Joint Inference for Event Coreference Resolution

Jing Lu[1] and **Deepak Venugopal**[2] and **Vibhav Gogate**[1] and **Vincent Ng**[1]
[1]Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080
[2]Department of Computer Science, The University of Memphis, Memphis, TN 38152
jxl125430@utdallas.edu, dvngopal@memphis.edu,
{vgogate,vince}@hlt.utdallas.edu

Abstract

Event coreference resolution is a challenging problem since it relies on several components of the information extraction pipeline that typically yield noisy outputs. We hypothesize that exploiting the inter-dependencies between these components can significantly improve the performance of an event coreference resolver, and subsequently propose a novel joint inference based event coreference resolver using Markov Logic Networks (MLNs). However, the rich features that are important for this task are typically very hard to explicitly encode as MLN formulas since they significantly increase the size of the MLN, thereby making joint inference and learning infeasible. To address this problem, we propose a novel solution where we implicitly encode rich features into our model by augmenting the MLN distribution with low dimensional unit clauses. Our approach achieves state-of-the-art results on two standard evaluation corpora.

1 Introduction

Within-document event coreference resolution is the task of determining which event mentions in a text refer to the same real-world event. Event coreference is arguably more challenging and less studied than entity coreference. The challenge stems in part from the fact that an event coreference resolver typically lies towards the end of the standard information extraction (IE) pipeline, assuming as input the noisy outputs of its upstream components. Specifically, a standard event coreference resolver takes as input the extracted event triggers, their arguments, and the entity coreference information, and aggregates this information through rules to resolve coreferent event mentions. Each component of this pipeline can introduce errors that naturally propagate to the event coreference resolver, thereby significantly affecting its performance. Further, the aforementioned pipeline architecture also fails to exploit inter-dependencies between the various components that can provide valuable insights to the resolver.

In light of these weaknesses, we propose a novel approach to within-document event coreference resolution based on Markov Logic Networks (MLNs) (Domingos and Lowd, 2009). In our approach, we jointly perform four key tasks in the IE pipeline: trigger identification and subtyping, argument identification and role determination, entity coreference resolution, and event coreference resolution. To our knowledge, this is the first attempt to design an MLN for event coreference resolution. MLNs are particularly well-suited for modeling *joint inference* tasks in natural language processing (NLP) due to the inherent relational structure and uncertainty typically associated with challenging NLP problems.

A major obstacle to the successful application of MLNs to NLP tasks is the high computational complexity of probabilistic inference and learning algorithms. The MLNs used in NLP are so large that even linear time inference algorithms are computationally infeasible. For instance, the rich sets of features that are typically used to solve the four tasks in the IE pipeline for event coreference, are ill-suited for modeling as explicit MLN formulas, since they will yield a large MLN having millions of features. Therefore, a major contribution of our work lies in the proposal of a novel hybrid approach where we embed such features as weighted unit clauses in a low-dimensional space, and then integrate these clauses with the rest of the MLN formulas during inference. Since this idea is generally applicable to modeling NLP tasks using MLNs, we believe that our work will be of interest to other NLP researchers as well.

Georges Cipriani$_{[Person]}$, a former militant of the French far-left group Action Directe, {left}$_{ev1}$ the prison$_{[Origin]}$ in Ensisheim in northern France on parole on Wednesday$_{[Time]}$. He$_{[Person]}$ {left}$_{ev2}$ Ensisheim$_{[Origin]}$ in a police vehicle$_{[Instrument]}$ bound for an open prison near Strasbourg.

Table 1: Event coreference resolution example.

We evaluate our approach on corpora involving two languages, the new KBP 2015 English corpus and the Chinese portion of the ACE 2005 corpus. On both corpora, our approach performs significantly better than the baseline pipeline-based resolver. In particular, on the KBP corpus, we achieve the best result reported to date surpassing the previous best result by around 0.43 percentage points in average F1-score.

2 Definitions and Corpora

2.1 Definitions

We employ the following definitions in our discussion of event extraction and coreference:

- An **event mention** is an explicit occurrence of an event consisting of a textual trigger, arguments or participants (if any), and the event type/subtype.
- An **event trigger** is a string of text that most clearly expresses the occurrence of event, usually a word or a multi-word phrase
- An **event argument** is an argument filler that plays a certain role in an event.
- An **event coreference chain** (a.k.a. an **event hopper**) is a group of event mentions that refer to the same real-world event. They must have the same event (sub)type.

To understand these definitions, consider first the example shown in Table 1, which contains two event mentions, $ev1$ and $ev2$. Here, *left* is the trigger for both $ev1$ and $ev2$ with subtype Movement.Transport-Person. $ev1$ has three arguments, *Georges Cipriani*, *prison*, and *Wednesday* with roles *Person*, *Origin*, and *Time* respectively. $ev2$ also has three arguments, *He* and *Ensisheim*, and *police vehicle* with roles *Person*, *Origin*, and *Instrument* respectively.

2.2 Corpora

We employ two text corpora in two languages for evaluation. The *English* corpus was used in the Event Nugget Detection and Coreference task in the TAC KBP 2015 Event Track (henceforth the KBP 2015 corpus). This corpus is composed of two types of documents, newswire documents and discussion forum documents. The training set consists of 158 documents with 6538 event mentions distributed over 3335 event coreference chains, and the test set consists of 202 documents with 6438 event mentions distributed over 4125 event coreference chains. The *Chinese* corpus is the Chinese portion of the ACE 2005 training corpus. This corpus is composed of documents taken from six sources, and consists of 633 documents with 3333 event mentions distributed over 2521 event coreference chains. Note that ACE and KBP employ slightly different event ontologies: ACE defines 33 event subtypes and KBP defines 38 event subtypes, among which 31 subtypes are shared by both ontologies.

2.3 Key Differences between ACE and KBP

While both ACE and KBP rely on the aforementioned definitions, the guidelines they employ when annotating triggers and event coreference chains are slightly different. Below we highlight the differences that are relevant to our discussion.[1]

First, there are slight differences w.r.t. the annotation of triggers. ACE only allows single-word triggers, whereas KBP additionally allows multi-word triggers (e.g., *laid off*). Also, each word in ACE may trigger at most one event mention, whereas each (multi-)word in KBP may trigger multiple event mentions (e.g., *murder* can trigger two event mentions with subtypes Life.Die and Conflict.Attack).

[1]For detailed definitions, see http://cairo.lti.cs.cmu.edu/kbp/2015/event/annotation and http://www.itl.nist.gov/iad/mig/tests/ace/2005/ for the definitions of event coreference adopted by KBP 2015 and ACE 2005 respectively.

Second, KBP adopts a more relaxed definition of event coreference than ACE. Specifically, KBP requires that two event mentions be coreferent as long as they *intuitively* refer to the same real-world event. In our running example, $ev1$ and $ev2$ are coreferent according to KBP because they both refer to the same event of Cipriani leaving the prison. ACE, on the other hand, *additionally* requires that the corresponding *arguments* in the two event mentions be coreferent. In the example, $ev1$ and $ev2$ are *not* coreferent according to ACE because their *Origin* arguments are not coreferent (one Origin argument involves a prison and the other involves the city Ensisheim). Note that determining whether two entity mentions are coreferent is the task of entity coreference. Like event mentions, entity mentions have corpus-specific *entity types*.

3 Background

In this section, we give a brief overview of MLNs and discuss related work in event coreference resolution.

3.1 Markov Logic Networks

Formally, an MLN $\mathcal{M}$ is a set of pairs (f_i, θ_i) where f_i is a formula in first-order logic and θ_i is a real number. Given a set of constants, an MLN represents a ground Markov network, in which we have one binary random variable for each possible ground atom and one propositional feature for each possible grounding of each first-order formula. The weight associated with the feature is the weight attached to the corresponding formula. The ground Markov network represents the following probability distribution:

$$P_{\mathcal{M}}(\omega) = \frac{1}{Z} \exp\left(\sum_{f_i} \theta_i N_{f_i}(\omega)\right) \quad (1)$$

where $N_{f_i}(\omega)$ is the number of groundings of f_i that evaluate to True given a world ω (an assignment of $\{0, 1\}$ to all ground atoms). The use of first-order logic enables the user to succinctly represent prior, relational knowledge about the application domain, while the weights help model uncertainty in the truth of the first-order logic sentences.

3.2 Related Work

Existing within-document English event coreference resolvers have been evaluated on different corpora, such as MUC (e.g., Humphreys et al. (1997)), ACE (e.g., Ahn (2006), McConky et al. (2012), Chen and Ji (2009), S. and Arock (2012)), OntoNotes (e.g., Chen et al. (2011)) the (not publicly-available) Intelligence Community (IC) corpus (e.g., Cybulska and Vossen (2012), Araki et al. (2014)); the ECB corpus (e.g., Bejan and Harabagiu (2010; 2014), Lee et al. (2012)) and its extension, ECB+ (e.g.,Yang et al. (2015)); and ProcessBank (e.g., Araki and Mitamura (2015)). The newest event coreference corpus is the one used in the KBP 2015 Event Nugget Detection and Coreference shared task, in which the best performers are RPI's system (Hong et al., 2015), LCC's system (Monahan et al., 2015), and UI-CCG's system (Sammons et al., 2015). Among these corpora, ACE is the only one that is additionally composed of event coreference-annotated Chinese documents. It has been used to train SinoCoreferencer (Chen and Ng, 2014), a publicly-available Chinese event coreference resolver. Not all such corpora were carefully annotated: as Liu et al. (2014) pointed out, OntoNotes and ECB have only been partially annotated with event coreference links, for instance.

4 Baseline System

Our pipeline-based baseline system has five steps:

Step 1: Entity extraction. Our entity extraction model jointly identifies the entity mentions and their entity types. We train this model using CRF++[2], treating each sentence as a word sequence. Specifically, we create one instance for each word w and assign it a class label that indicates whether it begins an entity mention with type t_j (B-t_j), is inside an entity mention with type t_j (I-t_j), or is outside an entity mention

[2] https://taku910.github.io/crfpp/

(O). The features used to represent each instance for training the English CRF and the Chinese CRF are shown in Tables 2(a) and 3(a), respectively.

Step 2: Entity coreference resolution. Our entity coreference classifier is a pairwise classifier that determines whether two entity mentions are coreferent or not. To train this classifier, we employ SVM^{light} (Joachims, 1999), creating training instances using Soon et al.'s (2001) training instance creation method. Each training instance represents two entity mentions in each training document. The class value of a training instance is either positive or negative, depending on whether the two entity mentions are coreferent in the associated text. The features used to represent each instance for training the entity coreference classifiers for English and Chinese are shown in Tables 2(b) and 3(b), respectively.

After training, the resulting classifier can be used to classify each pair of entity mentions extracted in Step 1 as coreferent or not. We select as the antecedent of an entity mention em the closest preceding mention that is classified as coreferent with em.

Step 3: Trigger identification and subtyping. Since ACE allows only single-word triggers, our SVM-based Chinese trigger classifier takes as input a candidate trigger *word* (i.e., a word that survives Li et al.'s (2012) filtering rules) and outputs its event subtype (if it is a true trigger) or *None* (if it is not a trigger). In essence, it jointly (1) identifies event trigger words and (2) assigns a subtype to each identified trigger. To train this classifier, we create one training instance for each word w_i in each training document. If the word does not correspond to a trigger, the class label of the corresponding instance is *None*. Otherwise, the class label is the subtype of the trigger. The features used to represent each instance for training this classifier are shown in Table 3(c).

Because KBP additionally allows multi-word triggers, we recast the task of identifying English triggers as a sequence labeling task, where we train models using CRF++. Recall that since each (multi-)word may trigger multiple event mentions having different (sub)types, we train one CRF for each type. Specifically, to train the CRF for type t_j, we create one instance for each word w_i, assigning it a class label that indicates whether it begins a trigger with subtype s_{jk} (B-s_{jk}), is inside a trigger with subtype s_{jk} (I-s_{jk}), begins a trigger with other types (B-$\mathrm{t}_{m\neq j}$), is inside a trigger with other types (I-$\mathrm{t}_{m\neq j}$) or is outside a trigger (O). The features used to represent each instance for training this CRF are shown in Table 2(c). To improve the recall of event trigger detection, we augment the CRF output with heuristically extracted triggers. Specifically, we first construct a wordlist containing triggers that appear infrequently (less than 10 times) in the training data and do not belong more than one subtype according to the training data. Then, we extract any word as a trigger with the corresponding subtype as long as it appears in the wordlist.

Step 4: Argument identification and role labeling. Our argument identifier and role labeler is a classifier trained using SVM^{light} that jointly learns the tasks of (1) identifying the true arguments of an event mention and (2) assigning a role to each of its true arguments. To train this classifier, we create the training instances by pairing each true event mention em (i.e., event mention consisting of a true trigger) with each of em's candidate event arguments, considering an entity mention extracted in Step 1 a candidate argument of em if it appears in the same sentence as em. If the candidate argument is indeed a true argument of em, the class label of the training instance is the argument's role. Otherwise, its class label is *None*. The features used to represent each instance for training the English classifier and the Chinese classifier are shown in Tables 2(d) and 3(d), respectively.

After training, we can apply this classifier to classify test instances. To create test instances, we pair each *candidate* trigger (extracted in Step 3) with each of its candidate event arguments.

Step 5: Event coreference resolution. The event coreference classifier is a pairwise classifier that determines whether two event mentions are coreferent. To train this classifier, we use SVM^{light}, creating training instances using Soon et al.'s (2001) training instance creation method. The features used to represent each instance for training the event coreference classifier for English and Chinese are shown in Tables 2(e) and 3(e), respectively.

After training, we apply the resulting classifier to classify test instances. We select as the antecedent of an extracted event mention e the closest preceding mention that is classified as coreferent with e.

(a) Features for entity extraction. w is the word under consideration.

Lexical	word unigrams, bigrams, and trigrams formed from w with a window size of five.
Grammatical	w's part-of-speech (POS) tag; whether w is part of a NP; whether w is part of a pronoun, whether w is capitalized.
Semantic	the WordNet synset id of w; the WordNet synset ids of w's hypernym, its parent, and its grandparent.

(b) Features for entity coreference resolution. en_2 is the entity mention to be resolved and en_1 is a candidate antecedent of en_2.

Lexical	whether en_1 is a pronoun; whether en_1 is the subject of the sentence; whether en_1 is a noun; whether en_2 is a pronoun; whether en_1 is a noun; whether en_1 and en_2 have the exactly the same string; whether the modifiers of en_1 and en_2 match; the sentence distance between the strings of $en1$ and en_2.
Grammatical	the number, gender and animacy of en_1 and en_2; whether en_1 and en_2 agree w.r.t. number; whether en_1 and en_2 agree w.r.t. gender; whether en_1 and en_2 agree w.r.t. animacy.

(c) Features for event trigger identification and subtyping. t is the candidate trigger.

Lexical	t's POS tag, lemmatized and unlemmatized word unigrams, word bigrams, and word trigrams formed from t with a window size of five.
Syntactic	depth of t in its syntactic parse tree; path from the leaf node of t to the root in its syntactic parse tree; phrase structure expanded by the parent of t's node; phrase type of t's node.
Semantic	WordNet synset id of t; WordNet synset ids of t's hypernym, its parent, and its grandparent.

(d) Features for event argument identification and role labeling. en is a candidate argument of trigger t.

Basic	t's event subtype; en's entity type; en's head word; event subtype + head word; event subtype + entity type; t's POS tag.
Neighboring words	left/right neighbor word of en; left/right neighbor word of en + the word's POS; left/right neighbor word of en + the word's POS.
Syntactic	the phrase structure obtained by expanding the parent of t in the constituent parse tree; the phrase type of t; the path from en to t in the constituent parse tree; the dependency path from en to t.

(e) Features for event coreference resolution. ev_2 is the event mention to be resolved and ev_1 is a candidate antecedent of ev_2.

Event type features	whether ev_1 and ev_2 agree w.r.t. event type; whether they agree w.r.t. event subtype; the concatenation of their event types; and the concatenation of their event subtypes.
Trigger features	whether ev_1 and ev_2 have the same trigger; whether they have the same lemmatized trigger; whether the triggers of ev_1 and ev_2 or the hypernyms of these triggers are in the same WordNet synset; the concatenation of their triggers; the concatenation of POS tags of their triggers; whether their triggers agree in number if they are nouns; whether their triggers have the same modifiers and they are in the same entity coreference chain if they are nouns; the sentence distance between the triggers of ev_1 and ev_2; whether the triggers of ev_1 and ev_2 appear in a training document as a coreferent event mention pair; whether the triggers of ev_1 and ev_2 appear in the first sentence and headline if this is a newswire document; whether the sentence containing the the triggers of ev_1 and ev_2 are identical if this is a discussion forum document.
Argument features	whether ev_1 and ev_2 have arguments with the same role; whether the arguments have the same head word; whether they are in the same coreference chains; whether they have the same modifiers; the roles and number of the arguments that only appear in ev_1; and the roles and number of the arguments that only appear in ev_2.

Table 2: Features used in the English baseline system. POS tags, constituent parses and dependency parses are provided by CoreNLP (Manning et al., 2014). For all uses of WordNet (Fellbaum, 1998), only the first synset is used.

5 Joint Model

In this section, we describe our MLN-based joint model for event coreference resolution.

5.1 MLN Structure

Figure 1 shows our proposed MLN for event coreference resolution. It has five predicates subdivided into three categories: query, hidden and evidence.

The *query* predicate `EventCoref`(d,t_1,t_2) is true when two event mentions t_1 and t_2 in document d are coreferent. The *hidden* predicates are those that cannot be directly observed in the data. Our model contains three hidden predicates: (1) `Trigger`(d,t,p) is true when mention t in document d has event/trigger subtype p. A special type called "None" indicates that t does not contain a trigger. (2) `Argument`(d,t,a,r) asserts that entity mention a is an argument of event mention t in document d and its role is r. Again, we include a special role called "None", which indicates that the entity mention is not an argument of the event mention. The ! symbol in the predicate definition indicates that every entity mention must take one and only one argument role. (3) `EntityCoref`(d,a_1,a_2) is true when entity mentions a_1 and a_2 in document d are coreferent. The *evidence* predicates represent (ground) random variables

(a) Features for entity extraction. w is the word under consideration.

Lexical	word unigrams, bigrams, and trigrams formed from w with a window size of five.
Grammatical	w's POS tag; whether w is in a NP; whether w is part of a pronoun.
Wordlist-based	whether w can be found in each of the following 10 wordlists: Chinese surnames; famous GPE and location names (three wordlists); Chinese location suffixes; Chinese GPE suffixes; famous international organization names; famous company names; famous person names; and a list of pronouns.

(b) Features for entity coreference resolution. en_2 is an entity mention to be resolved and en_1 is a candidate antecedent of en_2.

Lexical	whether en_1 is a pronoun; whether en_1 is the subject of the sentence; whether en_1 is a noun; whether en_2 is a pronoun; whether en_1 is a noun; whether en_1 and en_2 are the same string; whether the modifiers of en_1 and en_2 match; the sentence distance between $en1$ and en_2.
Grammatical	the number, gender and animacy of en_1 and en_2; whether en_1 and en_2 agree w.r.t. number; whether en_1 and en_2 agree w.r.t. gender; whether en_1 and en_2 agree w.r.t. animacy.

(c) Features for event trigger identification and subtyping. t is a candidate trigger.

Lexical	word and POS n-grams formed from t with a window size of three
Syntactic	depth of t in its syntactic parse tree; path from the leaf node of t to the root in its syntactic parse tree; phrase structure expanded by the parent of t's node; the path from the leaf node of t to the governing clause; phrase type of t's node.
Semantic	whether t exists in a predicate list from the Chinese PropBank (Xue and Palmer, 2009); the entry number of t in a Chinese synonym dictionary.
Closest entity information	entity type of the syntactically/textually nearest entity to t in its syntactic parse tree; entity type of the syntactically/textually left/right nearest entity to t in its syntactic parse tree + entity.

(d) Features for event argument identification and role labeling. en is a candidate argument of trigger t.

Basic	t's event subtype; en's entity type; en's head word; t's subtype + en's head word; t's event subtype + en's entity type; t's POS tag.
Neighboring words	left/right neighbor word of en; left/right neighbor word of en + the word's POS tag; left/right neighbor word of t + the word's POS tag.
Syntactic	the phrase structure obtained by expanding the parent of t in the constituent parse tree; the phrase type of t; the path from en to t in the constituent parse tree; the dependency path from en to t.

(e) Features for event coreference resolution. ev_2 is the event mention to be resolved and ev_1 is a candidate antecedent of ev_2.

Event type features	whether ev_1 and ev_2 agree w.r.t. event type; whether they agree w.r.t. event subtype; the concatenation of their event types; and the concatenation of their event subtypes.
Trigger features	whether ev_1 and ev_2 have the same trigger; whether the trigger of ev_1 and ev_2 partially matched; whether they have the same lemmatized trigger; the concatenation of their triggers; the concatenation of part-of-speech tags of their triggers; whether their triggers agree in number if they are nouns; whether their triggers have the same modifiers if they are nouns; the sentence distance between the triggers of ev_1 and ev_2; the number of words between ev_1 and ev_2; whether the triggers of ev_1 and ev_2 appear in a training document as a coreferent event mention pair.
Argument features	whether ev_1 and ev_2 have arguments of the same role; whether the arguments have the same head word; whether they are in the same coreference chains; whether they have the same modifiers; the roles and number of the arguments that only appear in ev_1; and the roles and number of the arguments that only appear in ev_2.

Table 3: Features used in the Chinese baseline system. POS tags, constituent parses, and dependency parses are provided by CoreNLP (Manning et al., 2014). A detailed description of the wordlists used in the wordlist-based features can be found in Chen and Ng (2016). The Chinese synonym dictionary is HIT-SCIR's Tongyici cilin (extended).[3]

that can be directly observed in the data. In our MLN, we assume that we only observe the words; the predicate `Word`(d,t,w) is true when mention t in document d equals word w.

The MLN formulas are of two types. The first six formulas have infinite weight, which means that they are hard formulas and must always be satisfied. The last two formulas are soft, and their weights will be learned from the data. All logical variables in our formulas are universally quantified and therefore for brevity, we do not use them in the formulas. Formula 1 encodes the hard constraint that if two event mentions are coreferent, then they should share the same trigger subtype. Formula 2 specifies the hard constraint that if event mentions are coreferent, then their triggers subtypes cannot be "None." Formulas 3−6, all of which are hard formulas, specify the commutative and transitive properties of coreferent event and entity mentions. Formula 7, which is a soft formula, specifies the following dependency between coreferent entity mentions and coreferent event mentions: for two event mentions t_1 and t_2 having the same trigger subtype, if there exists an argument role r that is filled by argument a_1 in t_1

[3] `http://ir.hit.edu.cn/`

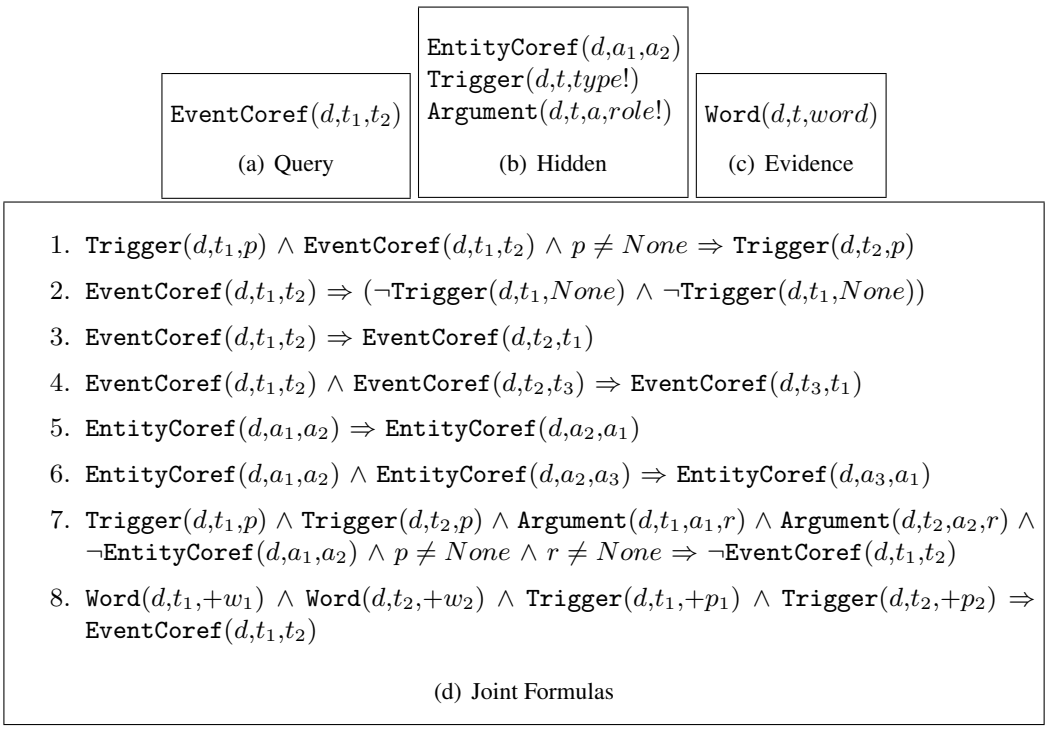

Figure 1: MLN structure.

and by a_2 in t_2, then t_1 and t_2 are not event coreferent if a_1 and a_2 are not entity coreferent.[4] Formula 8, which is also a soft formula, encodes the dependency between words in the text, trigger subtypes and event coreference. The $+$ sign in this formula indicates that for every grounding of the variables marked by the $+$ sign, we use a different weight for the soft formula.

5.2 Augmenting the MLN Distribution

Notice that the MLN shown in Figure 1 does not model the features used in the baseline systems. These features typically have high dimensionality and encoding them directly in the MLN is quite inefficient. For example, describing a trigram as an MLN formula results in d^3 ground formulas, where d is the number of words in our vocabulary. Therefore, the ground Markov network of an MLN that explicitly models all such high dimensional features would be extremely large and infeasible for inference. To address this issue, we implicitly encode the high-dimensional features by embedding them as weighted unit clauses, one for each grounding of the hidden and query predicates. Specifically, for each hidden/query ground atom X_i, we derive a weight $\phi(X_i)$ using the baseline system. This weight is computed as the distance from the hyperplane for the SVM-based classifiers and as a probability value for the CRF-based classifiers in the baseline system. We normalize each weight between the interval $[-1,1]$. The modified MLN distribution incorporating the new unit clauses is given by

$$P_{\mathcal{M}'}(\omega) \propto \exp\left(\sum_{f_i} \theta_i N_{f_i}(\omega)\right) \Phi(\omega) \tag{2}$$

where ω is a world (assignment on every ground atom) and $\Phi(\omega)$ acts as a prior on the set of hidden (**H**) and query (**Y**) ground atoms in the original MLN and is given by,

$$\Phi(\omega) = \exp\left(\sum_{X \in \mathbf{H} \cup \mathbf{Y}} \mathbb{I}_X(\omega)\phi(X)\right)$$

[4] According to the event coreference task definitions, arguments with certain roles cannot satisfy Formula 7. Hence, to reduce memory requirements, we restrict the application of Formula 7 to arguments having the following roles: Position, Person, Entity, Organization, Attack, Defendant, Adjudicator, Giver, Agent, Target, and Thing. In addition, we make it a soft (rather than hard) formula in view of the noisy outputs of our entity coreference resolver.

where $\mathbb{I}_X(\omega)$ is an indicator function that is equal to 1 if X is true in ω and 0 otherwise.

5.3 Setting the Soft Formula Weights

During inference time, we dynamically set the weights for the soft formulas (Formulas 7 and 8 in Figure 1) as follows. For each ground soft formula where its evidence atoms do not make it false, we set its weight to be the sum of the (normalized) SVM weights or CRF probabilities corresponding to its hidden and query atoms. We then multiply the soft weights with hyper-parameters η_1 and η_2 for Formulas 7 and 8 respectively and tune η_1 and η_2 using a grid search over the values $\{0.1, 0.25, 0.5, 0.75, 1.0\}$ to optimize the F1-score of event coreference resolution on the development set.

5.4 Inference

Given the prior-augmented MLN, $\mathcal{M}'$, the key task we are interested in is finding a truth assignment to all ground atoms of `EventCoref` that has the maximum probability given evidence on all ground atoms of `Word`. The following standard MAP inference task, which computes a joint assignment to all hidden and query variables given evidence, can be used to find the desired truth assignment.

$$\arg\max_{\omega} \left\{ \exp\left(\sum_{f_i} \theta_i N_{f_i}(\omega) \right) \Phi(\omega) \right\} \tag{3}$$

Unfortunately, the optimization problem given above is NP-hard in general. Moreover, the number of possible worlds in $\mathcal{M}'$ is extremely large and as a result naively searching over this large space (in order to solve the optimization problem) is computationally infeasible. As a concrete example, for the KBP 15 training dataset, we have 50 million ground atoms.

Fortunately, we can exploit the structure of the MLN given in Figure 1 in order to scale up MAP inference. In particular, the subset of ground atoms corresponding to two distinct documents are independent of each other. More formally, let $\mathbf{X}_i$ and $\mathbf{X}_j$ be the subset of ground atoms corresponding to two documents, say D_i and D_j respectively, then $\mathbf{X}_i$ is conditionally independent of $\mathbf{X}_j$ given evidence. Thus, given D documents in our corpus, the joint distribution represented by our MLN can be expressed as a product of D distributions. We can then perform inference independently over each such distribution, which greatly reduces the complexity of inference. Our inference procedure therefore follows an efficient, lazy, semi-lifted grounding strategy (Gogate and Domingos, 2011) that grounds the MLN for each document independently and solves Eq. (3) for each document separately using Gurobi (2013), a state-of-the-art integer linear programming solver.

6 Evaluation

6.1 Experimental Setup

We perform our evaluation on two corpora, the KBP 2015 English corpus and the Chinese portion of the ACE 2005 training corpus. For English, we train models on 128 of the training documents, tune parameters (the regularization parameters in SVM classifiers and the weights of the soft MLN formulas) on the remaining 30 training documents, and report results on the official test set.[5] For Chinese, since the ACE 2005 test set is not publicly available, we report five-fold cross validation results on the ACE 2005 training corpus. For each fold experiment, we employ three folds for classifier training, one fold for development (parameter tuning), and one fold for testing.

To evaluate event coreference performance on KBP, we follow the official KBP evaluation and employ four commonly-used scoring measures as implemented in version 1.7 of the official scorer provided by the KBP 2015 organizers, namely MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF_e (Luo, 2005) and BLANC (Recasens and Hovy, 2011), as well as the unweighted average of their F-scores.[6]

[5] Since the KBP 2015 corpus was not annotated with event arguments and entity coreference links, we train our entity mention extractor, our entity coreference resolver, and our event argument identification and role classification model on two LDC corpora provided by the TAC KBP 2015 task organizers (LDC2015E29 and LDC2015E68), as permitted by the rules of the shared task.

[6] The official KBP scorer is available at `http://cairo.lti.cs.cmu.edu/kbp/2015/event/scoring`.

Metric	English/KBP 2015						Chinese/ACE 2005					
	Baseline			**MLNs**			**Baseline**			**MLNs**		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
B^3	53.48	39.21	45.20	50.27	41.63	**45.54**	38.21	37.93	37.66	36.87	42.54	**39.50**
CEAF_e	42.33	38.54	**40.35**	47.53	33.48	39.29	40.28	37.76	38.98	41.02	41.19	**41.10**
MUC	50.52	29.13	36.96	47.07	38.21	**42.18**	40.02	40.27	40.14	39.37	44.70	**41.86**
BLANC	41.16	26.17	32.00	40.61	28.96	**33.30**	24.75	25.67	25.20	22.41	29.07	**25.29**
	Average = 38.64			**Average = 40.08**			CoNLL = 39.02			**CoNLL = 40.82**		

Table 4: Results for event coreference resolution on KBP 2015 and ACE 2005.

English/KBP 2015						Chinese/ACE 2005					
Baseline			**MLNs**			**Baseline**			**MLNs**		
Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
65.05	51.43	57.45	67.97	50.51	**57.95**	67.08	56.44	61.30	66.39	57.37	**61.55**

Table 5: Results for event trigger identification and subtyping on KBP 2015 and ACE 2005.

To evaluate event coreference performance on ACE, we follow previous work on event coreference (e.g., Yang et al. (2015)) and employ the aforementioned four scoring measures as implemented in the latest version (v8) of the CoNLL scorer (Pradhan et al., 2014), as well as the CoNLL score, which is the unweighted average of the MUC, B^3, and CEAF_e F-scores.[7] To our knowledge, there is only one difference between the implementations of the four scoring measures in the two scorers: while the CoNLL scorer considers an event mention correctly detected as long as it has an exact match with a gold event mention in terms of its left and right boundaries, the KBP 2015 scorer is stricter in that it considers an event mention correctly detected by additionally requiring that its event subtype be correctly determined.

6.2 Results and Discussion

The left half of Table 4 shows the results for English event coreference resolution on the KBP 2015 dataset. As can be seen, MLNs outperform the baseline system when evaluated on all but the CEAF_e metrics. W.r.t. the Average metric, MLNs achieve an F-score of 40.08, outperforming the baseline significantly by 1.44 points (paired t-tests, $p < 0.05$). To our knowledge, this is the best result reported to date on this corpus, with the top system in the KBP 2015 shared task achieving an Average F-score of 39.65. In general, the MLN could detect more event coreference chains than the baseline system, as seen from its higher recall in all but the CEAF_e metrics.[8]

The right half of Table 4 shows the results for event coreference resolution on the ACE 2005 Chinese corpus. As can be seen, MLNs outperform the baseline significantly by 1.8 points w.r.t. the CoNLL metric. In fact, MLNs achieve a higher score than the baseline w.r.t. each of the four scoring measures. Similar to what we observed on the KBP corpus, the consistently superior performance achieved by the MLN-based resolver can be attributed to its substantially higher recall accompanied by a slightly lower precision. In particular, since MUC is a link-based metric, the fact that the MLNs achieve a higher MUC recall on both datasets suggest that the MLNs are better at discovering event coreference links than the baseline.

One may argue that the MLNs may *not* be better than the baseline at discovering event coreference links: it may simply be the case that the joint inference process has allowed additional triggers to be extracted, which in turn allowed additional event coreference links to be established. To understand whether this is indeed the case, we compute the results for trigger identification and subtyping in Table 5. As can be seen, fewer English triggers are extracted after joint inference, whereas the reverse is true for Chinese. These results suggest that at least for English, the higher event coreference recall achieved by the MLNs is not attributable to better trigger identification and subtyping.

A closer examination of the outputs reveals that our resolver is comparatively better at extracting two types of coreference links that are traditionally considered difficult to extract. The first type involves triggers that are lexically different. For example, in the text segment "The former mayor of Detroit,

[7] The CoNLL scorer is available at `https://github.com/conll/reference-coreference-scorers`.

[8] As is commonly known, CEAF_e sometimes produces unintuitive scores. Specifically, the CEAF_e F-score may drop as more coreference links are correctly identified. See Moosavi and Strube (2016) for a detailed discussion.

Michigan was sentenced to 28 years in prison . . . Prosecutors asked for a minimum of 28 years for Kilpatrick, who resigned from the mayor's office in 2008 . . .", the link between event mentions triggered by *former* and *resigned*, both of which have type Personnel.End-position, is discovered by our resolver but not the baseline. The second type involves links between event mentions that are far from each other.

6.3 Error Analysis

To better understand how to improve our MLN-based resolver and to provide directions for future work, we conduct a qualitative analysis of its major sources of error in this subsection.

6.3.1 Two Major Types of Precision Error

Erroneous triggers. For both languages, our trigger classifier had difficulties with correctly classifying certain frequently-occurring words that are sometimes used as triggers and sometimes not. Specifically, the classifier misclassified many non-trigger instances of these words as triggers, which were subsequently used to establish coreference links by our resolver. A particularly interesting and challenging example involves the word "violent". Consider two sentences that appear in the same document: "The violent arrest of Ahmed al-Alwani is likely to inflame tensions in Sunni-dominated Anbar" and "Iraq troops arrest leading Sunni MP in violent raid". The first sentence contains two event mentions, one triggered by *violent* with type Conflict.Attack and the other triggered by *arrest* with type Justice.Arrestjail. The second sentence, contains only one event mention: it is triggered by *raid* with type Conflict.Attack and is coreferent with *violent*. While our system successfully detects all three triggers, it also erroneously detects *violent* in the second sentence as a trigger. This error gets propagated to our event coreference resolver, which posits the two occurrences of *violent* as coreferent.

Failure to extract arguments. Recall that our argument classifier does not extract any argument of an event mention that does not appear in the same sentence as its trigger. This severely limits its ability to extract arguments and has caused many spurious event coreference links to be established. For instance, our resolver erroneously posits two *violence* events as coreferent: it does not know that the two events took place in different countries, as the argument classifier failed to extract their location arguments (one is *Honduras* and the other is *Venezuela*).

6.3.2 Two Major Types of Recall Error

Missing triggers. For both languages, the trigger classifier failed to identify trigger words/phrases that are unseen or rarely-occurring in the training data. As a result, many links cannot be established.

Insufficient knowledge. Recall that our MLN-based resolver has achieved a higher recall than the baseline by doing a better job at establishing links between event mentions containing lexically different triggers. However, there are still many links between event mentions with lexically different triggers that our resolver fails to discover owing to the insufficient knowledge made available to it. This type of error is especially prominent on the Chinese corpus.

7 Conclusion

We proposed a novel joint inference based event coreference resolver using MLNs. Since encoding rich NLP features in MLNs is a challenging task, we encoded these features implicitly by adding weighted unit clauses to the MLN distribution. Results on an English corpus (KBP 2015) and a Chinese corpus (ACE 2005) show that our MLN based system achieved statistically significantly better performance than a pipeline-based resolver. Future work includes transferring our approach to other NLP tasks and exploring the possibility of incorporating active learning into our approach.

Acknowledgments

We thank the three anonymous reviewers for their detailed comments. This work was supported in part by NSF Grants IIS-1219142 and IIS-1528037, and by the DARPA PPAML Program under AFRL prime contract number FA8750-14-C-0005. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF, DARPA and AFRL.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2074–2080.

Jun Araki, Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014. Detecting subevent structure for event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4553–4558.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, pages 563–566.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422.

Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Unsupervised event coreference resolution. *Computational Linguistics*, 40(2):311–347.

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57.

Chen Chen and Vincent Ng. 2014. SinoCoreferencer: An end-to-end Chinese event coreference resolver. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4532–4538.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2913–2920.

Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. 2011. A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of the 4th International Joint Conference on Natural Language Processing*, pages 102–110.

Agata Cybulska and Piek Vossen. 2012. Using semantic relations to solve event coreference in text. In *Proceedings of the LREC Workshop on Semantic Relations-II Enhancing Resources and Applications (SemRel 2012)*, pages 60–67.

Pedro Domingos and Daniel Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool, San Rafael, CA.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Vibhav Gogate and Pedro Domingos. 2011. Probabilistic theorem proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 256–265.

Gurobi. 2013. *Gurobi Optimizer Reference Manual*. Gurobi Inc.

Yu Hong, Di Lu, Dian Yu, Xiaoman Pan, Xiaobin Wang, Yadong Chen, Lifu Huang, and Heng Ji. 2015. RPI BLENDER TAC-KBP2015 system description. In *Proceedings of the Eighth Text Analysis Conference (TAC 2015)*.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of the Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 75–81.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Scholkopf and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 44–56. MIT Press.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500.

Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in chinese event extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1006–1016.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised within-document event coreference using information propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4539–4544.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Katie McConky, Rakesh Nagi, Moises Sudit, and William Hughes. 2012. Improving event co-reference by context extraction and dynamic feature weighting. In *Proceedings of the 2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 38–43.

Sean Monahan, Michael Mohler, Marc Tomlinson, Amy Book, Maxim Gorelkin, Kevin Crosby, and Mary Brunson. 2015. Populating a knowledge base with information about events. In *Proceedings of the Eighth Text Analysis Conference (TAC 2015)*.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35.

Marta Recasens and Eduard Hovy. 2011. BLANC: Implementing the Rand Index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Sangeetha S. and Michael Arock. 2012. Event coreference resolution using mincut based graph clustering. *International Journal of Computing and Information Sciences*, pages 253–260.

Mark Sammons, Haoruo Peng, Yangqiu Song, Shyam Upadhyay, Chen-Tse Tsai, Pavankumar Reddyand, Subhro Roy, and Dan Roth. 2015. Illinois CCG TAC 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *Proceedings of the Eighth Text Analysis Conference (TAC 2015)*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45–52.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering*, 15(1):143–172.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent Bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.