Unsupervised Coreference Resolution Using a Multi-pass Graph Labeling Approach

Nafise Sadat Moosavi, Gholamreza Ghassem-Sani

Sharif University of Technology
n_moosavi@ce.sharif.edu, sani@sharif.edu

Abstract

This paper proposes a new multi-pass graph labeling approach for unsupervised coreference resolution solved by relaxation labeling. A heart of our approach is an incremental graph development method which hierarchically deploys coreference features from higher precision to lower ones.

Keywords: unsupervised coreference resolution, graph labeling, relaxation labeling, hierarchical graph construction method.

1. Introduction

Coreferences are relations that hold between expressions which refer to the same entities. Expressions are often called mentions of the entities. A coreference is a reflexive, symmetric, and transitive equivalence relation. The reflexive and transitive closure over coreference relations generates equivalence classes of mentions, which are called a coreference chain.

Coreference resolution is equivalent to the set partitioning problem in which the search space is the set of all mutually disjoint subsets of a document mentions. From an entirely language engineering perspective, the accurate identification of the entities that are referred to is an important challenge (Denis, 2007). Indeed, numerous natural language processing (NLP) tasks such as information extraction, question answering, automatic summarization, machine translation, and natural language generation could benefit from the availability of a coreference resolution system.

Using coreference relationships, we can construct a graph \( G \), in which each mention is a vertex, and each coreference relation forms an edge between corresponding vertices. In this way, coreference resolution problem can be formulated as a graph labeling problem on such a graph.

Several earlier works modeled coreference resolution as a graph labeling or graph partitioning problem (e.g., McCallum and Wellner, 2003; Lang et al., 2009; Sapena et al., 2010). This paper presents a new unsupervised coreference resolution system, which casts coreference resolution as a graph labeling problem and employs relaxation labeling technique for labeling assignment. The presented system has been inspired by the success of two recent coreference systems (Sapena et al., 2010; Raghunathan et al. 2010). It deploys a new hierarchical graph construction method for developing the adjacency graph. By using this hierarchical graph construction, instead of the whole set of neighbors, the labeling algorithm considers only the most reliable set of neighbors at each pass of the algorithm.

2. Related Works

Recently, the accessibility to annotated coreference data of MUC conferences and ACE evaluations has brought up deployment of a wide variety of supervised machine learning approaches to the coreference resolution problem (Ng, 2010). The focus of statistical approaches to the coreference resolution has been moved from attainment of simple pairwise models, which determine whether two mentions are referring to the same entity (e.g., Soon et al., 2001; Yang et al., 2003) to the use of rich linguistic features (e.g., Ji et al., 2005; Ponzo et al. and Strube, 2006), and utilization of advanced learning techniques (Denis and Baldrige, 2007). More recent works on coreference resolution have shown that a rich feature space which can model lexical, syntactic, semantic, and discourse aspects of mentions is essential for the success of the coreference task (Bengtson and Roth, 2008; Haghhighi and Klein, 2010; Haghhighi and Klein, 2009). When these rich features are combined with the complexity of coreference models, supervised approaches will be more dependent on the annotated data and less appropriate for languages with insignificant or no annotated data. Because of the increasing importance of multilingual processing in the NLP community, developing unsupervised and semi-supervised methods for automatic processing of languages with limited resource has become more essential.

Unsupervised learning methods totally eliminate the need to annotated data. Remarkably, unsupervised coreference methods compete with their supervised counterparts (Haghhighi and Klein, 2010; Haghhighi and Klein, 2009; Ng, 2010; Raghunathan et al., 2010; Lee et al., 2011). Motivated in part by such observations, in this paper we present a new unsupervised model for probabilistically finding coreference partitions of unlabeled documents. While utilization of graph models in coreference resolution has shown promising results (Lang et al., 2009; Sapena et al., 2010), we model unsupervised coreference resolution as a graph labeling problem which is solved by a relaxation labeling algorithm. The proposed method has been inspired by the success of two recent coreference systems (Sapena et al., 2010; Raghunathan et al. 2010), and it benefits from some advantages of both approaches.

3. Task Description

The input of a coreference resolution system includes a document consisting of a set of mentions. Mentions are typically a number of noun phrases that are headed by
some pronounial or nominal terminals. An intra-document coreference resolution system partitions such mentions based on their underlying referent entities. The developed system’s results are reported using gold mention boundaries. However, predicted mention boundaries can be used instead of gold ones, by using a simple mention detection model (Haghighi and Klein, 2010).

3.1. Data Sets
In this work the following data sets have been used for Evaluation.

- ACE2004-ROTH-DEV: A development set of ACE 2004, which was first utilized in (Bengtson and Roth, 2008). It consists of 68 documents and 4536 mentions.
- ACE2004-CULOTTA-TEST: A test split of ACE 2004, which was first utilized in (Culotta et al., 2007). It consists of 107 documents and 5469 mentions.

3.2. Evaluation Metrics
The proposed system evaluated by two commonly used coreference resolution metrics:

- B-Cubed: B-Cubed F-Score (Bagga and Baldwin, 1998) is a measure of overlap between predicted mention clusters and true clusters.
- MUC: MUC measures how many predicted clusters are needed to be merged for covering true clusters.

4. Basic Definitions
Using relations between a document’s mentions, we can construct an undirected graph in which each mention is represented as a vertex, and each graph edge corresponds to a coreference relation between two mentions. In other words, assigning mentions to entity clusters can be formulated as a graph labeling problem (Sapena et al., 2010). As we consider graphs whose vertices represent mentions, vertices and mentions are being used interchangeably.

It is desired to model the mutual influence between neighboring mentions for simultaneously estimating the class labels of all mentions in a document. Theoretically, such a model can cover long-range influences among transitively related mentions. Such influences decrease as the distance of two mentions increases. However, for the tractability purpose, one should focus on the strongest dependencies among neighbors. Such a model, which is called first-order Markov random Fields (Pelckowitz, 1990), cannot be solved in a closed analytic form and therefore has been addressed by an iterative technique called relaxation labeling (Hummel and Zucker, 1983; Pelillo, 1997). Relaxation labeling is an iterative optimization process which efficiently solves the problem of assigning a set of labels to a set of variables satisfying a set of constraints. Relaxation labeling aims at a label assignment that satisfies as many constraints as possible. In other words, it uses contextual information, which is expressed as a number of constraint functions, for reducing local ambiguities in graph labeling.

One of significant features of relaxation labeling is its ability to deal with any kind of constraints. The algorithm is independent of the complexity of defined constraints (i.e., complexity of modeled application), and it can be improved by using any available constraints. Thus, complex constraints can be used without the need to change the algorithm.

This algorithm has been applied to various NLP tasks such as POS tagging (Marquez and Padro, 1997), shallow parsing (Voutilainen and Padro, 1997), word sense disambiguation (Padro, 1998), and supervised coreference resolution (Sapena et al., 2010).

4.1. Relaxation Labeling
Suppose that \( \lambda \) is the set of possible labels for the set of variable \( V = \{ \nu_1, ..., \nu_m \} \) (which in our modeling, corresponds to graph vertices), and \( R = \{ r_\lambda(\lambda) \} \) is a compatibility matrix that defines relations between variables (i.e., adjacency matrix in our problem). Each coefficient \( r_\lambda(\lambda) \) corresponds to a constraint that measures the certainty level of labeling variable \( \nu_i \) with \( \lambda \), where variable \( \nu_j \) has been labeled with \( \lambda \). In other words, the higher value of \( r_\lambda(\lambda) \) the more probable \( \nu_i \) is labeled with \( \lambda \), while \( \nu_j \) has been labeled with \( \lambda \).

Relaxation labeling starts by assigning initial labels to variables. It then iteratively modifies label assignments in a manner that the labeling satisfies as many constraints as possible, where constraints have been defined by the compatibility matrix. Information of the compatibility matrix and current label assignment are used for parallel update of labels. In other words, each variable \( \nu_i \in V \) gets an initial probability vector \( \overrightarrow{p}_i \), which has one element for each possible value (label) for \( \nu_i \). \( p^{(0)}(\lambda) \) indicates one element of \( \overrightarrow{p}^{(0)} \), which corresponds to the probability of assigning label \( \lambda \) to variable \( \nu_i \) at the \( t \)th iteration. The whole set of \( \overrightarrow{p} = \{ \overrightarrow{p}_1, ..., \overrightarrow{p}_m \} \) is denominated as weighted label assignments.

A support function is defined for each possible label \( \lambda \) of each variable \( \nu_i \). The compatibility of the current label assignments of neighbors of \( \nu_i \), and hypothesis "\( \lambda \) is the label of \( \nu_i \)", is measured by this support function. This support function is defined as follows:

\[
s_i^{(0)}(\lambda; \overrightarrow{p}) = \sum_{\text{neighbors}(\nu_i)} r_\lambda(\lambda) p_i^{(0)}(\lambda)
\]

Clearly, the higher value of support function the more confident we are about labeling \( \nu_i \) with \( \lambda \).

The support function is used for updating the label assignments; the high value of \( s_i(\lambda; \overrightarrow{p}) \) increases \( \overrightarrow{p}_i(\lambda) \).

Updating weighted label assignments is done as follow:

\[
p_i^{(t+1)}(\lambda) = \frac{p_i^{(t)}(\lambda) \times (-m + s_i^{(t)}(\lambda, \overrightarrow{p}^{(t)}))}{\sum_{\sigma \in \Lambda} p_i^{(t)}(\sigma) \times (-m + s_i^{(t)}(\sigma, \overrightarrow{p}^{(t)}))},
\]

where \( m = \min(s_i^{(t)}) \).
Term $m$ is added for negative support values, and denominator is for normalizing the result. It is necessary for $p^{\text{new}}(\lambda)$ to remain a probability. The process of calculating $p^{\text{new}}(\lambda)$ continues until the algorithm converges to stable values for $p$. Relaxation labeling complexity is linear in proportion to the number of variables (i.e., number of mentions in a document).

5. System Description

Our unsupervised model casts coreference resolution as graph labeling. This was first inspired by the successful results of (Sapena et al., 2010), which benefits from combining group classification and chain formation methods in a same step. Combination of group classification and chain formation methods in a global method ensures the consistency of solutions (Sapena et al., 2010).

The domain knowledge (i.e., coreference relation in our problem) is combined with the model through coefficients of compatibility matrix. While compatibility matrix causes changes in weighted label assignments, the choice of such coefficients is critical for the success of the algorithm. These coefficients can be set manually based on the problem specification. Alternatively, they can be learnt from a training set. For instance, Sapena et al. (2010) used a decision tree algorithm for learning compatibility coefficients. In order to use the relaxation labeling algorithm for unsupervised coreference resolution, compatibility coefficients should be determined in an unsupervised manner. We first considered the way that (Cardie and Wagstaff, 1999) used for deriving incompatibility functions out of amount of linguistic features to compute the compatibility coefficients. However, using this method brings up the concern of setting different heuristic and experimental parameters for the weights of compatibility functions (Cardie and Wagstaff, 1999).

We tackled this problem by incorporating the successful sieve architecture presented in (Raghunathan et al., 2010). This work was based on the fact that a small number of high precision features are often overwhelmed by the larger number of low precision ones. Thus, Raghunathan et al. (2010) proposed a multi-pass system, in which the higher precision features were deployed at earlier stages of coreference decisions.

We have deployed their multi-pass idea in our coreference resolution System. In other words, we have used a layered system, in which each layer is constructed based on different types of coreference knowledge, and each layer feeds its output forward to the next layer. The layers are then organized in a way that highest precision feature is used at the first layer, and successive layers deploy decreasing precision features.

The layered architecture has been deployed in the graph construction phase; graph is developed incrementally based on the specified features of each layer. In each pass, the relaxation labeling algorithm is applied to the current partially constructed graph, and thus the algorithm just considers more certain neighborhood relations (i.e., the neighborhood relations that contain higher precision features); unattached vertices will be later labeled at subsequent passes.

After determination of weighted label assignments in each partially constructed graph, some of the assignments are determined as being confident enough. Such assignments will not be affected by the weaker features of the later passes.

5.1. Hierarchical Graph Development

At each pass of the algorithm, the system processes all mentions of the document. Supposing the algorithm is in pass $j$, containing feature set $F = \{f_1', f_2', ..., f_d\}$, in which $m$ is the number of features enclosed in pass $j$, and if we are processing mention $m$, the adjacency graph development process would be followed as:

Every mention $m$, located before $m$, is considered as a candidate for graph development. Since cataphora is not handled in our current system, $m$ is skipped if it is a pronoun; otherwise if $m_i$ and $m_j$ have one of the features common, the vertices $v_i$ and $v_j$ corresponding to $m_i$ and $m_j$ would be connected by a new edge (only if they were unattached before). Because of this hierarchical architecture, we consider only two edge weights (except for the last pass) in each pass: +1 and -1. A weight of +1 represents a preference, and a weight of -1 represents a restriction. The partially constructed graph of each pass contains only the vertices that have at least one edge to some other vertices.

We now describe the seven passes of the proposed multi-pass graph labeling algorithm. It is notable that the first 6 passes mostly consider non-pronoun mentions and the last pass is only for pronouns. You can refer to (Bengtson and Roth, 2008; Haghighi and Klein, 2009; Raghunathan et al., 2010) for detailed explanation of the used features in each pass.

Initialization and post-processing of each Pass

We have used the same approach as Sapena et al. (2010) for initializing weighted label assignments. The first non-pronoun mention has no previous mention to be referred to, and it will be considered as the beginning of a new entity; thus, the label assignment of this mention is marked as a first confident assignment in our model.

The final label assignments of each step are considered as the initial label assignments of the next step. Indeed, if a vertex has a positive neighbor (i.e., a neighbor with a positive edge weight) with a confident label assignment, its weighted label assignment will also be marked as a confident assignment at the end of each given pass, and therefore, it would not be changed at later passes.

Pass 1
If entire extent of $m_i$ and $m_j$ match exactly, $v_i$ and $v_j$ are connected by an edge with weight +1 in pass 1.

Pass 2
In this pass, $v_i$ and $v_j$ are connected by an edge with weight -1 where one of the following conditions hold: The gender and/or number of $m_i$ and $m_j$ does not match, or $m_i$ and $m_j$ have different predicted entity type.
Similarly, one of the following conditions is required for \( v_i \) and \( v_j \) vertices to be connected by an edge with weight +1 in pass 2:

1. \( m_i \) and \( m_j \) are in appositive or role appositive relation.
2. One of the \( m_i \) or \( m_j \) mentions is an alias or demonym of the other.
3. \( m_i \) is a relative pronoun of \( m_j \).

**Pass 3**

If all three following conditions hold, the graph is developed by adding a +1 edge between the corresponding vertices:

1. \( m_i \) and \( m_j \) have the same head.
2. All non-stop words of \( m_i \) are included in the set of \( m_j \)'s non-stop words, or vice versa.
3. All modifiers of \( m_i \) are included in the set of \( m_j \)'s modifiers, or vice versa.

**Pass 4**

If \( m_i \) and \( m_j \) have the same head, and one of the second or the third conditions of pass 3 holds, the two vertices are attached by an edge weight +1.

**Pass 5**

If \( m_i \) and \( m_j \) have the same head, their corresponding vertices will be attached in this pass by an edge weight +1.

**Pass 6**

If the head of \( m_i \) matches any word of mention \( m_j \), the graph is developed by attaching \( v_i \) and \( v_j \) by an edge with weight +1. After weighted label assignments have been determined on the result graph of this pass, every assignment corresponding to a non-pronoun mention will be marked as a confident assignment. In this way, the next pass will be dedicated to resolving only pronouns.

**Pass 7**

The following features are used for extending the graph in phase 7. As mentioned before, this pass only considers pronouns. \( m_i \) is attached to its previous mention \( m_i \) based on the following conditions:

1. \( m_i \) and \( m_j \) have the same gender.
2. \( m_i \) and \( m_j \) have the same number.
3. \( m_i \) and \( m_j \) have the same predicted entity type.
4. \( m_i \) and \( m_j \) appearing within two words of a verb representing "to say".
5. \( m_i \) and \( m_j \) have the same animacy.

Besides the used features, another difference between this pass and previous passes is its edge's weights. Since all of these conditions may be present between two mentions and the larger number of satisfied conditions the more preferences to link these mentions, each of these satisfied conditions adds the weight \( \frac{1}{2} \) to the attaching links.

### 6. Experimental Results

The experimental results of our approach are presented in Table 1. In order to measure the impact of hierarchical graph development, we have also presented results of a single pass flat variant of our system. This variant constructs the adjacency graph in a single step and uses all features of the multi-pass system in one step. In this version, edge weights are computed as follow:

\[
\omega_{ij} = \min(1, \sum_{k \in F} \delta_k f_k(m_i, m_j))
\]

where \( \delta_k \) is a fix weight considered for each feature \( f_k \). Since the first three passes of the multi-pass system contain higher precision features, \( \delta_k \) is considered +1 for such features (\( \delta_k \) is -1 for the features of the pass 2 which adds a negative edge). \( \delta_k \) is set to 0.25 for other features. The pre-processing pipelines of both variants of the proposed system are the same as that of (Bengtson and Roth, 2008).

<table>
<thead>
<tr>
<th>Proposed system</th>
<th>MUC</th>
<th>B-Cubed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>ACE2004-NWIRE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>single pass</td>
<td>53.0</td>
<td>72.1</td>
</tr>
<tr>
<td>multi-pass</td>
<td>69.1</td>
<td>67.2</td>
</tr>
<tr>
<td>ACE2004-ROTH-DEV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>single pass</td>
<td>52.8</td>
<td>70.8</td>
</tr>
<tr>
<td>multi-pass</td>
<td>69.1</td>
<td>67.1</td>
</tr>
<tr>
<td>ACE2004-CULOTTA-TEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>single pass</td>
<td>52.5</td>
<td>73.5</td>
</tr>
<tr>
<td>multi-pass</td>
<td>63.5</td>
<td>60.7</td>
</tr>
</tbody>
</table>

Table 1. Experimental results.

As it is shown, results of the multi-pass system are considerably higher than that of the single pass variant. However, we still need some further work to reach the performance of more successful unsupervised coreference systems (e.g. Haghhighi and Klein, 2009; Raghunathan et al., 2010; Lee et al., 2011).

### 7. Error Analysis

Table 2 shows the number of pairwise errors made by the proposed multi-pass system on ACE2004-NWIRE. As it is shown, most errors are made on the nominal mentions with nominal antecedents. There are several reasons causing a rather high rate for this type of errors. Typically, such errors are caused by wrong head match assumptions, and missing semantic and syntactic compatibility information of the two nominal mentions. Raghunathan et al. (2010) just used the first mention of each cluster at each pass; the first mention of each cluster is more representative than other mentions of that cluster. This can reduce errors such as those made by some wrong head match assumptions. Raghunathan et al. (2010) also used additional syntactic information like “predicate nominative” and “not i-within-i”.

<table>
<thead>
<tr>
<th>Antecedent Type</th>
<th>PROPER</th>
<th>NOMINAL</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPER</td>
<td>241/1140</td>
<td>53/171</td>
<td>2943/1131</td>
</tr>
<tr>
<td>NOMINAL</td>
<td>56/257</td>
<td>493/921</td>
<td>549/1178</td>
</tr>
<tr>
<td>PRONOUN</td>
<td>268/566</td>
<td>285/451</td>
<td>553/1017</td>
</tr>
</tbody>
</table>

Table 2: Pairwise errors made by our model on ACE2004-NWIRE. Each cell indicates error rate made on the specified configuration.

Using additional linguistic knowledge such as parse trees, binding theory, salience hierarchy, richer semantic
knowledge, and cluster-wise feature sets can further decrease coreference decision errors. We intend to use such information in the next version of the proposed system.

8. Conclusions and Future Works

In this paper, we examined and evaluated the applicability of relaxation labeling to unsupervised coreference resolution, which has been inspired by the earlier work of Sapena et al. (2010), who used relaxation labeling technique for supervised coreference resolution. In comparison to (Sapena et al., 2010), our model is totally unsupervised (i.e., it does not need any labeled data for determining edge weights), and its major novelty lies in the proposed hierarchical graph development algorithm. This hierarchical graph construction method prevents the small numbers of high precision features to be overwhelmed by the larger number low precision ones. In hierarchical graph construction, in each step instead of considering the whole set of neighbors, the labeling algorithm considers just the most reliable set of neighbors for labeling a mention.

Although the presented system underperforms the state-of-the-art systems, it shows promising results and can be further improved in several ways. A natural way to extend the model is to incorporate more linguistic knowledge sources, such as those used by (Haghighi and Klein, 2009; Raghunathan et al., 2010).

References


