.8k	4.5k
Dev	1.7k	6.3k	1.2k	0.7k	1.1k
Test	1.5k	5.3k	1.1k	0.6k	1.0k

Table 2: Data set

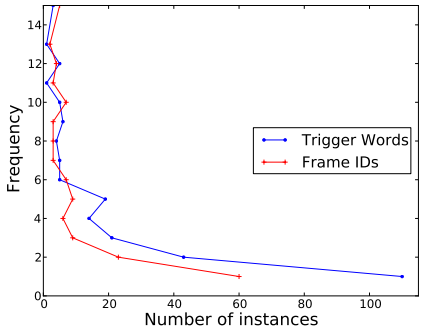


Figure 2: Distribution of triggers and their frames.

2.4 Semantic Frame Features

One major challenge of constructing information networks is the data sparsity problem in extracting event triggers. For instance, in the sentence: "*Others were* **mutilated** *beyond recognition.*" The *Injure* trigger "*mutilated*" does not occur in our training data. But there are some similar words such as "*stab*" and "*smash*". We utilize FrameNet (Baker and Sato, 2003) to solve this problem. FrameNet is a lexical resource for semantic frames. Each frame characterizes a basic type of semantic concept, and contains a number of words (lexical units) that evoke the frame. Many frames are highly related with ACE events. For example, the frame "*Cause_harm*" is closely related with *Injure* event and contains 68 lexical units such as "*stab*", "*smash*" and "*mutilate*".

Figure 2 compares the distributions of trigger words and their frame IDs in the training data. We can clearly see that the trigger word distribution suffers from the long-tail problem, while Frames reduce the number of triggers which occur only

Methods	Entity Mention (%)			Relation (%)			Event Trigger (%)			Event Argument (%)		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Pipelined Baseline	83.6	75.7	79.5	68.5	41.4	51.6	71.2	58.7	64.4	64.8	24.6	35.7
Pipeline + Li et al. (2013)				N/A			74.5	56.9	64.5	67.5	31.6	43.1
Li and Ji (2014)	85.2	76.9	80.8	68.9	41.9	52.1	N/A					
Joint w/ Avg. Perceptron	85.1	77.3	81.0	70.5	41.2	52.0	67.9	62.8	65.3	64.7	35.3	45.6
Joint w/ MIRA w/ F_1 Loss	83.1	75.3	79.0	65.5	39.4	49.2	59.6	63.5	61.5	60.6	38.9	47.4
Joint w/ MIRA w/ 0-1 Loss	84.2	76.1	80.0	65.4	41.8	51.0	65.6	61.0	63.2	60.5	39.6	47.9
Joint w/ MIRA w/ L_3 Loss	85.3	76.5	80.7	70.8	42.1	52.8	70.3	60.9	65.2	66.4	36.1	46.8

Table 3: Overall performance on test set.

once in the training data from 100 to 60 and alleviate the sparsity problem. For each token, we exploit the frames that contain the combination of its lemma and POS tag as features. For the above example, "*Cause_harm*" will be a feature for "*mutilated*". We only consider tokens that appear in at most 2 frames, and omit the frames that occur fewer than 20 times in our training data.

3 Experiments

3.1 Data and Evaluation

We use ACE'05 corpus to evaluate our method with the same data split as in (Li and Ji, 2014). Table 2 summarizes the statistics of the data set. We report the performance of extracting entity mentions, relations, event triggers and arguments separately using the standard F_1 measures as defined in (Ji and Grishman, 2008; Chan and Roth, 2011):

- An entity mention is correct if its entity type (7 in total) and head offsets are correct.
- A relation is correct if its type (6 in total) and the head offsets of its two arguments are correct.
- An event trigger is correct if its event subtype (33 in total) and offsets are correct.
- An argument link is correct if its event subtype, offsets and role match those of any of the reference argument mentions.

In this paper we focus on entity arguments while disregard values and time expressions because they can be most effectively extracted by handcrafted patterns (Chang and Manning, 2012).

3.2 Results

Based on the results of our development set, we trained all models with 21 iterations and chose the beam size to be 8. For the k-best MIRA updates, we set k as 3. Table 3 compares the overall performance of our approaches and baseline methods.

Our joint model with perceptron update outperforms the state-of-the-art pipelined approach in (Li et al., 2013; Li and Ji, 2014), and further improves the joint event extraction system in (Li et al., 2013) ($p < 0.05$ for entity mention extraction, and $p < 0.01$ for other subtasks, according to Wilcoxon Signed RankTest). For the k-best MIRA update, the L_3 loss function achieved better performance than F_1 loss and 0-1 loss on all sub-tasks except event argument extraction. It also significantly outperforms perceptron update on relation extraction and event argument extraction ($p < 0.01$). It is particularly encouraging to see the end output of an IE system (event arguments) has made significant progress (12.2% absolute gain over traditional pipelined approach).

3.3 Discussions

3.3.1 Feature Study

Rank	Feature		Weight
1	Frame=Killing	Die	0.80
2	Frame=Travel	Transport	0.61
3	Physical(Artifact, Destination)		0.60
4	w_1="home"	Transport	0.59
5	Frame=Arriving	Transport	0.54
6	ORG-AFF(Person, Entity)		0.48
7	Lemma=charge	Charge-Indict	0.45
8	Lemma=birth	Be-Born	0.44
9	Physical(Artifact,Origin)		0.44
10	Frame=Cause_harm	Injure	0.43

Table 4: Top Features about Event Triggers.

Table 4 lists the weights of the most significant features about event triggers. The 3rd, 6th, and 9th rows are joint relation-event features. For instance, *Physical(Artifact, Destination)* means the arguments of a *Physical* relation participate in a *Transport* event as *Artifact* and *Destination*. We can see that both the joint relation-event features

and FrameNet based features are of vital importance to event trigger labeling. We tested the impact of each type of features by excluding them in the experiments of "MIRA w/ L_3 loss". We found that FrameNet based features provided 0.8% and 2.2% F_1 gains for event trigger and argument labeling respectively. Joint relation-event features also provided 0.6% F_1 gain for relation extraction.

3.3.2 Remaining Challenges

Event trigger labeling remains a major bottleneck. In addition to the sparsity problem, the remaining errors suggest to incorporate external world knowledge. For example, some words act as triggers for some certain types of events only when they appear together with some particular arguments:

- "*Williams picked up the child again and this time,* **threw**$_{\texttt{Attack}}$ **her** *out the window.*"
  The word "*threw*" is used as an *Attack* event trigger because the *Victim* argument is a "*child*".
- "*Ellison to spend $10.3 billion to* **get**$_{\texttt{Merge_Org}}$ *his* **company**." The common word "*get*" is tagged as a trigger of *Merge_Org*, because its object is "*company*".
- "*We believe that the likelihood of them* **using**$_{\texttt{Attack}}$ *those* **weapons** *goes up.*"
  The word "*using*" is used as an *Attack* event trigger because the *Instrument* argument is "*weapons*".

Another challenge is to distinguish physical and non-physical events. For example, in the sentence:

- "*we are paying great attention to their ability to* **defend**$_{\texttt{Attack}}$ *on the ground.*",

our system fails to extract "*defend*" as an *Attack* trigger. In the training data, "*defend*" appears multiple times, but none of them is tagged as *Attack*. For instance, in the sentence:

- "*North Korea could do everything to* **defend** *itself.* "

"*defend*" is not an *Attack* trigger since it does not relate to physical actions in a war. This challenge calls for deeper understanding of the contexts.

Finally, some pronouns are used to refer to actual events. Event coreference is necessary to recognize them correctly. For example, in the following two sentences from the same document:

- "*It's important that people all over the world know that we don't believe in the* **war**$_{\texttt{Attack}}$.",
- "*Nobody questions whether* **this**$_{\texttt{Attack}}$ *is right or not.*"

"*this*" refers to "*war*" in its preceding contexts. Without event coreference resolution, it is difficult to tag it as an *Attack* event trigger.

4 Conclusions

We presented the first joint model that effectively extracts entity mentions, relations and events based on a unified representation: information networks. Experiment results on ACE'05 corpus demonstrate that our approach outperforms pipelined method, and improves event-argument performance significantly over the state-of-the-art. In addition to the joint relation-event features, we demonstrated positive impact of using FrameNet to handle the sparsity problem in event trigger labeling.

Although our primary focus in this paper is information extraction in the ACE paradigm, we believe that our framework is general to improve other tightly coupled extraction tasks by capturing the inter-dependencies in the joint search space.

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, Disney 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

Collin F. Baker and Hiroaki Sato. 2003. The framenet data and software. In *Proc. ACL*, pages 161–164.

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

Angel X. Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Proc. LREC*, pages 3735–3740.

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.

Johannes Dölling. 2011. Aspectual coercion and eventuality structure. pages 189–226.

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. 2008. Refining event extraction through cross-document inference. In *Proc. ACL*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proc. ACL*.

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

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. ACL*, pages 91–98.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proc. EMNLP*.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *Proc. the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*.

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.

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*.

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