Joint Learning for Coreference Resolution with Markov Logic

Yang Song[1], **Jing Jiang**[2], **Wayne Xin Zhao**[3], **Sujian Li**[1], **Houfeng Wang**[1]
[1]Key Laboratory of Computational Linguistics (Peking University) Ministry of Education,China
[2]School of Information Systems, Singapore Management University, Singapore
[3]School of Electronics Engineering and Computer Science, Peking University, China
{ysong, lisujian, wanghf}@pku.edu.cn, jingjiang@smu.edu.sg, batmanfly@gmail.com

Abstract

Pairwise coreference resolution models must merge pairwise coreference decisions to generate final outputs. Traditional merging methods adopt different strategies such as the best-first method and enforcing the transitivity constraint, but most of these methods are used independently of the pairwise learning methods as an isolated inference procedure at the end. We propose a joint learning model which combines pairwise classification and mention clustering with Markov logic. Experimental results show that our joint learning system outperforms independent learning systems. Our system gives a better performance than all the learning-based systems from the CoNLL-2011 shared task on the same dataset. Compared with the best system from CoNLL-2011, which employs a rule-based method, our system shows competitive performance.

1 Introduction

The task of noun phrase coreference resolution is to determine which mentions in a text refer to the same real-world entity. Many methods have been proposed for this problem. Among them the mention-pair model (McCarthy and Lehnert, 1995) is one of the most influential ones and can achieve the state-of-the-art performance (Bengtson and Roth, 2008). The mention-pair model splits the task into three parts: mention detection, pairwise classification and mention clustering. Mention detection aims to identify anaphoric noun phrases, including proper nouns, common noun phrases and pronouns. Pairwise classification takes a pair of detected anaphoric noun phrase candidates and determines whether they refer to the same entity. Because these classification decisions are local, they do not guarantee that candidate mentions are partitioned into clusters. Therefore a mention clustering step is needed to resolve conflicts and generate the final mention clusters.

Much work has been done following the mention-pair model (Soon et al., 2001; Ng and Cardie, 2002). In most work, pairwise classification and mention clustering are done sequentially. A major weakness of this approach is that pairwise classification considers only local information, which may not be sufficient to make correct decisions. One way to address this weakness is to jointly learn the pairwise classification model and the mention clustering model. This idea has been explored to some extent by McCallum and Wellner (2005) using conditional undirected graphical models and by Finley and Joachims (2005) using an SVM-based supervised clustering method.

In this paper, we study how to use a different learning framework, Markov logic (Richardson and Domingos, 2006), to learn a joint model for both pairwise classification and mention clustering under the mention-pair model. We choose Markov logic because of its appealing properties. Markov logic is based on first-order logic, which makes the learned models readily interpretable by humans. Moreover, joint learning is natural under the Markov logic framework, with local pairwise classification and global mention clustering both formulated as weighted first-order clauses. In fact, Markov logic has been previously used by Poon and Domingos (2008) for coreference resolution and achieved good

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1245–1254, Jeju Island, Korea, 12–14 July 2012.

results, but it was used for unsupervised coreference resolution and the method was based on a different model, the entity-mention model.

More specifically, to combine mention clustering with pairwise classification, we adopt the commonly used strategies (such as best-first clustering and transitivity constraint), and formulate them as first-order logic formulas under the Markov logic framework. Best-first clustering has been previously studied by Ng and Cardie (2002) and Bengtson and Roth (2008) and found to be effective. Transitivity constraint has been applied to coreference resolution by Klenner (2007) and Finkel and Manning (2008), and also achieved good performance.

We evaluate Markov logic-based method on the dataset from CoNLL-2011 shared task. Our experiment results demonstrate the advantage of joint learning of pairwise classification and mention clustering over independent learning. We examine best-first clustering and transitivity constraint in our methods, and find that both are very useful for coreference resolution. Compared with the state of the art, our method outperforms a baseline that represents a typical system using the mention-pair model. Our method is also better than all learning systems from the CoNLL-2011 shared task based on the reported performance. Even with the top system from CoNLL-2011, our performance is still competitive.

In the rest of this paper, we first describe a standard pairwise coreference resolution system in Section 2. We then present our Markov logic model for pairwise coreference resolution in Section 3. Experimental results are given in Section 4. Finally we discuss related work in Section 5 and conclude in Section 6.

2 Standard Pairwise Coreference Resolution

In this section, we describe standard learning-based framework for pairwise coreference resolution. The major steps include mention detection, pairwise classification and mention clustering.

2.1 Mention Detection

For mention detection, traditional methods include learning-based and rule-based methods. Which kind of method to choose depends on specific dataset. In this paper, we first consider all the noun phrases in the given text as candidate mentions. Without gold standard mention boundaries, we use a well-known preprocessing tool from Stanford's NLP group[1] to extract noun phrases. After obtaining all the extracted noun phrases, we also use a rule-based method to remove some erroneous candidates based on previous studies (e.g. Lee et al. (2011), Uryupina et al. (2011)). Some examples of these erroneous candidates include stop words (e.g. *uh*, *hmm*), web addresses (e.g. `http://www.google.com`), numbers (e.g. $9,000) and pleonastic "it" pronouns.

2.2 Pairwise Classification

For pairwise classification, traditional learning-based methods usually adopt a classification model such as maximum entropy models and support vector machines. Training instances (i.e. positive and negative mention pairs) are constructed from known coreference chains, and features are defined to represent these instances.

In this paper, we build a baseline system that uses maximum entropy models as the classification algorithm. For generation of training instances, we follow the method of Bengtson and Roth (2008). For each predicted mention m, we generate a positive mention pair between m and its closest preceding antecedent, and negative mention pairs by pairing m with each of its preceding predicted mentions which are not coreferential with m. To avoid having too many negative instances, we impose a maximum sentence distance between the two mentions when constructing mention pairs. This is based on the intuition that for each anaphoric mention, its preceding antecedent should appear quite near it, and most coreferential mention pairs which have a long sentence distance can be resolved using string matching. During the testing phase, we generate mention pairs for each mention candidate with each of its preceding mention candidates and use the learned model to make coreference decisions for these mention pairs. We also impose the sentence distance constraint and use string matching for mention pairs with a sentence distance exceeding the threshold.

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

2.3 Mention Clustering

After obtaining the coreferential results for all mention pairs, some clustering method should be used to generate the final output. One strategy is the single-link method, which links all the mention pairs that have a prediction probability higher than a threshold value. Two other alternative methods are the best-first clustering method and clustering with the transitivity constraint. Best-first clustering means that for each candidate mention m, we select the best one from all its preceding candidate mentions based on the prediction probabilities. A threshold value is given to filter out those mention pairs that have a low probability to be coreferential. Transitivity constraint means that if a and b are coreferential and b and c are coreferential, then a and c must also be coreferential. Previous work has found that best-first clustering and transitivity constraint-based clustering are better than the single-link method. Finally we remove all the singleton mentions.

3 Markov Logic for Pairwise Coreference Resolution

In this section, we present our method for joint learning of pairwise classification and mention clustering using Markov logic. For mention detection, training instance generation and postprocessing, our method follows the same procedures as described in Section 2. In what follows, we will first describe the basic Markov logic networks (MLN) framework, and then introduce the first-order logic formulas we use in our MLN including local formulas and global formulas which perform pairwise classification and mention clustering respectively. Through this way, these two isolated parts are combined together, and joint learning and inference can be performed in a single framework. Finally we present inference and parameter learning methods.

3.1 Markov Logic Networks

Markov logic networks combine Markov networks with first-order logic (Richardson and Domingos, 2006; Riedel, 2008). A Markov logic network consists of a set of first-order clauses (which we will refer to as *formulas* in the rest of the paper) just like in first-order logic. However, different from first-order logic where a formula represents a hard constraint, in an MLN, these constraints are softened and they can be violated with some penalty. An MLN $\mathcal{M}$ is therefore a set of *weighted* formulas $\{(\phi_i, w_i)\}_i$, where ϕ_i is a first order formula and w_i is the penalty (the formula's weight). These weighted formulas define a probability distribution over sets of ground atoms or so-called possible worlds. Let y denote a possible world, then we define $p(\mathbf{y})$ as follows:

$$p(\mathbf{y}) = \frac{1}{Z} exp\left(\sum_{(\phi_i, w_i) \in M} w_i \sum_{\mathbf{c} \in C^{n_{\phi_i}}} f_{\mathbf{c}}^{\phi_i}(\mathbf{y}) \right). \quad (1)$$

Here each $\mathbf{c}$ is a binding of free variables in ϕ_i to constants. Each $f_{\mathbf{c}}^{\phi_i}$ represents a binary feature function that returns 1 if the ground formula we get by replacing the free variables in ϕ_i with the constants in $\mathbf{c}$ under the given possible world $\mathbf{y}$ is true, and 0 otherwise. n_{ϕ_i} denotes the number of free variables of a formula ϕ_i. $C^{n_{\phi_i}}$ is the set of all bindings for the free variables in ϕ_i. Z is a normalization constant. This distribution corresponds to a Markov network where nodes represent ground atoms and factors represent ground formulas.

Each formula consists of a set of first-order predicates, logical connectors and variables. Take the following formula as one example:

$$(\phi_i, w_i) : headMatch(a, b) \wedge (a \neq b) \Rightarrow coref(a, b).$$

The formula above indicates that if two different candidate mentions a and b have the same head word, then they are coreferential. Here a and b are variables which can represent any candidate mention, *headMatch* and *coref* are observed predicate and hidden predicate respectively. An observed predicate is one whose value is known from the observations when its free variables are assigned some constants. A hidden predicate is one whose value is not known from the observations. From this example, we can see that *headMatch* is an observed predicate because we can check whether two candidate mentions have the same head word. *coref* is a hidden predicate because this is something we would like to predict.

3.2 Formulas

We use two kinds of formulas for pairwise classification and mention clustering, respectively. For

describing the attributes of $\mathbf{m_i}$	
mentionType(i,t)	m_i has mention type NAM(named entities), NOM(nominal) or PRO(pronouns).
entityType(i,e)	m_i has entity type PERSON, ORG, GPE or UN...
genderType(i,g)	m_i has gender type MALE, FEMALE, NEUTRAL or UN.
numberType(i,n)	m_i has number type SINGULAR, PLURAL or UN.
hasHead(i,h)	m_i has head word h, here h can represent all possible head words.
firstMention(i)	m_i is the first mention in its sentence.
reflexive(i)	m_i is reflexive.
possessive(i)	m_i is possessive.
definite(i)	m_i is definite noun phrase.
indefinite(i)	m_i is indefinite noun phrase.
demonstrative(i)	m_i is demonstrative.
describing the attributes of relations between $\mathbf{m_j}$ and $\mathbf{m_i}$	
mentionDistance(j,i,m)	Distance between m_j and m_i in mentions.
sentenceDistance(j,i,s)	Distance between m_j and m_i in sentences.
bothMatch(j,i,b)	Gender and number of both m_j and m_i match: AGREE_YES, AGREE_NO and AGREE_UN).
closestMatch(j,i,c)	m_j is the first agreement in number and gender when looking backward from m_i: CAGREE_YES, CAGREE_NO and CAGREE_UN.
exactStrMatch(j,i)	Exact strings match between m_j and m_i.
pronounStrMatch(j,i)	Both are pronouns and their strings match.
nopronounStrMatch(j,i)	Both are not pronouns and their strings match.
properStrMatch(j,i)	Both are proper names and their strings match.
headMatch(j,i)	Head word strings match between m_j and m_i.
subStrMatch(j,i)	Sub-word strings match between m_j and m_i.
animacyMatch(j,i)	Animacy types match between m_j and m_i.
nested(j,i)	$m_{j/i}$ is included in $m_{i/j}$.
c_command(j,i)	$m_{j/i}$ C-Commands $m_{i/j}$.
sameSpeaker(j,i)	m_j and m_i have the same speaker.
entityTypeMatch(j,i)	Entity types match between m_j and m_i.
alias(j,i)	$m_{j/i}$ is an alias of $m_{i/j}$.
srlMatch(j,i)	m_j and m_i have the same semantic role.
verbMatch(j,i)	m_j and m_i have semantic role for the same verb.

Table 1: Observed predicates.

pairwise classification, because the decisions are local, we use a set of *local* formulas. For mention clustering, we use *global* formulas to implement best-first clustering or transitivity constraint. We naturally combine pairwise classification with mention clustering via local and global formulas in the Markov logic framework, which is the essence of "joint learning" in our work.

3.2.1 Local Formulas

A local formula relates any observed predicates to exactly one hidden predicate. For our problem, we define a list of observed predicates to describe the properties of individual candidate mentions and the relations between two candidate mentions, shown in Table 1. For our problem, we have only one hidden predicate, i.e. *coref*. Most of our local formulas are from existing work (e.g. Soon et al. (2001), Ng and Cardie (2002), Sapena et al. (2011)). They are listed in Table 2, where the symbol "+" indicates that for every value of the variable preceding "+" there is a separate weight for the corresponding formula.

3.2.2 Global Formulas

Global formulas are designed to add global constraints for hidden predicates. Since in our problem there is only one hidden predicate, i.e. *coref*, our global formulas incorporate correlations among different ground atoms of the *coref* predicates. Next we will show the best-first and transitivity global constraints. Note that we treat them as hard constraints so we do not set any weights for these global formulas.

Lexical Features
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ exactStrMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ pronounStrMatch (j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ properStrMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ nopronounStrMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ headMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ subStrMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
hasHead(j,h_1+) $\wedge$ hasHead(i,h_2+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
Grammatical Features
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ genderType(j,g_1+) $\wedge$ genderType(i,g_2+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ numberType(j,n_1+) $\wedge$ numberType(i,n_2+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ bothMatch(j,i,b+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ closestMatch(j,i,c+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ animacyMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ nested(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ c_command(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(mentionType(j,t_1+) $\vee$ mentionType(i,t_2+)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(reflexive(j) $\vee$ reflexive(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(possessive(j) $\vee$ possessive(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(definite(j) $\vee$ definite(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(indefinite(j) $\vee$ indefinite(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(demonstrative(j) $\vee$ demonstrative(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
Distance and position Features
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ sentenceDistance(j,i,s+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ mentionDistance (j,i,m+) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(firstMention(j) $\vee$ firstMention(i)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
Semantic Features
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ alias(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ sameSpeaker(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ entityTypeMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ srlMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
mentionType(j,t_1+) $\wedge$ mentionType(i,t_2+) $\wedge$ verbMatch(j,i) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)
(entityType(j,e_1+) $\vee$ entityType(i,e_2+)) $\wedge$ j$\neq$ i $\Rightarrow$ coref(j,i)

Table 2: Local Formulas.

Best-First constraint:

$$coref(j,i) \Rightarrow \neg coref(k,i) \quad \forall j,k < i (k \neq j) \quad (2)$$

Here we assume that *coref(j,i)* returns true if candidate mentions j and i are coreferential and false otherwise. Therefore for each candidate mention i, we should only select at most one candidate mention j to return true for the predicate *coref(j,i)* from all its preceding candidate mentions.

Transitivity constraint:

$$coref(j,k) \wedge coref(k,i) \wedge j < k < i \Rightarrow coref(j,i) \quad (3)$$

$$coref(j,k) \wedge coref(j,i) \wedge j < k < i \Rightarrow coref(k,i) \quad (4)$$

$$coref(j,i) \wedge coref(k,i) \wedge j < k < i \Rightarrow coref(j,k) \quad (5)$$

With the transitivity constraint, it means for given mentions j, k and i, if any two pairs of them are coreferential, then the third pair of them should be also coreferential.

We use best-first clustering and transitivity constraint in our joint learning model respectively. Detailed comparisons between them will be shown in Section 4.

3.3 Inference

We use MAP inference which is implemented by Integer Linear Programming (ILP). Its objective is to maximize a posteriori probability as follows. Here we use x to represent all the observed ground atoms and y to represent the hidden ground atoms. Formally, we have

$$\hat{y} = \arg\max_y \ p(y|x) \simeq \arg\max_y s(y,x),$$

where

$$s(y,x) = \sum_{(\phi_i, w_i) \in M} w_i \sum_{\mathbf{c} \in C^{n_{\phi_i}}} f_{\mathbf{c}}^{\phi_i}(y,x). \quad (6)$$

Each hidden ground atom can only takes a value of either 0 or 1. And global formulas should be satisfied as hard constraints when inferring the best $\hat{y}$. So

the problem can be easily solved using ILP. Detailed introduction about transforming ground Markov networks in Markov logic into an ILP problem can be found in (Riedel, 2008).

3.4 Parameter Learning

For parameter learning, we employ the online learner MIRA (Crammer and Singer, 2003), which establishes a large margin between the score of the gold solution and all wrong solutions to learn the weights. This is achieved by solving the quadratic program as follows

$$\min \parallel \mathbf{w}_t - \mathbf{w}_{t-1} \parallel . \quad (7)$$

$$s.t. \quad s(y_i, x_i) - s(y', x_i) \geq L(y_i, y')$$
$$\forall y' \neq y_i, \quad (y_i, x_i) \in D$$

Here $D = \{(y_i, x_i)\}_{i=1}^{N}$ represents N training instances (each instance represents one single document in the dataset) and t represents the number of iterations. In our problem, we adopt 1-best MIRA, which means that in each iteration we try to find $\mathbf{w}_t$ which can guarantee the difference between the right solution y_i and the best solution y' (i.e. the one with the highest score s(y', x_i), equivalent to $\hat{y}$ in Section 3.3)) is at least as big as the loss $L(y_i, y')$, while changing $\mathbf{w}_{t-1}$ as little as possible. The number of false ground atoms of *coref* predicate is selected as loss function in our experiments. Hard global constraints (i.e. best-first clustering or transitivity constraint) must be satisfied when inferring the best y' in each iteration, which can make learned weights more effective.

4 Experiments

In this section, we will first describe the dataset and evaluation metrics we use. We will then present the effect of our joint learning method, and finally discuss the comparison with the state of the art.

4.1 Data Set

We use the dataset from the CoNLL-2011 shared task, "Modeling Unrestricted Coreference in OntoNotes" (Pradhan et al., 2011)[2]. It uses the English portion of the OntoNotes v4.0 corpus. There are three important differences between OntoNotes and another well-known coreference dataset from ACE. First, OntoNotes does not label any singleton entity cluster, which has only one reference in the text. Second, only identity coreference is tagged in OntoNotes, but not appositives or predicate nominatives. Third, ACE only considers mentions which belong to ACE entity types, whereas OntoNotes considers more entity types. The shared task is to automatically identify both entity coreference and event coreference, although we only focus on entity coreference in this paper. We don't assume that gold standard mention boundaries are given. So we develop a heuristic method for mention detection. See details in Section 2.1.

The training set consists of 1674 documents from newswire, magazine articles, broadcast news, broadcast conversations and webpages, and the development set consists of 202 documents from the same source. For training set, there are 101264 mentions from 26612 entities. And for development set, there are 14291 mentions from 3752 entities (Pradhan et al., 2011).

4.2 Evaluation Metrics

We use the same evaluation metrics as used in CoNLL-2011. Specifically, for mention detection, we use precision, recall and the F-measure. A mention is considered to be correct only if it matches the exact same span of characters in the annotation key. For coreference resolution, MUC (Vilain et al., 1995), B-CUBED (Bagga and Baldwin, 1998) and CEAF-E (Luo, 2005) are used for evaluation. The unweighted average F score of them is used to compare different systems.

4.3 The Effect of Joint Learning

To assess the performance of our method, we set up several variations of our system to compare with the joint learning system. The *MLN-Local* system uses only the local formulas described in Table 2 without any global constraints under the MLN framework. By default, the *MLN-Local* system uses the single-link method to generate clustering results. The *MLN-Local+BF* system replaces the single-link method with best-first clustering to infer mention clustering results after learning the weights for all the local formulas. The *MLN-Local+Trans* system replaces the best-first clustering with transitivity

[2]http://conll.cemantix.org/2011/

System	Mention Detection			MUC			B-cube			CEAF			Avg
	R	P	F	R	P	F	R	P	F	R	P	F	F
MLN-Local	62.52	74.75	68.09	56.07	65.55	60.44	65.67	72.95	69.12	45.55	37.19	40.95	56.84
MLN-Local+BF	65.74	73.2	69.27	56.79	64.08	60.22	65.71	74.18	69.69	47.29	40.53	43.65	57.85
MLN-Local+Trans	**68.49**	70.32	69.40	**57.16**	60.98	59.01	**66.97**	72.90	69.81	46.96	**43.34**	**45.08**	57.97
MLN-Joint(BF)	64.36	75.25	69.38	55.47	66.95	60.67	64.14	77.75	70.29	50.47	39.85	44.53	58.50
MLN-Joint(Trans)	64.46	**75.37**	**69.49**	55.48	**67.15**	**60.76**	64.00	**78.11**	**70.36**	**50.63**	39.84	44.60	**58.57**

Table 3: Comparison between different MLN-based systems, using 10-fold cross validation on the training dataset.

constraint. The *MLN-Joint* system is a joint model for both pairwise classification and mention clustering. It can combine either best-first clustering or enforcing transitivity constraint with pairwise classification, and we denote these two variants of *MLN-Joint* as *MLN-Joint(BF)* and *MLN-Joint(Trans)* respectively.

To compare the performance of the various systems above, we use 10-fold cross validation on the training dataset. We empirically find that our method has a fast convergence rate, to learn the MLN model, we set the number of iterations to be 10.

The performance of these compared systems is shown in Table 3. To provide some context for the performance of this task, we report the median average F-score of the official results of CoNLL-2011, which is 50.12 (Pradhan et al., 2011). We can see that *MLN-Local* achieves an average F-score of 56.84, which is well above the median score. When adding best-first or transitivity constraint which is independent of pairwise classification, *MLN-Local+BF* and *MLN-Local+Trans* achieve better results of 57.85 and 57.97. Most of all, we can see that the joint learning model (*MLN-Joint(BF)* or *MLN-Joint(Trans)*) significantly outperforms independent learning model (*MLN-Local+BF* or *MLN-Local+Trans*) no matter whether best-first clustering or transitivity constraint is used (based on a paired 2-tailed t-test with $p < 0.05$) with the score of 58.50 or 58.57, which shows the effectiveness of our proposed joint learning method.

Best-first clustering and transitivity constraint are very useful in Markov logic framework, and both *MLN-Local* and *MLN-Joint* benefit from them. For *MLN-Joint*, these two clustering methods result in similar performance. But actually, transitivity is harder than best-first, because it significantly increases the number of formulas for constraints and slows down the learning process. In our experiments, we find that *MLN-Joint(Trans)*[3] is much slower than *MLN-Joint(BF)*. Overall, *MLN-Joint(BF)* has a good trade-off between effectiveness and efficiency.

4.4 Comparison with the State of the Art

In order to compare our method with the state-of-the-art systems, we consider the following systems. We implemented a traditional pairwise coreference system using Maximum Entropy as the base classifier and best-first clustering to link the results. We used the same set of local features in *MLN-Joint*. We refer to this system as *MaxEnt+BF*. To replace best-first clustering with transitivity constraint, we have another system named as *MaxEnt+Trans*. We also consider the best 3 systems from CoNLL-2011 shared task. Chang's system uses ILP to perform best-first clustering after training a pairwise coreference model. Sapena's system uses a relaxation labeling method to iteratively perform function optimization for labeling each mention's entity after learning the weights for features under a C4.5 learner. Lee's system is a purely rule-based one. They use a battery of sieves by precision (from highest to lowest) to iteratively choose antecedent for each mention. They obtained the highest score in CoNLL-2011.

Table 4 shows the comparisons of our system with the state-of-the-art systems on the development set of CoNLL-2011. From the results, we can see that our joint learning systems are obviously better than

[3]For MLN-Joint(Trans), not all training instances can be learnt in a reasonable amount of time, so we set up a time out threshold of 100 seconds. If the model cannot response in 100 seconds for some training instance, we remove it from the training set.

System	Mention Detection			MUC			B-cube			CEAF			Avg
	R	P	F	R	P	F	R	P	F	R	P	F	F
MLN-Joint(BF)	67.33	72.94	**70.02**	**58.03**	64.05	60.89	67.11	73.88	70.33	47.6	41.92	44.58	58.60
MLN-Joint(Trans)	67.28	72.88	69.97	58.00	64.10	**60.90**	67.12	74.13	**70.45**	47.70	41.96	44.65	**58.67**
MaxEnt+BF	60.54	**76.64**	67.64	52.20	**68.52**	59.26	60.85	80.15	69.18	51.6	37.05	43.13	57.19
MaxEnt+Trans	61.36	76.11	67.94	51.46	68.40	58.73	59.79	**81.69**	69.04	**53.03**	37.84	44.17	57.31
Lee's System	-	-	-	57.50	59.10	58.30	**71.00**	69.20	70.10	48.10	**46.50**	**47.30**	58.60
Sapena's System	**92.45**	27.34	42.20	54.53	62.25	58.13	63.72	73.83	68.40	47.20	40.01	43.31	56.61
Chang's System	-	-	64.69	-	-	55.8	-	-	69.29	-	-	43.96	56.35

Table 4: Comparisons with state-of-the-art systems on the development dataset.

MaxEnt+BF and *MaxEnt+Trans*. They also outperform the learning-based systems of Sapena et al. (2011) and Chang et al. (2011), and perform competitively with Lee's system (Lee et al., 2011). Note that Lee's system is purely rule-based, while our methods are developed in a theoretically sound way, i.e., Markov logic framework.

5 Related Work

Supervised noun phrase coreference resolution has been extensively studied. Besides the mention-pair model, two other commonly used models are the entity-mention model (Luo et al., 2004; Yang et al., 2008) and ranking models (Denis and Baldridge, 2008; Rahman and Ng, 2009). Interested readers can refer to the literature review by Ng (2010).

Under the mention-pair model, Klenner (2007) and Finkel and Manning (2008) applied Integer Linear Programming (ILP) to enforce transitivity on the pairwise classification results. Chang et al. (2011) used the same ILP technique to incorporate best-first clustering and generate the mention clusters. In all these studies, however, mention clustering is combined with pairwise classification only at the inference stage but not at the learning stage.

To perform joint learning of pairwise classification and mention clustering, in (McCallum and Wellner, 2005), each mention pair corresponds to a binary variable indicating whether the two mentions are coreferential, and the dependence between these variables is modeled by conditional undirected graphical models. Finley and Joachims (2005) proposed a general SVM-based framework for supervised clustering that learns item-pair similarity measures, and applied the framework to noun phrase coreference resolution. In our work, we take a different approach and apply Markov logic. As we have shown in Section 3, given the flexibility of Markov logic, it is straightforward to perform joint learning of pairwise classification and mention clustering.

In recent years, Markov logic has been widely used in natural language processing problems (Poon and Domingos, 2009; Yoshikawa et al., 2009; Che and Liu, 2010). For coreference resolution, the most notable one is unsupervised coreference resolution by Poon and Domingos (2008). Poon and Domingos (2008) followed the entity-mention model while we follow the mention-pair model, which are quite different approaches. To seek good performance in an unsupervised way, Poon and Domingos (2008) highly rely on two important strong indicators: appositives and predicate nominatives. However, OntoNotes corpus (state-of-art NLP data collection) on coreference layer for CoNLL-2011 has excluded these two conditions of annotations (appositives and predicate nominatives) from their judging guidelines. Compared with it, our methods are more applicable for real dataset. Huang et al. (2009) used Markov logic to predict coreference probabilities for mention pairs followed by correlation clustering to generate the final results. Although they also perform joint learning, at the inference stage, they still make pairwise coreference decisions and cluster mentions sequentially. Unlike their method, We formulate the two steps into a single framework.

Besides combining pairwise classification and mention clustering, there has also been some work that jointly performs mention detection and coreference resolution. Daumé and Marcu (2005) developed such a model based on the Learning as

Search Optimization (LaSO) framework. Rahman and Ng (2009) proposed to learn a cluster-ranker for discourse-new mention detection jointly with coreference resolution. Denis and Baldridge (2007) adopted an Integer Linear Programming (ILP) formulation for coreference resolution which models anaphoricity and coreference as a joint task.

6 Conclusion

In this paper we present a joint learning method with Markov logic which naturally combines pairwise classification and mention clustering. Experimental results show that the joint learning method significantly outperforms baseline methods. Our method is also better than all the learning-based systems in CoNLL-2011 and reaches the same level of performance with the best system.

In the future we will try to design more global constraints and explore deeper relations between training instances generation and mention clustering. We will also attempt to introduce more predicates and transform structure learning techniques for MLN into coreference problems.

Acknowledgments

Part of the work was done when the first author was a visiting student in the Singapore Management University. And this work was partially supported by the National High Technology Research and Development Program of China(863 Program) (No.2012AA011101), the National Natural Science Foundation of China (No.91024009, No.60973053, No.90920011), and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20090001110047).

References

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

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*.

K. Chang, R. Samdani, A. Rozovskaya, N. Rizzolo, M. Sammons, and D. Roth. 2011. Inference protocols for coreference resolution. In *CoNLL Shared Task*, pages 40–44, Portland, Oregon, USA. Association for Computational Linguistics.

Wanxiang Che and Ting Liu. 2010. Jointly modeling wsd and srl with markov logic. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING*, pages 161–169. Tsinghua University Press.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

III Hal Daumé and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 97–104, Morristown, NJ, USA. Association for Computational Linguistics.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, New York, April. Association for Computational Linguistics.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *EMNLP*, pages 660–669.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *ACL (Short Papers)*, pages 45–48. The Association for Computer Linguistics.

T. Finley and T. Joachims. 2005. Supervised clustering with support vector machines. In *International Conference on Machine Learning (ICML)*, pages 217–224.

Shujian Huang, Yabing Zhang, Junsheng Zhou, and Jiajun Chen. 2009. Coreference resolution using markov logic networks. In *Proceedings of Computational Linguistics and Intelligent Text Processing: 10th International Conference, CICLing 2009*.

M. Klenner. 2007. Enforcing consistency on coreference sets. In *RANLP*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA, June. Association for Computational Linguistics.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, A Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*, pages 135–142.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proc. of HLT/EMNLP*, pages 25–32.

Andrew McCallum and Ben Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. In *Advances in Neural Information Processing Systems*, pages 905–912. MIT Press.

J. McCarthy and W. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the ACL*, pages 104–111.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *ACL*, pages 1396–1411. The Association for Computer Linguistics.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *EMNLP*, pages 650–659.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *EMNLP*, pages 1–10.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA, June. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of EMNLP*, pages 968–977.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of map inference for markov logic. In *UAI*, pages 468–475. AUAI Press.

Emili Sapena, Lluís Padró, and Jordi Turmo. 2011. Relaxcor participation in conll shared task on coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 35–39, Portland, Oregon, USA, June. Association for Computational Linguistics.

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

Olga Uryupina, Sriparna Saha, Asif Ekbal, and Massimo Poesio. 2011. Multi-metric optimization for coreference: The unitn / iitp / essex submission to the 2011 conll shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 61–65, Portland, Oregon, USA, June. Association for Computational Linguistics.

Marc B. Vilain, John D. Burger, John S. Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC*, pages 45–52.

Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *ACL*, pages 843–851. The Association for Computer Linguistics.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with markov logic. In *ACL/AFNLP*, pages 405–413. The Association for Computer Linguistics.