Coreference or Not: A Twin Model for Coreference Resolution

Xiaoqiang Luo
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598, U.S.A.
{xiaoluo}@us.ibm.com

Abstract

A twin-model is proposed for coreference resolution: a *link* component, modeling the coreferential relationship between an anaphor and a candidate antecedent, and a *creation* component modeling the possibility that a phrase is not coreferential with any candidate antecedent. The creation model depends on all candidate antecedents and is often expensive to compute; Therefore constraints are imposed on feature forms so that features in the creation model can be efficiently computed from feature values in the link model. The proposed twin-model is tested on the data from the 2005 Automatic Content Extraction (ACE) task and the proposed model performs better than a thresholding baseline without tuning free parameter.

1 Introduction

Coreference resolution aims to find multiple mentions of an entity (e.g., PERSON, ORGANIZATION) in a document. In a typical machine learning-based coreference resolution system (Soon et al., 2001; Ng and Cardie, 2002b; Yang et al., 2003; Luo et al., 2004), a statistical model is learned from training data and is used to measure how likely an anaphor [1] is coreferential to a candidate antecedent. A related, but often overlooked, problem is that the anaphor may be non-coreferential to any candidate, which arises from scenarios such as an identified anaphor is truly generic and there does not exist an antecedent in the discourse context, or an anaphor is the first mention (relative to processing order) in a coreference chain.

In (Soon et al., 2001; Ng and Cardie, 2002b), the problem is treated by thresholding the scores returned by the coreference model. That is, if the maximum coreference score is below a threshold, then the anaphor is deemed non-referential to any candidate antecedent. The threshold approach does not model non-coreferential events directly, and is by no means the optimal approach to the problem. It also introduces a free parameter which has to be set by trial-and-error. As an improvement, Ng and Cardie (2002a) and Ng (2004) train a separate model to classify an anaphor as either *anaphoric* or *non-anaphoric*. The output of this classifier can be used either as a pre-filter (Ng and Cardie, 2002a) so that *non-anaphoric* anaphors will not be precessed in the coreference system, or as a set of features in the coreference model (Ng, 2004). By rejecting any anaphor classified as *non-anaphoric* in coreference resolution, the filtering approach is meant to handle non-anaphoric phrases (i.e., no antecedent exists in the discourse under consideration), not the first mention in a coreference chain.

In this paper, coreference is viewed as a process of sequential operations on anaphor mentions: an anaphor can either be *linked* with its antecedent if the antecedent is available or present. If the anaphor, on the other hand, is discourse new (relative to the process order), then a new entity is *created*. Corresponding to the two types of operations, a twin-model is proposed to resolve coreferential relationships in a document. The first component is a statistical model measuring how likely an anaphor is coreferential to a candidate antecedent; The second one explicitly models the non-

[1] In this paper, "anaphor" includes all kinds of phrases to be resolved, which can be named, nominal or pronominal phrases.

Proceedings of NAACL HLT 2007, pages 73–80,
Rochester, NY, April 2007.

coreferential events. Both models are trained automatically and are used simultaneously in the coreference system. The twin-model coreference system is tested on the 2005 ACE (Automatic Content Extraction, see (NIST, 2005)) data and the best performance under both ACE-Value and entity F-measure can be obtained without tuning a free parameter.

The rest of the paper is organized as follows. The twin-model is presented in Section 2. A maximum-entropy implementation and features are then presented in Section 3. The experimental results on the 2005 ACE data is presented in Section 4. The proposed twin-model is compared with related work in Section 5 before the paper is concluded.

2 Coreference Model

A phrasal reference to an entity is called a mention. A set of mentions referring to the same physical object is said to belong to the same entity. For example, in the following sentence:

(I) `John` `said` `Mary` `was` `his` `sister`.

there are four mentions: `John`, `Mary`, `his`, and `sister`. `John` and `his` belong to the same entity since they refer to the same person; So do `Mary` and `sister`. Furthermore, `John` and `Mary` are *named* mentions, `sister` is a *nominal* mention and `his` is a *pronominal* mention.

In our coreference system, mentions are processed sequentially, though not necessarily in chronological order. For a document with n mentions $\{m_i : 1 \leq i \leq n\}$, at any time $t(t > 1)$, mention m_1 through m_{t-1} have been processed and each mention is placed in one of $N_t(N_t \leq (t-1))$ entities: $E_t = \{e_j : 1 \leq j \leq N_t\}$. Index i in m_i indicates the order in which it is processed, not necessarily the order in which it appears in a document. The basic step is to extend E_t to E_{t+1} with m_t.

Let us use the example in Figure 1 to illustrate how this is done. Note that Figure 1 contains one possible processing order for the four mentions in Example (I): first name mentions are processed, followed by nominal mentions, followed by pronominal mentions. At time $t = 1$, there is no existing entity and the mention m_1=`John` is placed in an initial entity (entity is signified by a solid rectangle). At time $t = 2$, m_2=`Mary` is processed and a new entity containing `Mary` is created. At time $t = 3$, the nominal mention m_3=`sister` is processed. At this point, the set of existing entities

$$E_3 = \Big\{\{\texttt{John}\}, \{\texttt{Mary}\}\Big\}.$$

m_3 is linked with the existing entity $\{\texttt{Mary}\}$. At the last step $t = 4$, the *pronominal* mention `his` is linked with the entity $\{\texttt{John}\}$.

The above example illustrates how a sequence of coreference steps lead to a particular coreference result. Conversely, if the processing order is known and fixed, every possible coreference result can be decomposed and mapped to a unique sequence of such coreference steps. Therefore, if we can score the set of coreference sequences, we can score the set of coreference results as well.

In general, when determining if a mention m_t is coreferential with any entity in E_t, there are two types of actions: one is that m_t is coreferential with one of the entities; The other is that m_t is not coreferential with any. It is important to distinguish the two cases for the following reason: if m_t is coreferential with an entity e_j, in most cases it is sufficient to determine the relationship by examining m_t and e_j, and their local context; But if m_t is not coreferential with any existing entities, we need to consider m_t with all members in E_t. This observation leads us to propose the following twin-model for coreference resolution.

The first model, $P(L|e_j, m_t)$, is conditioned on an entity e_j and the current mention m_t and measure how likely they are coreferential. L is a binary variable, taking value 1 or 0, which represents positive and negative coreferential relationship, respectively. The second model, on the other hand, $P(C|E_t, m_t)$, is conditioned on the past entities E_t and the current mention m_t. The random variable C is also binary: when C is 1, it means that a new entity $\{m_t\}$ will be created. In other words, the second model measures the probability that m_t is *not* coreferential to any existing entity. To avoid confusion in the subsequent presentation, the first model will be written as $P_l(\cdot|e_j, m_t)$ and called *link* model; The second model is written as $P_c(\cdot|E_t, m_t)$ and called *creation* model.

For the time being, let's assume that we have the link and creation model at our disposal, and we will show how they can be used to score coreference decisions.

Given a set of existing entities $E_t = \{e_j\}_1^{N_t}$, formed by mentions $\{m_i\}_{i=1}^{t-1}$, and the current mention m_t, there are $N_t + 1$ possible actions: we can either link m_t with an existing entity e_j $(j = 1, 2, \cdots, N_t)$, or create a new entity containing m_t. The link action between e_j and m_t can be scored by $P_l(1|e_j, m_t)$ while the creation action can be measured by $P_c(1|E_t, m_t)$. Each possible coreference outcome consists of n such actions $\{a_t : t = 1, 2, \cdots, n\}$, each of which can be scored by either the link model $P_l(\cdot|e_j, m_t)$ or the cre-

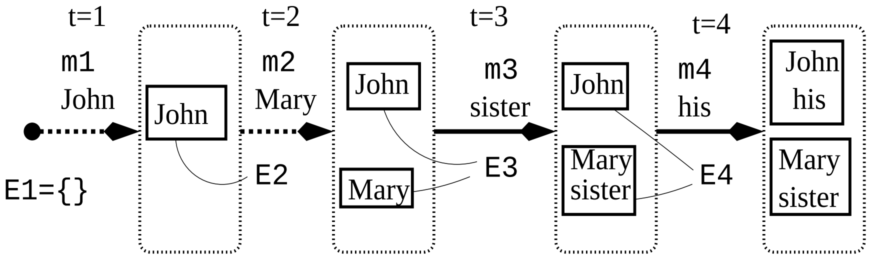

Figure 1: Coreference process for the four mentions in Example (I). Mentions in a document are processed sequentially: first name mentions, then nominal mentions, and then pronominal mentions. A dashed arrow signifies that a new entity is created, while a solid arrow means that the current mention is linked with an existing entity.

ation model $P_c(\cdot|E_t, m_t)$. Denote the score for action a_t by $S(a_t|a_1^{t-1})$, where dependency of a_t on a_1 through a_{t-1} is emphasized. The coreference result corresponding to the action sequence is written as $E_n(\{a_i\}_{i=1}^n)$. When it is clear from context, we will drop $\{a_i\}_{i=1}^n$ and write E_n only.

With this notation, the score for a coreference outcome $E_n(\{a_i\}_{i=1}^n)$ is the product of individual scores assigned to the corresponding action sequence $\{a_i\}_{i=1}^n$, and the best coreference result is the one with the highest score:

$$\begin{aligned}\hat{E}_n &= arg\max_{E_n} S(E_n)\\ &= arg\max_{\{a_t\}_1^n} \prod_{t=1}^{n} S(a_t|a_1^{t-1}).\end{aligned} \tag{1}$$

Given n mentions, the number of all possible entity outcomes is the Bell Number (Bell, 1934): $B(n) = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}$. Exhaustive search is out of the question. Thus, we organize hypotheses into a Bell Tree (Luo et al., 2004) and use a beam search with the following pruning strategy: first, a maximum beam size (typically 20) S is set, and we keep only the top S hypotheses; Second, a relative threshold r (we use 10^{-5}) is set to prune any hypothesis whose score divided by the maximum score falls below the threshold.

To give an concrete example, we use the example in Figure 1 again. The first step at $t = 1$ creates a new entity and is therefore scored by $P_c(1|\{\}, \texttt{John})$; the second step also creates an entity and is scored by $P_c(1|\{\texttt{John}\}, \texttt{Mary})$; the step $t = 3$, however, links `sister` with $\{\texttt{Mary}\}$ and is scored by $P_l(1|\{\texttt{Mary}\}, \texttt{sister})$; Similarly, the last step is scored by $P_l(1|\{\texttt{John}\}, \texttt{his})$. The score for this coreference outcome is the product of the four numbers:

$$\begin{aligned}&S(\Big\{\{\texttt{John,his}\}, \{\texttt{Mary,sister}\}\Big\})\\ =&P_c(1|\{\}, \texttt{John})P_c(1|\{\texttt{John}\}, \texttt{Mary})\cdot\\ &P_l(1|\{\texttt{Mary}\}, \texttt{sister})\cdot\\ &P_l(1|\{\texttt{John}\}, \texttt{his}).\end{aligned} \tag{2}$$

Other coreference results for these four mentions can be scored similarly. For example, if `his` at the last step is linked with $\{\texttt{Mary,sister}\}$, the score would be:

$$\begin{aligned}&S(\Big\{\{\texttt{John}\}, \{\texttt{Mary,sister,his}\}\Big\})\\ =&P_c(1|\{\}, \texttt{John})P_c(1|\{\texttt{John}\}, \texttt{Mary})\cdot\\ &P_l(1|\{\texttt{Mary}\}, \texttt{sister})\cdot\\ &P_l(1|\{\texttt{Mary,sister}\}, \texttt{his}).\end{aligned} \tag{3}$$

At testing time, (2) and (3), among other possible outcomes, will be searched and compared, and the one with the highest score will be output as the coreference result.

Examples in (2) and (3) indicate that the link model $P_l(\cdot|e_j, m_t)$ and creation model $P_c(\cdot|E_t, m_t)$ form an integrated coreference system and are applied simultaneously at testing time. As will be shown in the next section, features in the creation model $P_c(\cdot|E_t, m_t)$ can be computed from their counterpart in the link model $P_l(\cdot|e_j, m_t)$ under some mild constraints. So the two models' training procedures are tightly coupled. This is different from (Ng and Cardie, 2002a; Ng, 2004) where their anaphoricty models are trained independently of the coreference model, and it is either used as a pre-filter, or its output is used as features in the coreference model. The creation model $P_c(\cdot|E_t, m_t)$ proposed here bears similarity to the *starting* model in (Luo et al., 2004). But there is a crucial difference: the *starting* model in (Luo et al., 2004) is an ad-hoc use of the link scores and is not learned automatically, while $P_c(\cdot|E_t, m_t)$ is fully trained. Training $P_c(\cdot|E_t, m_t)$ is covered in the next section.

3 Implementation

3.1 Feature Structure

To implement the twin model, we adopt the log linear or maximum entropy (MaxEnt) model (Berger et al., 1996) for its flexibility of combining diverse sources of information. The two models are of the form:

$$P_l(L|e_j, m_t) = \frac{exp\Big(\sum_k \lambda_k g_k(e_j, m_t, L)\Big)}{Y(e_j, m_t)} \quad (4)$$

$$P_c(C|E_t, m_t) = \frac{exp\Big(\sum_i \nu_i h_i(E_t, m_t, C)\Big)}{Z(E_t, m_t)}, \quad (5)$$

where L and C are binary variables indicating either m_t is coreferential with e_j, or m_t is used to create a new entity. $Y(e_j, m_t)$ and $Z(e_j, m_t)$ are normalization factors to ensure that $P_l(\cdot|e_j, m_t)$ and $P_c(\cdot|E_t, m_t)$ are probabilities; λ_k and ν_i are the weights for feature $g_k(e_j, m_t, L)$ and $h_i(E_t, m_t, C)$, respectively. Once the set of features functions are selected, algorithm such as improved iterative scaling (Berger et al., 1996) or sequential conditional generalized iterative scaling (Goodman, 2002) can be used to find the optimal parameter values of $\{\lambda_k\}$ and $\{\nu_i\}$.

Computing features $\{g_k(e_j, m_t, \cdot)\}$ for the link model $P_l(L|e_j, m_t)$ [2] is relatively straightforward: given an entity e_j and the current mention m_t, we just need to characterize things such as lexical similarity, syntactic relationship, and/or semantic compatibility of the two. It is, however, very challenging to compute the features $\{h_i(E_t, m_t, \cdot)\}$ for the creation model $P_c(\cdot|E_t, m_t)$ since its conditioning includes a set of entities E_t, whose size grows as more and more mentions are processed. The problem exists because the decision of creating a new entity with m_t has to be made after examining all preceding entities. There is no reasonable modeling assumption one can make to drop some entities in the conditioning.

To overcome the difficulty, we impose the following constraints on the features of the link and creation model:

$$\begin{aligned} g_k(e_j, m_t, L) =& g_k^{(1)}(e_j, m_t) g_k^{(2)}(L) \quad (6)\\ h_i(E_t, m_t, C) =& h_i^{(1)}\big(\{g_k^{(1)}(e, m_t) : e \in E_t\}\big)\cdot \\ & h_i^{(2)}(C), \text{ for some } k. \quad (7) \end{aligned}$$

(6) states that a feature in the link model is separable and can be written as a product of two functions: the first one, $g_k^{(1)}(\cdot, \cdot)$, is a binary function depending on the conditioning part only; the second one, $g_k^{(2)}(\cdot)$, is an indicator function depending on the prediction part L only. Like $g_k^{(2)}(\cdot)$, $h_i^{(2)}(\cdot)$ is also a binary indicator function.

(7) implies that features in the creation model are also separable; Moreover, the conditioning part $h_i^{(1)}\big(\{g_k^{(1)}(e, m_t) : e \in E_t\}\big)$, also a binary function, only depends on the function values of the set of link features $\{g_k^{(1)}(e, m_t) : e \in E_t\}$ (for some k). In other words, once $\{g_k^{(1)}(e, m_t) : e \in E_t\}$ and C are known, we can compute $h_i(E_t, m_t, C)$ without actually comparing m_t with any entity in E_t. Using binary features is a fairly mild constraint as non-binary features can be replaced by a set of binary features through quantization.

How fast $h_i^{(1)}\big(\{g_k^{(1)}(e, m_t) : e \in E_t\}\big)$ can be computed depends on how $h_i^{(1)}$ is defined. In most cases – as will be shown in Section 3.2, it boils down testing if any member in $\{g_k^{(1)}(e, m_t) : e \in E_t\}$ is nonzero; or counting how many non-zero members there are in $\{g_k^{(1)}(e, m_t) : e \in E_t\}$. Both are simple operations that can be carried out quickly. Thus, the assumption (7) makes it possible to compute efficiently $h_i(E_t, m_t, C)$.

3.2 Features in the Creation Model

We describe features used in our coreference system. We will concentrate on features used in the creation model since those in the link model can be found in the literature (Soon et al., 2001; Ng and Cardie, 2002b; Yang et al., 2003; Luo et al., 2004). In particular, we show how features in the creation model can be computed from a set of feature values from the link model for a few example categories. Since $g_k^{(2)}(\cdot)$ and $h_i^{(2)}(\cdot)$ are simple indicator functions, we will focus on $g_k^{(1)}(\cdot, \cdot)$ and $h_i^{(1)}(\cdot)$.

3.2.1 Lexical Features

This set of features computes if two surface strings (spellings of two mentions) match each other, and are

[2] The link model is actually implemented as: $P_l(L|e_j, m_t) \approx \max_{m' \in e_j} \hat{P}_l(L|e_j, m', m_t)$. Some features are computed on a pair of mentions (m', m_t) while some are computed at entity level. See (Luo and Zitouni, 2005) and (Daumé III and Marcu, 2005).

applied to name and nominal mentions only. For the link model, a lexical feature $g_k^{(1)}(e_j, m_t)$ is 1 if e_j contains a mention matches m_t, where a match can be exact, partial, or one is an acronym of the other.

Since $g_k(e_j, m_t)$ is binary, one corresponding feature used in the creation model is the disjunction of the values in the link model, or

$$h_i^{(1)}(E_t, m_t) = \vee_{e \in E_t}\{g_k^{(1)}(e, m_t)\}, \qquad (8)$$

where $\vee$ is a binary "or" operator. The intuition is that if there is any mention in E_t matching m_t, then the probability to create a new entity with m_t should be low; Conversely, if none of the mentions in E_t matches m_t, then m_t is likely to be the first mention of a new entity.

Take $t = 2$ in Figure 1 as an example. There is only one partially-established entity $\{\texttt{John}\}$, so $E_2 = \{\texttt{John}\}$, and $m_2 = \texttt{Mary}$. The exact string match feature $g_{em}^{(1)}(\cdot, \cdot)$ would be

$$g_{em}^{(1)}(\{\texttt{John}\}, \texttt{Mary}) = 0,$$

and the corresponding string match feature in the creation model is

$$\begin{aligned} h_{em}^{(1)}(\{\texttt{John}\}, \texttt{Mary}) &= \vee_{e \in E_t}\{g_{em}^{(1)}(e, \texttt{Mary})\} \\ &= 0. \end{aligned}$$

Disjunction is not the only operation we can use. Another possibility is counting how many times m_t matches mentions in E_t, so (8) becomes:

$$h_i^{(1)}(E_t, m_t) = Q\big[\sum_{e \in E_t}\{g_k^{(1)}(e, m_t)\}\big], . \qquad (9)$$

where $Q[\cdot]$ quantizes raw counts into bins.

3.2.2 Attribute Features

In the link model, features in this category compare the properties of the current mention m_t with that of an entity e_j. Properties of a mention or an entity, whenever applicable, include gender, number, entity type, reflexivity of pronouns etc. Similar to what done in the lexical feature, we again synthesize a feature in the creation model by taking the disjunction of the corresponding set of feature values in the link model, or

$$h_i^{(1)}(E_t, m_t) = \vee_{e \in E_t}\{g_k^{(1)}(e, m_t)\},$$

where $g_k^{(1)}(e, m_t)$ takes value 1 if entity e and mention m_t share the same property; Otherwise its value is 0. The intuition is that if there is an entity having the same property as the current mention, then the probability for the current mention to be linked with the entity should be higher than otherwise; Conversely, if none of the entities in E_t shares a property with the current mention, the probability for the current mention to create a new entity ought to be higher.

Consider the gender attribute at $t = 4$ in Figure 1. Let $g_{gender}^{(1)}(\cdot, \cdot)$ be the gender feature in the link model, assume that we know the gender of John, Mary and his. Then $g_{gender}^{(1)}(\{\{\texttt{John}\}, \texttt{his})$ is 1, while $g_{gender}^{(1)}(\{\texttt{Mary, sister}\}, \texttt{his})$ is 0. Therefore, the gender feature for the creation model would be

$$\begin{aligned} &h_{gender}^{(1)}(\{\{\texttt{John}\}, \{\texttt{Mary, sister}\}\}, \texttt{his}) \\ =&0 \vee 1 = 1, \end{aligned}$$

which means that there is at least one mention which has the same the gender of the current mention m_t.

3.2.3 Distance Feature

Distance feature needs special treatment: while it makes sense to talk about the distance between a pair of mentions, it is not immediately clear how to compute the distance between a set of entities E_t and a mention m_t. To this end, we compute the minimum distance between the entities and the current mention with respect to a "fired" link feature, as follows.

For a particular feature $g_k^{(1)}(\cdot, \cdot)$ in the link model, define the minimum distance to be

$$\begin{aligned} \hat{d}(E_t, m_t; g_k) = \min\{d(m, m_t) : m \in E_t, \\ \text{and } g_k^{(1)}(m, m_t) = 1\}, \qquad (10) \end{aligned}$$

where $d(m, m_t)$ is the distance between mention $m$and m_t. The distance itself can be the number of tokens, or the number of intervening mentions, or the number of sentences. The minimum distance $\hat{d}(E_t, m_t; g_k)$ is quantized and represented as binary feature in the creation model. The idea here is to encode what is the nearest place where a feature fires.

Again as an example, consider the gender attribute at $t = 4$ in Figure 1. Assuming that $d(m, m_t)$ is the number of tokens. Since only John matches the gender of his,

$$\hat{d}(E_4, m_4; g_{gender}) = 3.$$

The number is then quantized and used as a binary feature to encode the information that "there is a mention whose gender matches the current mention within in a token distance range including 3."

In general, binary features in the link model which measure the similarity between an entity and a mention can be turned into features in the creation model in the same manner as described in Section 3.2.1 and 3.2.2. For example, syntactic features (Ng and Cardie, 2002b; Luo and Zitouni, 2005) can be computed this way and are used in our system.

4 Experiments

4.1 Data and Evaluation Metric

We report the experimental results on ACE 2005 data (NIST, 2005). The dataset consists of 599 documents from a rich and diversified sources, which include newswire articles, web logs, and Usenet posts, transcription of broadcast news, broadcast conversations and telephone conversations. We reserve the last 16% documents of each source as the test set and use the rest of the documents as the training set. Statistics such as the number of documents, words, mentions and entities of this data split is tabulated in Table 1.

DataSet	#Docs	#Words	#Mentions	#Entities
Training	499	253771	46646	16102
Test	100	45659	8178	2709
Total	599	299430	54824	18811

Table 1: Statistics of ACE 2005 data: number of documents, words, mentions and entities in the training and test set.

The link and creation model are trained at the same time. Besides the basic feature categories described in Section 3.2, we also compute composite features by taking conjunctions of the basic features. Features are selected by their counts with a threshold of 8.

ACE-Value is the official score reported in the ACE task and will be used to report our coreference system's performance. Its detailed definition can be found in the official evaluation document [3]. Since ACE-Value is a weighted metric measuring a coreference system's relative value, and it is not sensitive to certain type of errors (e.g., false-alarm entities if these entities contain correct mentions), we also report results using unweighted entity F-measure.

4.2 Results

To compare the proposed twin model with simple thresholding (Soon et al., 2001; Ng and Cardie, 2002b), we first train our twin model. To simulate the thresholding approach, a baseline coreference system is created by replacing the creation model with a constant, i.e.,

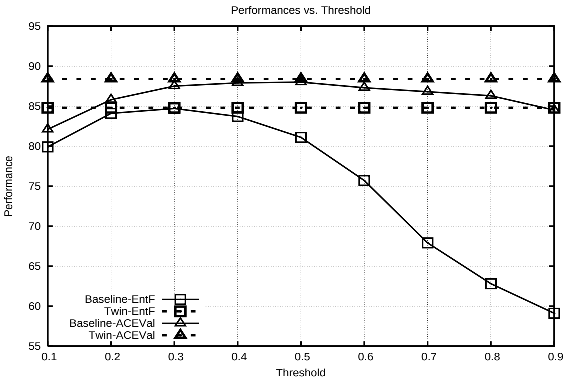

Figure 2: Performance comparison between a thresholding baseline and the twin-model: lines with square points are the entity F-measure (x100) results; lines with triangle points are ACE-Value (in %). Solid lines are baseline while dashed lines are twin-model.

$$P_c(1|E_t, m_t) = \theta, \qquad (11)$$

where θ is a number between 0 and 1. At testing time, a new entity is created with score θ when

$$P_l(1|e_j, m_t) < \theta, \ \ \forall e_j \in E_t.$$

The decision rule simply implies that if the scores between the current mention m_t and all candidate entities $e_j \in E_t$ are below the threshold θ, a new entity will be created.

Performance comparison between the baseline and the twin-model is plotted in Figure 2. X-axis is the threshold varying from 0.1 to 0.9 with a step size 0.1. Two metrics are used to compare the results: two lines with square data points are the entity F-measure results, and two lines with triangle points are ACE-Value. Note that performances for the twin-model are constant since it does not use thresholding.

As shown in the graph, the twin-model (two dashed lines) always outperforms the baseline (two solid lines). A "bad" threshold impacts the entity F-measure much more than ACE-Value, especially in the region with high threshold value. Note that a large θ will lead to more false-alarm entities. The graph suggests that ACE-Value is much less sensitive than the un-weighted F-measure in measuring false-alarm errors. For example, at $\theta = 0.9$, the baseline F-measure is 0.591 while

[3] The official evaluation document can be found at: `www.nist.gov/speech/tests/ace/ace05/doc/ace05-evalplan.v3.pdf`.

the twin model F-measure is 0.848, a 43.5% difference; On the other hand, the corresponding ACE-Values are 84.5% (baseline) vs. 88.4% (twin model), a mere 4.6% relative difference. There are at least two reasons: first, ACE-Value discounts importance of nominal and pronoun entities, so more nominal and pronoun entity errors are not reflected in the metric; Second, ACE-Value does not penalize false-alarm entities if they contain correct mentions. The problem associated with ACE-Value is the reason we include the entity F-measure results.

Another interesting observation is that an optimal threshold for the entity F-measure is not necessarily optimal for ACE-Value, and vice versa: $\theta = 0.3$ is the best threshold for the entity F-measure, while $\theta = 0.5$ is optimal for ACE-Value. This is highlighted in Table 2, where row "B-opt-F" contains the best results optimizing the entity F-measure (at $\theta = 0.3$), row "B-opt-AV" contains the best results optimizing ACE-Value (at $\theta = 0.5$), and the last line "Twin-model" contains the results of the proposed twin-model. It is clear from Table 2 that thresholding cannot be used to optimize the entity F-measure and ACE-Value simultaneously. A sub-optimal threshold could be detrimental to an unweighted metric such as the entity F-measure. The proposed twin model eliminates the need for thresholding, a benefit of using the principled creation model. In practice, the optimal threshold is a free parameter that has to be tuned every time when a task, dataset and model changes. Thus the proposed twin model is more portable when a task or dataset changes.

System	F-measure	ACE-Value
B-opt-F	84.7	87.5
B-opt-AV	81.1	88.0
Twin-model	**84.8**	**88.4**

Table 2: Comparison between the thresholding baseline and the twin model: optimal threshold depends on performance metric. The proposed twin-model outperforms the baseline without tuning the free parameter.

5 Related Work

Some earlier work (Lappin and Leass, 1994; Kennedy and Boguraev, 1996) use heuristic to determine whether a phrase is anaphoric or not. Bean and Riloff (1999) extracts rules from non-anaphoric noun phrases and noun phrases patterns, which are then applied to test data to identify *existential* noun phrases. It is intended as as pre-filtering step before a coreference resolution system is run. Ng and Cardie (2002a) trains a separate anaphoricity classifier in addition to a coreference model. The anaphoricity classifier is applied as a filter and only anaphoric mentions are later considered by the coreference model. Ng (2004) studies what is the best way to make use of anaphoricity information and concludes that the constrained-based and globally-optimized approach works the best. Poesio et al. (2004) contains a good summary of recent research work on discourse new or anaphoricity. Luo et al. (2004) uses a *start* model to determine whether a mention is the first one in a coreference chain, but it is computed ad hoc without training. Nicolae and Nicolae (2006) constructs a graph where mentions are nodes and an edge represents the likelihood two mentions are in an entity, and then a graph-cut algorithm is employed to produce final coreference results.

We take the view that determining whether an anaphor is coreferential with any candidate antecedent is part of the coreference process. But we do recognize that the disparity between the two types of events: while a coreferential relationship can be resolved by examining the local context of the anaphor and its antecedent, it is necessary to compare the anaphor with all the preceding candidates before it can be declared that it is not coreferential with any. Thus, a *creation* component $P_c(\cdot|E_t, m_t)$ is needed to model the second type of events. A problem arising from the adoption of the creation model is that it is very expensive to have a conditional model depending on all preceding entities E_t. To solve this problem, we adopt the MaxEnt model and impose some reasonable constraints on the feature functions, which makes it possible to synthesize features in the creation model from those of the link model. The twin model components are intimately trained and used simultaneously in our coreference system.

6 Conclusions

A twin-model is proposed for coreference resolution: one *link* component computes how likely a mention is coreferential with a candidate entity; the other component, called *creation* model, computes the probability that a mention is not coreferential with any candidate entity. Log linear or MaxEnt approach is adopted for building the two components. The twin components are trained and used simultaneously in our coreference system.

The creation model depends on all preceding entities and is often expensive to compute. We impose some reasonable constraints on feature functions which makes it feasible to compute efficiently the features in the creation model from a subset of link feature values. We test the proposed twin-model on the ACE 2005 data and the proposed model outperforms a thresholding baseline. Moreover, it is observed that the optimal threshold in the baseline depends on performance metric, while the proposed model eliminates the need of tuning the optimal threshold.

Acknowledgments

This work was partially supported by the Defense Advanced Research Projects Agency under contract No. HR0011-06-2-0001. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the U.S. government and no official endorsement should be inferred.

I would like to thank Salim Roukos for helping to improve the writing of the paper. Suggestions and comments from three anonymous reviewers are also gratefully acknowledged.

References

David L. Bean and Ellen Riloff. 1999. Corpus-based identification of non-anaphoric noun phrases. In *Proc. ACL*.

E.T. Bell. 1934. Exponential numbers. *Amer. Math. Monthly*, pages 411–419.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, March.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proc. of HLT and EMNLP*, pages 97–104, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Joshua Goodman. 2002. Sequential conditional generalized iterative scaling. In *Pro. of the 40th ACL*.

Christopher Kennedy and Branimir Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of COLING-96 (16th International Conference on Computational Linguistics)*, Copenhagen,DK.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4), December.

Xiaoqiang Luo and Imed Zitouni. 2005. Multi-lingual coreference resolution with syntactic features. In *Proc. of Human Language Technology (HLT)/Empirical Methods in Natural Language Processing (EMNLP)*.

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

Vincent Ng and Claire Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of COLING*.

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

Vincent Ng. 2004. Learning noun phrase anaphoricity to improve conference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 151–158, Barcelona, Spain, July.

Cristina Nicolae and Gabriel Nicolae. 2006. BESTCUT: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 275–283, Sydney, Australia, July. Association for Computational Linguistics.

NIST. 2005. ACE 2005 evaluation. www.nist.gov/speech/tests/ace/ace05/index.htm.

M. Poesio, O. Uryupina, R. Vieira, M. Alexandrov-Kabadjov, and R. Goulart. 2004. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *ACL 2004: Workshop on Reference Resolution and its Applications*, pages 47–54, Barcelona, Spain, July.

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.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competition learning approach. In *Proc. of the* 41^{st} *ACL*.