On Coreference Resolution Performance Metrics

Xiaoqiang Luo
1101 Kitchawan Road, Room 23-121
IBM T.J. Wastson Research Center
Yorktown Heights, NY 10598, U.S.A.
xiaoluo@us.ibm.com

Abstract

The paper proposes a Constrained Entity-Alignment F-Measure (CEAF) for evaluating coreference resolution. The metric is computed by aligning reference and system entities (or coreference chains) with the constraint that a system (reference) entity is aligned with at most one reference (system) entity. We show that the best alignment is a maximum bipartite matching problem which can be solved by the Kuhn-Munkres algorithm. Comparative experiments are conducted to show that the widely-known MUC F-measure has serious flaws in evaluating a coreference system. The proposed metric is also compared with the ACE-Value, the official evaluation metric in the Automatic Content Extraction (ACE) task, and we conclude that the proposed metric possesses some properties such as symmetry and better interpretability missing in the ACE-Value.

1 Introduction

A working definition of coreference resolution is partitioning the noun phrases we are interested in into equivalence classes, each of which refers to a physical entity. We adopt the terminologies used in the Automatic Content Extraction (ACE) task (NIST, 2003a) and call each individual phrase a *mention* and equivalence class an *entity*. For example, in the following text segment,

> (1): "The American Medical Association voted yesterday to install the heir apparent as its president-elect, rejecting a strong, upstart challenge by a district doctor who argued that the nation's largest physicians' group needs stronger ethics and new leadership."

mentions are underlined, "American Medical Association", "its" and "group" refer to the same organization (object) and they form an entity. Similarly, "the heir apparent" and "president-elect" refer to the same person and they form another entity. It is worth pointing out that the entity definition here is different from what used in the Message Understanding Conference (MUC) task (MUC, 1995; MUC, 1998) – ACE entity is called coreference chain or equivalence class in MUC, and ACE mention is called entity in MUC.

An important problem in coreference resolution is how to evaluate a system's performance. A good performance metric should have the following two properties:

- Discriminativity: This refers to the ability to differentiate a good system from a bad one. While this criterion sounds trivial, not all performance metrics used in the past possess this property.

- Interpretability: A good metric should be easy to interpret. That is, there should be an intuitive sense of how good a system is when a metric suggests that a certain percentage of coreference results are correct. For example, when a metric reports 95% or above correct for a system, we would expect that the vast majority of mentions are in right entities or coreference chains.

A widely-used metric is the link-based F-measure (Vilain et al., 1995) adopted in the MUC task. It is computed by first counting the number of common links between the reference (or "truth") and the system output (or "response"); the link precision is the number of common links divided by the number of links in the system output, and the link recall is the number of common links divided by the number of links in the reference. There are known problems associated with the link-based F-measure. First, it ignores single-mention entities since no link can be found in these entities; Second, and more importantly, it fails to distinguish system outputs with different qualities: the link-based F-measure intrinsically favors systems producing fewer entities, and may result

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 25–32, Vancouver, October 2005.

in higher F-measures for worse systems. We will revisit these issues in Section 3.

To counter these shortcomings, Bagga and Baldwin (1998) proposed a B-cubed metric, which first computes a precision and recall for each individual mention, and then takes the weighted sum of these individual precisions and recalls as the final metric. While the B-cubed metric fixes some of the shortcomings of the MUC F-measure, it has its own problems: for example, the mention precision/recall is computed by comparing entities containing the mention and therefore an entity can be used more than once. The implication of this drawback will be revisited in Section 3.

In the ACE task, a value-based metric called ACE-value (NIST, 2003b) is used. The ACE-value is computed by counting the number of false-alarm, the number of miss, and the number of mistaken entities. Each error is associated with a cost factor that depends on things such as entity type (e.g., "LOCATION", "PERSON"), and mention level (e.g., "NAME," "NOMINAL," and "PRONOUN"). The total cost is the sum of the three costs, which is then normalized against the cost of a nominal system that does not output any entity. The ACE-value is finally computed by subtracting the normalized cost from 1. A perfect coreference system will get a 100% ACE-value while a system outputs no entities will get a 0 ACE-value. A system outputting many erroneous entities could even get negative ACE-value. The ACE-value is computed by aligning entities and thus avoids the problems of the MUC F-measure. The ACE-value is, however, hard to interpret: a system with 90% ACE-value does not mean that 90% of system entities or mentions are correct, but that the cost of the system, relative to the one outputting no entity, is 10%.

In this paper, we aim to develop an evaluation metric that is able to measure the quality of a coreference system – that is, an intuitively better system would get a higher score than a worse system, and is easy to interpret. To this end, we observe that coreference systems are to recognize *entities* and propose a metric called Constrained Entity-Aligned F-Measure (CEAF). At the core of the metric is the optimal one-to-one map between subsets of reference and system entities: system entities and reference entities are aligned by maximizing the total entity similarity under the constraint that a reference entity is aligned with at most one system entity, and vice versa. Once the total similarity is defined, it is straightforward to compute recall, precision and F-measure. The constraint imposed in the entity alignment makes it impossible to "cheat" the metric: a system outputting too many entities will be penalized in precision while a system outputting two few entities will be penalized in recall. It also has the property that a perfect system gets an F-measure 1 while a system outputting no entity or no common mentions gets an F-measure 0. The proposed CEAF has a clear meaning: for mention-based CEAF, it reflects the percentage of mentions that are in the correct entities; For entity-based CEAF, it reflects the percentage of correctly recognized entities.

The rest of the paper is organized as follows. In Section 2, the Constrained Entity-Alignment F-Measure is presented in detail: the constraint entity alignment can be represented by a bipartite graph and the optimal alignment can be found by the Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957). We also present two entity-pair similarity measures that can be used in CEAF: one is the absolute number of common mentions between two entities, and the other is a "local" mention F-measure between two entities. The two measures lead to the mention-based and entity-based CEAF, respectively. In Section 3, we compare the proposed metric with the MUC link-based metric and ACE-value on both artificial and real data, and point out the problems of the MUC F-measure.

2 Constrained Entity-Alignment F-Measure

Some notations are needed before we present the proposed metric and the algorithm to compute the metric.

Let reference entities in a document d be

$$\mathcal{R}(d) = \{R_i : i = 1, 2, \cdots, |\mathcal{R}(d)|\},$$

and system entities be

$$\mathcal{S}(d) = \{S_i : i = 1, 2, \cdots, |\mathcal{S}(d)|\}.$$

To simplify typesetting, we will omit the dependency on d when it is clear from context, and write $\mathcal{R}(d)$ as $\mathcal{R}$ and $\mathcal{S}(d)$ as $\mathcal{S}$.

Let

$$m = \min\{|\mathcal{R}|, |\mathcal{S}|\}$$
$$M = \max\{|\mathcal{R}|, |\mathcal{S}|\},$$

and let $\mathcal{R}_m \subset \mathcal{R}$ and $\mathcal{S}_m \subset \mathcal{S}$ be any subsets with m entities. That is, $|\mathcal{R}_m| = m$ and $|\mathcal{S}_m| = m$. Let $G(\mathcal{R}_m, \mathcal{S}_m)$ be the set of one-to-one entity maps from $\mathcal{R}_m$ to $\mathcal{S}_m$, and G_m be the set of all possible one-to-one maps between the size-m subsets of $\mathcal{R}$ and $\mathcal{S}$. Or

$$G(\mathcal{R}_m, \mathcal{S}_m) = \{g : \mathcal{R}_m \mapsto \mathcal{S}_m\},$$
$$G_m = \cup_{(\mathcal{R}_m, \mathcal{S}_m)} G(\mathcal{R}_m, \mathcal{S}_m).$$

The requirement of one-to-one map means that for any $g \in G(\mathcal{R}_m, \mathcal{S}_m)$, and any $R \in \mathcal{R}_m$ and $R' \in \mathcal{R}_m$, we have that $R \neq R'$ implies that $g(R) \neq g(R')$, and $g(R) \neq g(R')$ implies that $R \neq R'$. Clearly, there are $m!$ one-to-one maps from $\mathcal{R}_m$ to $\mathcal{S}_m$ (or $|G(\mathcal{R}_m, \mathcal{S}_m)| = m!$), and $|G_m| = \binom{M}{m} m!$.

Let $\phi(R, S)$ be a "similarity" measure between two entities R and S. $\phi(R, S)$ takes non-negative value: zero

value means that R and S have nothing in common. For example, $\phi(R, S)$ could be the number of common mentions shared by R and S, and $\phi(R, R)$ the number of mentions in entity R.

For any $g \in G_m$, the total similarity $\Phi(g)$ for a map g is the sum of similarities between the aligned entity pairs: $\Phi(g) = \sum_{R \in \mathcal{R}_m} \phi(R, g(R))$. Given a document d, and its reference entities $\mathcal{R}$ and system entities $\mathcal{S}$, we can find the best alignment maximizing the total similarity:

$$\begin{aligned} g^* &= \arg\max_{g \in G_m} \Phi(g) \\ &= \arg\max_{g \in G_m} \sum_{R \in \mathcal{R}_m} \phi(R, g(R)). \end{aligned} \tag{1}$$

Let $\mathcal{R}^*_m$ and $\mathcal{S}^*_m = g^*(\mathcal{R}^*_m)$ denote the reference and system entity subsets where g^* is attained, respectively. Then the maximum total similarity is

$$\Phi(g^*) = \sum_{R \in \mathcal{R}^*_m} \phi(R, g^*(R)). \tag{2}$$

If we insist that $\phi(R, S) = 0$ whenever R or S is empty, then the non-negativity requirement of $\phi(R, S)$ makes it unnecessary to consider the possibility of mapping one entity to an empty entity since the one-to-one map maximizing $\Phi(g)$ must be in G_m.

Since we can compute the entity self-similarity $\phi(R, R)$ and $\phi(S, S)$ for any $R \in \mathcal{R}$ and $S \in \mathcal{S}$ (i.e., using the identity map), we are now ready to define the precision, recall and F-measure as follows:

$$p = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)} \tag{3}$$

$$r = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \tag{4}$$

$$F = \frac{2pr}{p + r}. \tag{5}$$

The optimal alignment g^* involves only $m = \min\{|\mathcal{R}|, |\mathcal{S}|\}$ reference and system entities, and entities not aligned do not get credit. Thus the F-measure (5) penalizes a coreference system that proposes too many (i.e., lower precision) or too few entities (i.e., lower recall), which is a desired property.

In the above discussion, it is assumed that the similarity measure $\phi(R, S)$ is computed for all entity pair (R, S). In practice, computation of $\phi(R, S)$ can be avoided if it is clear that R and S have nothing in common (e.g., if no mention in R and S overlaps, then $\phi(R, S) = 0$). These entity pairs are not linked and they will not be considered when searching for the optimal alignment. Consequently the optimal alignment could involve less than m reference and system entities. This can speed up considerably the F-measure computation when the majority of entity pairs have zero similarity. Nevertheless, summing over m entity pairs in the general formulae (2) does not change the optimal total similarity between $\mathcal{R}$ and $\mathcal{S}$ and hence the F-measure.

In formulae (3)-(5), there is only one document in the test corpus. Extension to corpus with multiple test documents is trivial: just accumulate statistics on the per-document basis for both denominators and numerators in (3) and (4), and find the ratio of the two.

So far, we have tacitly kept abstract the similarity measure $\phi(R, S)$ for entity pair R and S. We will defer the discussion of this metric to Section 2.2. Instead, we first present the algorithm computing the F-measure (3)-(5).

2.1 Computing Optimal Alignment and F-measure

A naive implementation of (1) would enumerate all the possible one-to-one maps (or alignments) between size-m (recall that $m = \min\{|\mathcal{R}|, |\mathcal{S}|\}$) subsets of $\mathcal{R}$ and size-m subsets of $\mathcal{S}$, and find the best alignment maximizing the similarity. Since this requires computing the similarities between mM entity pairs and there are $|G_m| = \binom{M}{m} m!$ possible one-to-one maps, the complexity of this implementation is $O(Mm + \binom{M}{m} m!)$. This is not satisfactory even for a document with a moderate number of entities: it will have about 3.6 million operations for $M = m = 10$, a document with only 10 reference and 10 system entities.

Fortunately, the entity alignment problem under the constraint that an entity can be aligned at most once is the classical maximum bipartite matching problem and there exists an algorithm (Kuhn, 1955; Munkres, 1957) (henceforth Kuhn-Munkres Algorithm) that can find the optimal solution in polynomial time. Casting the entity alignment problem as the maximum bipartite matching is trivial: each entity in $\mathcal{R}$ and $\mathcal{S}$ is a vertex and the node pair (R, S), where $R \in \mathcal{R}$, $S \in \mathcal{S}$, is connected by an edge with the weight $\phi(R, S)$. Thus the problem (1) is exactly the maximum bipartite matching.

With the Kuhn-Munkres algorithm, the procedure to compute the F-measure (5) can be described as Algorithm 1.

Algorithm 1 Computing the F-measure (5).

Input: reference entities:$\mathcal{R}$; system entities: $\mathcal{S}$
Output: optimal alignment g^*; F-measure (5).
1:Initialize: $g^* = \emptyset$; $\Phi(g^*) = 0$.
2:**For** $i = 1$ **to** $|\mathcal{R}|$
3: **For** $j = 1$ **to** $|\mathcal{S}|$
4: **Compute** $\phi(R_i, S_j)$.
5:$[g^*, \Phi(g^*)]$=**KM**$\big(\{\phi(R, S) : R \in \mathcal{R}, S \in \mathcal{S}\}\big)$.
6:$\Phi(\mathcal{R}) = \sum_{R \in \mathcal{R}} \phi(R, R)$; $\Phi(\mathcal{S}) = \sum_{S \in \mathcal{S}} \phi(S, S)$.
7:$r = \frac{\Phi(g^*)}{\Phi(\mathcal{R})}$; $p = \frac{\Phi(g^*)}{\Phi(\mathcal{S})}$; $F = \frac{2pr}{p+r}$.
8:**return** g^* and F.

The input to the algorithm are reference entities $\mathcal{R}$ and system entities $\mathcal{S}$. The algorithm returns the best one-to-

one map g^* and F-measure in equation (5). Loop from line 2 to 4 computes the similarity between all the possible reference and system entity pairs. The complexity of this loop is $O(Mm)$. Line 5 calls the Kuhn-Munkres algorithm, which takes as input the entity-pair scores $\{\phi(R,S)\}$ and outputs the best map g^* and the corresponding total similarity $\Phi(g^*)$. The worst case (i.e., when all entries in $\{\phi(R,S)\}$ are non-zeros) complexity of the Kuhn-Algorithm is $O(Mm^2 \log m)$. Line 6 computes "self-similarity" $\Phi(\mathcal{R})$ and $\Phi(\mathcal{S})$ needed in the F-measure computation at Line 7.

The core of the F-measure computation is the Kuhn-Munkres algorithm at line 5. The algorithm is initially discovered by Kuhn (1955) and Munkres (1957) to solve the matching (a.k.a assignment) problem for square matrices. Since then, it has been extended to rectangular matrices (Bourgeois and Lassalle, 1971) and parallelized (Balas et al., 1991). A recent review can be found in (Gupta and Ying, 1999), which also details the techniques of fast implementation. A short description of the algorithm is included in Appendix for the sake of completeness.

2.2 Entity Similarity Metric

In this section we consider the entity similarity metric $\phi(R,S)$ defined on an entity pair (R,S). It is desirable that $\phi(R,S)$ is large when R and S are "close" and small when R and S are very different. Some straight-forward choices could be

$$\phi_1(R,S) = \begin{cases} 1, & \text{if } R = S \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

$$\phi_2(R,S) = \begin{cases} 1, & \text{if } R \cap S \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

(6) insists that two entity are the same if all the mentions are the same, while (7) goes to the other extreme: two entities are the same if they share at least one common mention.

(6) does not offer a good granularity of similarity: For example, if $R = \{a,b,c\}$, and one system response is $S_1 = \{a,b\}$, and the other system response $S_2 = \{a\}$, then clearly S_1 is more similar to R than S_2, yet $\phi(R,S_1) = \phi(R,S_2) = 0$. For the same reason, (7) lacks of the desired discriminativity as well.

From the above argument, it is clear that we want to have a metric that can measure the degree to which two entities are similar, not a binary decision. One natural choice is measuring how many common mentions two entities share, and this can be measured by the absolute number or relative number:

$$\phi_3(R,S) = |R \cap S| \quad (8)$$

$$\phi_4(R,S) = \frac{2|R \cap S|}{|R| + |S|}. \quad (9)$$

Metric (8) simply counts the number of common mentions shared by R and S, while (9) is the mention F-measure between R and S, a relative number measuring how similar R and S are. For the abovementioned example,

$$\phi_3(R,S_1) = \phi_3(\{a,b,c\},\{a,b\}) = 2$$
$$\phi_3(R,S_2) = \phi_3(\{a,b,c\},\{a\}) = 1$$
$$\phi_4(R,S_1) = \phi_4(\{a,b,c\},\{a,b\}) = 0.8$$
$$\phi_4(R,S_2) = \phi_4(\{a,b,c\},\{a\}) = 0.5,$$

thus both metrics give the desired ranking $\phi_3(R,S_1) > \phi_3(R,S_2)$, $\phi_4(R,S_1) > \phi_4(R,S_2)$.

If $\phi_3(\cdot,\cdot)$ is adopted in Algorithm 1, $\Phi(g^*)$ is the number of total common mentions corresponding to the best one-to-one map g^* while the denominators of (3) and (4) are the number of proposed mentions and the number of system mentions, respectively. The F-measure in (5) can be interpreted as the ratio of mentions that are in the "right" entities. Similarly, if $\phi_4(\cdot,\cdot)$ is adopted in Algorithm 1, the denominators of (3) and (4) are the number of proposed entities and the number of system entities, respectively, and the F-measure in (5) can be understood as the ratio of correct entities. Therefore, (5) is called mention-based CEAF and entity-based CEAF when (8) and (9) are used, respectively.

$\phi_3(\cdot,\cdot)$ and $\phi_4(\cdot,\cdot)$ are two reasonable entity similarity measures, but by no means the only choices. At mention level, partial credit could be assigned to two mentions with different but overlapping spans; or when mention type is available, weights defined on the type confusion matrix can be incorporated. At entity level, entity attributes, if avaiable, can be weighted in the similarity measure as well. For example, ACE data defines three entity classes: NAME, NOMINAL and PRONOUN. Different weights can be assigned to the three classes.

No matter what entity similarity measure is used, it is crucial to have the constraint that the document-level similarity between reference entities and system entities is calculated over the best one-to-one map. We will see examples in Section 3 that misleading results could be produced without the alignment constraint.

Another observation is that the same evaluation paradigm can be used in any scenario that needs to measure the "closeness" between a set of system and reference objects, provided that a similarity between two objects is defined. For example, the 2004 ACE tasks include detecting and recognizing relations in text documents. A relation instance can be treated as an object and the same evaluation paradigm can be applied.

3 Comparison with Other Metrics

In this section, we compare the proposed F-measure with the MUC link-based F-measure (and its variation B-cube F-measure) and the more recent ACE-value. The

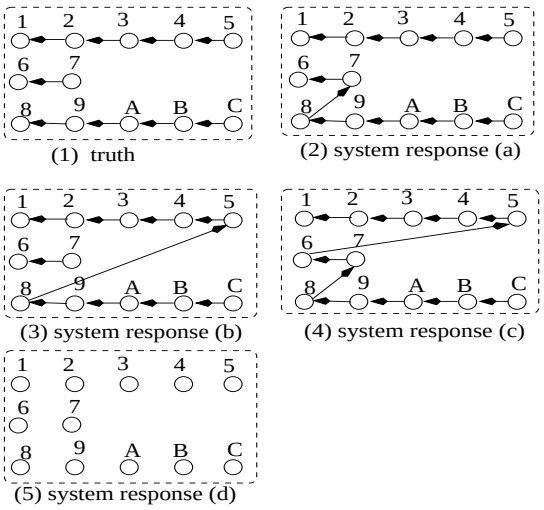

Figure 1: Example entities: (1)truth; (2)system response (a); (3)system response (b); (4)system response (c); (5)system response (d)

proposed metric has fixed problems associated with the MUC and B-cube F-measure, and has better interpretability than the ACE-value.

3.1 Comparison with the MUC F-measure and B-cube Metric on Artificial Data

We use the example in Figure 1 to compare the MUC link-based F-measure, B-cube, and the proposed mention- and entity-based CEAF. In Figure 1, mentions are represented in circles and mentions in an entity are connected by arrows. Intuitively, if each mention is treated equally, the system response (a) is better than the system response (b) since the latter mixes two big entities, $\{1,2,3,4,5\}$ and $\{8,9,A,B,C\}$, while the former mixes a small entity $\{6,7\}$ with one big entity $\{8,9,A,B,C\}$. System response (b) is clearly better than system response (c) since the latter puts all the mentions into a single entity while (b) has correctly separated the entity $\{6,7\}$ from the rest. The system response (d) is the worst: the system does not link any mentions and outputs 12 single-mention entities.

Table 1 summarizes various F-measures for system response (a) to (d): the first column contains the indices of the system responses found in Figure 1; the second and third columns are the MUC F-measure and B-cubic F-measure respectively; the last two columns are the proposed CEAF F-measures, using the entity similarity metric $\phi_3(\cdot,\cdot)$ and $\phi_4(\cdot,\cdot)$, respectively.

As shown in Table 1, the MUC link-based F-measure fails to distinguish the system response (a) and the system response (b) as the two are assigned the same F-measure. The system response (c) represents a trivial output: all mentions are put in the same entity. Yet the MUC metric will lead to a 100% recall (9 out of 9 reference links are

System response	MUC	B-cube	CEAF	
			$\phi_3(\cdot,\cdot)$	$\phi_4(\cdot,\cdot)$
(a)	0.947	0.865	0.833	0.733
(b)	0.947	0.737	0.583	0.667
(c)	0.900	0.545	0.417	0.294
(d)	–	0.400	0.250	0.178

Table 1: Comparison of coreference evaluation metrics

correct) and a 81.2% precision (9 out of 11 system links are correct), which gives rise to a 90% F-measure. It is striking that a "bad" system response gets such a high F-measure. Another problem with the MUC link-based metric is that it is not able to handle single-mention entities, as there is no link for a single mention entity. That is why the entry for system response (d) in Table 1 is empty.

B-cube F-measure ranks the four system responses in Table 1 as desired. This is because B-cube metric (Bagga and Baldwin, 1998) is computed based on mentions (as opposed to links in the MUC F-measure). But B-cube uses the same entity "intersecting" procedure found in computing the MUC F-measure (Vilain et al., 1995), and it sometimes can give counter-intuitive results. To see this, let us take a look at recall and precision for system response (c) and (d) for B-cube metric. Notice that all the reference entities are found after intersecting with the system responsce (c): $\{\{1,2,3,4,5\},\{6,7\},\{8,9,A,B,C\}\}$. Therefore, B-cube recall is 100% (the corresponding precision is $\frac{1}{12} * (10 * \frac{5}{12} + 2 * \frac{2}{12}) = 0.375$). This is counter-intuitive because the set of reference entities is not a subset of the proposed entities, thus the system response should not have gotten a 100% recall. The same problem exists for the system response (d): it gets a 100% B-cube precision (the corresponding B-cube recall is $\frac{1}{12}(5 * \frac{1}{5} + 2 * \frac{1}{2} + 5 * \frac{1}{5}) = 0.25$), but clearly not all the entities in the system response (d) are correct! These numebrs are summarized in Table 2, where columns with R and P represent recall and precision, respectively.

System response	B-cube		CEAF			
	R	P	ϕ_3-R	ϕ_3-P	ϕ_4-R	ϕ_4-P
(c)	1.0	0.375	0.417	0.417	0.196	0.588
(d)	0.25	1.0	0.250	0.250	0.444	0.111

Table 2: Example of counter-intuitive B-cube recall or precision: system repsonse (c) gets 100% recall (column R) while system repsonse (d) gets 100% precision (column P). The problem is fixed in both CEAF metrics.

The counter-intuitive results associated with the MUC and B-cube F-measures are rooted in the procedure of "intersecting" the reference and system entities, which allows an entity to be used more than once! We will come back to this after discussing the CEAF numbers.

From Table 1, we see that both mention-based (col-

umn under $\phi_3(\cdot,\cdot)$) CEAF and entity-based ($\phi_4(\cdot,\cdot)$) CEAF are able to rank the four systems properly: system (a) to (d) are increasingly worse. To see how the CEAF numbers are computed, let us take the system response (a) as an example: first, the best one-one entity map is determined. In this case, the best map is: the reference entity $\{1,2,3,4,5\}$ is aligned to the system entity $\{1,2,3,4,5\}$, the reference entity $\{8,9,A,B,C\}$ is aligned to the system $\{6,7,8,9,A,B,C\}$ and the reference entity $\{6,7\}$ is unaligned. The number of common mentions is therefore 10 which results in a mention-based ($\phi_3(\cdot,\cdot)$) recall $\frac{5}{6}$ and precision $\frac{5}{6}$. Since $\phi_4(\{1,2,3,4,5\},\{1,2,3,4,5\}) = 1$, and $\phi_4(\{8,9,A,B,C\},\{6,7,8,9,A,B,C\}) = \frac{10}{12}$, $\Phi(g^*) = 1+\frac{10}{12}$ (c.f. equation (4) and (3)), and the entity-based F-measure (c.f. equation (9)) is therefore

$$\frac{2*(1+\frac{10}{12})}{3+2} = \frac{11}{15} = 0.733.$$

CEAF for other system responses are computed similarly.

CEAF recall and precision breakdown for system (c) and (d) are listed in column 4 through 7 of Table 1. As can be seen, neither mention-based nor entity-based CEAF has the abovementioned problem associated with the B-cube metric, and the recall and precision numbers are more or less compatible with our intuition: for instance, for system (c), based on ϕ_3-CEAF number, we can say that about 41.7% mentions are in the right entity, and based on the ϕ_4-CEAF recall and precision, we can state that about 19.6% of "true" entities are recovered (recall) and about 58.8% of the proposed entities are correct.

A comparison of the procedures of computing the MUC F-measure/B-cube and CEAF reveals that the crucial difference is that the MUC and B-cube F-measure allow an entity to be used multiple times while CEAF insists that entity map be one-to-one. So an entity will never get double credit. Take the system repsonse (c) as an example, intersecting three reference entity in turn with the reference entities produces the same set of reference entities, which leads to 100% recall. In the intersection step, the system entity is effectively used three times. In contrast, the system entity is aligned to only one reference entity when computing CEAF.

3.2 Comparisons On Real Data

3.2.1 MUC F-measure and CEAF

We have seen the different behaviors of the MUC F-measure, B-cube F-measure and CEAF on the artificial data. We now compare the MUC F-measure, CEAF, and ACE-value metrics on real data (compasion between the MUC and B-cube F-measure can be found in (Bagga and Baldwin, 1998)). Comparsion between the MUC F-measure and CEAF is done on the MUC6 coreference test set, while comparison between the CEAF and ACE-value is done on the 2004 ACE data. The setup reflects the fact that the official MUC scorer and ACE scorer run on their own data format and are not easily portable to the other data set. All the experiments in this section are done on true mentions.

Penalty	#sys-ent	MUC-F	ϕ_3-CEAF
-0.6	561	.851	0.750
-0.8	538	.854	0.756
-0.9	529	.853	0.753
-1	515	.853	0.753
-1.1	506	.856	0.764
-1.2	483	.857	**0.768**
-1.4	448	.863	0.761
-1.5	425	.862	0.749
-1.6	411	.864	0.740
-1.7	403	.865	0.741
-10	113	**.902**	0.445

Table 3: MUC F-measure and mention-based CEAF on the official MUC6 test set. The first column contains the penalty value in decreasing order. The second column contains the number of system-proposed entities. The column under MUC-F is the MUC F-measure while ϕ_3-CEAF is the mention-based CEAF.

The coreference system is similar to the one used in (Luo et al., 2004). Results in Table 3 are produced by a system trained on the MUC6 training data and tested on the 30 official MUC6 test documents. The test set contains 460 reference entities. The coreference system uses a penalty parameter to balance miss and false alarm errors: the smaller the parameter, the fewer entities will be generated. We vary the parameter from -0.6 to -10, listed in the first column of Table 3, and compare the system performance measured by the MUC F-measure and the proposed mention-based CEAF.

As can be seen, the mention-based CEAF has a clear maximum when the number of proposed entities is close to the truth: at the penlaty value -1.2, the system produces 483 entities, very close to 460, and the ϕ_3-CEAF achieves the maximum 0.768. In contrast, the MUC F-measure increases almost monotonically as the system proposes fewer and fewer entities. In fact, the best system according to the MUC F-measure is the one proposing only 113 entities. This demonstrates a fundamental flaw of the MUC F-measure: the metric intrinsically favors a system producing fewer entities and therefore lacks of discriminativity.

3.2.2 ACE-Value and CEAF

Now let us turn to ACE-value. Results in Table 4 are produced by a system trained on the ACE 2002 and 2004 training data and tested on a separate test set, which contains 853 reference entities. Both ACE-value and the mention-based CEAF penalizes systems over-producing or under-producing entities: ACE-value is maximum

Penalty	#sys-ent	ACE-value(%)	ϕ_3-CEAF
0.6	1221	88.5	0.726
0.4	1172	89.1	0.749
0.2	1145	89.4	0.755
0	1105	89.7	0.766
-0.2	1050	**89.7**	0.775
-0.4	1015	89.7	0.780
-0.6	990	89.5	0.782
-0.8	930	88.6	**0.794**
-1	891	86.9	0.780
-1.2	865	86.7	0.778
-1.4	834	85.6	0.769
-1.6	790	83.8	0.761

Table 4: Comparison of ACE-value and mention-based CEAF. The first column contains the penalty value in decreasing order. The second column contains the number of system-proposed entities. ACE-values are in percentage. The number of reference entities is 853.

when the penalty value is -0.2 and CEAF is maximum when the penalty value is -0.8. However, the optimal CEAF system produces 930 entities while the optimal ACE-value system produces 1050 entities. Judging from the number of entities, the optimal CEAF system is closer to the "truth" than the counterpart of ACE-value. This is not very surprising since ACE-value is a weighted metric while CEAF treats each mention and entity equally. As such, the two metrics have very weak correlation.

While we can make a statement such as "the system with penalty -0.8 puts about 79.4% mentions in right entities", it is hard to interpret the ACE-value numbers.

Another difference is that CEAF is symmetric[1], but ACE-Value is not. Symmetry is a desirable property. For example, when comparing inter-annotator agreement, a symmetric metric is independent of the order of two sets of input documents, while an asymmetric metric such as ACE-Value needs to state the input order along with the metric value.

4 Conclusions

A coreference performance metric – CEAF – is proposed in this paper. The CEAF metric is computed based on the best one-to-one map between reference entities and system entities. Finding the best one-to-one map is a maximum bipartite matching problem and can be solved by the Kuhn-Munkres algorithm. Two example entity-pair similarity measures (i.e., $\phi_3(\cdot,\cdot)$ and $\phi_4(\cdot,\cdot)$) are proposed, resulting one mention-based CEAF and one entity-based CEAF, respectively. It has been shown that the proposed CEAF metric has fixed problems associated with the MUC link-based F-measure and B-cube F-measure. The proposed metric also has better interpretability than ACE-value.

[1]This was pointed out by Nanda Kambhatla.

Acknowledgments

This work was partially supported by the Defense Advanced Research Projects Agency and monitored by SPAWAR under contract No. N66001-99-2-8916. The views and findings contained in this material are those of the authors and do not necessarily reflect the position of policy of the Government and no official endorsement should be inferred.

The author would like to thank three reviewers and my colleagues, Hongyan Jing and Salim Roukos, for suggestions of improving the paper.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC'98)*, pages 563–566.

Egon Balas, Donald Miller, Joseph Pekny, and Paolo Toth. 1991. A parallel shortest augmenting path algorithm for the assignment problem. *Journal of the ACM (JACM)*, 38(4).

Francois Bourgeois and Jean-Claude Lassalle. 1971. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12).

R. Fletcher. 1987. *Practical Methods of Optimization*. John Wiley and Sons.

Anshul Gupta and Lexing Ying. 1999. Algorithms for finding maximum matchings in bipartite graphs. Technical Report RC 21576 (97320), IBM T.J. Watson Research Center, October.

H.W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(83).

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*.

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference(MUC-6)*, San Francisco, CA. Morgan Kaufmann.

MUC-7. 1998. *Proceedings of the Seventh Message Understanding Conference(MUC-7)*.

J. Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of SIAM*, 5:32–38.

NIST. 2003a. The ACE evaluation plan. www.nist.gov/speech/tests/ace/index.htm.

NIST. 2003b. Proceedings of ACE'03 workshop. Booklet, Alexandria, VA, September.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, , and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *In Proc. of MUC6*, pages 45–52.

Appendix: Kuhn-Munkres Algorithm

Let i index the reference entities $\mathcal{R}$ and j index the system entities $\mathcal{S}$, and $\phi(i,j)$ be the similarity between the i^{th} reference entity and the j^{th} system entity. Algebraically, the maximum bipartite matching can be stated as an integer programming problem:

$$\max_{\{x_{ij}\}} \phi(i,j)x_{ij} \tag{10}$$

$$\text{subject to: } \sum_j x_{ij} \leq 1, \forall i \tag{11}$$

$$\sum_i x_{ij} \leq 1, \forall j \tag{12}$$

$$x_{ij} \in \{0,1\}, \forall i,j. \tag{13}$$

If $x_{ij} = 1$, the i^{th} reference entity and the j^{th} system entity are aligned. Constraint (11) (or (12)) implies that a reference (or system) entity cannot be aligned more than once with a system (or reference) entity.

Observe that the coefficients of (11) and (12) are unimodular. Thus, Constraint (13) can be replaced by

$$x_{ij} \geq 0, \forall i,j. \tag{14}$$

The dual (cf. pp. 219 of (Fletcher, 1987)) to the optimization problem (10) with constraints (11),(12) and (14) is:

$$\min_{\{u_i\},\{v_j\}} \sum_i u_i + \sum_j v_j \tag{15}$$

$$s.t.: \ u_i + v_j \geq \phi(i,j), \forall i,j \tag{16}$$

$$u_i \geq 0, \forall i \tag{17}$$

$$v_j \geq 0, \forall j. \tag{18}$$

The dual has the same optimal objective value as the primal.

It can be shown that the optimal conditions for the dual problem (and hence the maximum similarity match) are:

$$u_i + v_j = \phi(i,j), \text{if } (i,j) \text{ is aligned} \tag{19}$$

$$u_i = 0, \text{if } i \text{ is free (i.e., not aligned)} \tag{20}$$

$$v_j = 0, \text{if } j \text{ is free.} \tag{21}$$

The Kuhn-Munkres algorithm starts with an empty match and an initial feasible set of $\{u_i\}$ and $\{v_j\}$, and iteratively increases the cardinality of the match while satisfying the optimal conditions (19)-(21). Notice that conceptually, a matching problem with a rectangular matrix $[\phi(i,j)]$ can always reduce to a square one by padding zeros (this is not necessary in practice, see, for instance (Bourgeois and Lassalle, 1971)). For this reason, we state the Kuhn-Munkres algorithm for the case where $|\mathcal{R}| = |\mathcal{S}|$ (or $M = m$) in Algorithm 2. The proof of correctness is omitted due to space limit.

Note that $P_{aug}(i,j)$ on line 9 stands for the augmenting (i.e., a free node followed by an aligned node, followed by a free node, ...) path from i to j in the corresponding bipartite graph. $A \oplus P_{aug}(i,j)$ is understood as edge "exclusive-or:" if an edge (k,l) is in A and on the path $P_{aug}(i,j)$, it will be removed from A; if the edge is in either A or $P_{aug}(i,j)$, it will be added.

Algorithm 2 Kuhn-Munkres Algorithm

Input: similarity matrix: $[\phi(i,j)]$
Output: best match $A = \{(i,j)\}$ and similarity Φ.
1:Initialize: $\forall i$, $u_i = \max_j \phi(i,j)$; $\forall j$, $v_j = 0$; $A = \emptyset$.
2:**For** $i = 1$ to M
3: **If** i is not free, Continue; **EndIF**.
4: $X = \{i\}$, $Y = \emptyset$;
5: **While** true
6: $N(X) = \{l : \exists k \in X, s.t. \phi(k,l) = u_k + v_l\}$
7: **If** $Y \subset N(X)$
8: pick $j \in N(X) \setminus Y$
9: **If** j is free
10: $A = A \oplus P_{aug}(i,j)$; break
11: **Else**
12: Find i' such that $(i',j) \in A$.
13: $X = X \cup \{i'\}, Y = Y \cup \{j\}$.
14: Goto line 6.
15: **EndIf**
16: **Else** $Y == N(X)$
17: $\delta = \min_{k \in X, l \in \bar{Y}} \{u_k + v_l - \phi(k,l)\}$
18: $(\bar{i}, \bar{j}) = arg\min_{k \in X, l \in \bar{Y}} \{u_k + v_l - \phi(k,l)\}$
19: $u_k = u_k - \delta$ for $k \in X$.
20: $v_l = v_l + \delta$ for $l \in Y$.
21: $j = \bar{j}$. Goto line 9.
22: **EndIf**
23: **EndWhile**
24:**EndFor**
25:$\Phi = \sum_{(k,l) \in A} \phi(k,l)$.
26:Return A and Φ.