Evaluation Metrics For End-to-End Coreference Resolution Systems

Jie Cai and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
Schloß-Wolfsbrunnenweg 35
69118 Heidelberg, Germany
(jie.cai|michael.strube)@h-its.org

Abstract

Commonly used coreference resolution evaluation metrics can only be applied to key mentions, i.e. already annotated mentions. We here propose two variants of the *B*^{3} and *CEAF* coreference resolution evaluation algorithms which can be applied to coreference resolution systems dealing with system mentions, i.e. automatically determined mentions. Our experiments show that our variants lead to intuitive and reliable results.

1 Introduction

The coreference resolution problem can be divided into two steps: (1) determining *mentions*, i.e., whether an expression is referential and can take part in a coreferential relationship, and (2) deciding whether mentions are coreferent or not. Most recent research on coreference resolution simplifies the resolution task by providing the system with *key mentions*, i.e. already annotated mentions (Luo et al. (2004), Denis & Baldridge (2007), Culotta et al. (2007), Haghighi & Klein (2007), inter alia; see also the task description of the recent SemEval task on coreference resolution at `http://stel.ub.edu/semeval2010-coref`), or ignores an important part of the problem by evaluating on key mentions only (Ponzetto & Strube, 2006; Bengtson & Roth, 2008, inter alia). We follow here Stoyanov et al. (2009, p.657) in arguing that such evaluations are "an unrealistic surrogate for the original problem" and ask researchers to evaluate end-to-end coreference resolution systems.

However, the evaluation of end-to-end coreference resolution systems has been inconsistent making it impossible to compare the results. Nicolae & Nicolae (2006) evaluate using the *MUC* score (Vilain et al., 1995) and the *CEAF* algorithm (Luo, 2005) without modifications. Yang et al. (2008) use only the *MUC* score. Bengtson & Roth (2008) and Stoyanov et al. (2009) derive variants from the *B*^{3} algorithm (Bagga & Baldwin, 1998). Rahman & Ng (2009) propose their own variants of *B*^{3} and *CEAF*. Unfortunately, some of the metrics' descriptions are so concise that they leave too much room for interpretation. Also, some of the metrics proposed are too lenient or are more sensitive to mention detection than to coreference resolution. Hence, though standard corpora are used, the results are not comparable.

This paper attempts to fill that desideratum by analysing several variants of the *B*^{3} and *CEAF* algorithms. We propose two new variants, namely B^3_{sys} and $CEAF_{sys}$, and provide algorithmic details in Section 2. We describe two experiments in Section 3 showing that B^3_{sys} and $CEAF_{sys}$ lead to intuitive and reliable results. Implementations of B^3_{sys} and $CEAF_{sys}$ are available open source along with extended examples[1].

2 Coreference Evaluation Metrics

We discuss the problems which arise when applying the most prevalent coreference resolution evaluation metrics to end-to-end systems and propose our variants which overcome those problems. We provide detailed analyses of illustrative examples.

2.1 *MUC*

The MUC score (Vilain et al., 1995) counts the minimum number of links between mentions to be inserted or deleted when mapping a system response to a gold standard key set. Although pairwise links capture the information in a set, they cannot represent singleton entities, i.e. entities, which are mentioned only once. Therefore, the MUC score is not suitable for the ACE data (`http://www.itl.nist.`

[1] `http://www.h-its.org/nlp/download`

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 28–36,
The University of Tokyo, September 24-25, 2010. ©2010 Association for Computational Linguistics

gov/iad/mig/tests/ace/), which includes singleton entities in the keys. Moreover, the MUC score does not give credit for separating singleton entities from other chains. This becomes problematic in a realistic system setup, when mentions are extracted automatically.

2.2 B^3

The B^3 algorithm (Bagga & Baldwin, 1998) overcomes the shortcomings of the MUC score. Instead of looking at the links, B^3 computes precision and recall for all mentions in the document, which are then combined to produce the final precision and recall numbers for the entire output.

For each mention, the B^3 algorithm computes a precision and recall score using equations 1 and 2:

$$Precision(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|R_{m_i}|} \quad (1)$$

$$Recall(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|K_{m_i}|} \quad (2)$$

where R_{m_i} is the response chain (i.e. the system output) which includes the mention m_i, and K_{m_i} is the key chain (manually annotated gold standard) with m_i. The overall precision and recall are computed by averaging them over all mentions.

Since B^3's calculations are based on mentions, singletons are taken into account. However, a problematic issue arises when system mentions have to be dealt with: B^3 assumes the mentions in the key and in the response to be identical. Hence, B^3 has to be extended to deal with system mentions which are not in the key and key mentions not extracted by the system, so called *twinless mentions* (Stoyanov et al., 2009).

2.2.1 Existing B^3 variants

A few variants of the B^3 algorithm for dealing with system mentions have been introduced recently. Stoyanov et al. (2009) suggest two variants of the B^3 algorithm to deal with system mentions, B^3_0 and B^3_{all}[2]. For example, a key and a response are provided as below:

Key : {a b c}
Response: {a b d}

B^3_0 discards all twinless system mentions (i.e. mention d) and penalizes recall by setting $recall_{m_i} = 0$ for all twinless key mentions (i.e. mention c). The B^3_0 precision, recall and F-score (i.e. $F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$) for the example are calculated as:

$$Pr_{B^3_0} = \tfrac{1}{2}(\tfrac{2}{2} + \tfrac{2}{2}) = 1.0$$
$$Rec_{B^3_0} = \tfrac{1}{3}(\tfrac{2}{3} + \tfrac{2}{3} + 0) \doteq 0.444$$
$$F_{B^3_0} = 2 \times \tfrac{1.0 \times 0.444}{1.0 + 0.444} \doteq 0.615$$

		Set 1		
System 1	key response	{a b c} {a b d}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key response	{a b c} {a b d e}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.375	0.556	0.448
$B^3_{r\&n}$		0.375	0.556	0.448
B^3_{sys}		0.5	0.556	0.527
$CEAF_{sys}$		0.4	0.667	0.500

Table 1: Problems of B^3_0

B^3_{all} retains twinless system mentions. It assigns $1/|R_{m_i}|$ to a twinless system mention as its precision and similarly $1/|K_{m_i}|$ to a twinless key mention as its recall. For the same example above, the B^3_{all} precision, recall and F-score are given by:

$$Pr_{B^3_{all}} = \tfrac{1}{3}(\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{1}{3}) \doteq 0.556$$
$$Rec_{B^3_{all}} = \tfrac{1}{3}(\tfrac{2}{3} + \tfrac{2}{3} + \tfrac{1}{3}) \doteq 0.556$$
$$F_{B^3_{all}} = 2 \times \tfrac{0.556 \times 0.556}{0.556 + 0.444} \doteq 0.556$$

Tables 1, 2 and 3 illustrate the problems with B^3_0 and B^3_{all}. The rows labeled *System* give the original keys and system responses while the rows labeled B^3_0, B^3_{all} and B^3_{sys} show the performance generated by Stoyanov et al.'s variants and the one we introduce in this paper, B^3_{sys} (the row labeled $CEAF_{sys}$ is discussed in Subsection 2.3).

In Table 1, there are two system outputs (i.e. *System 1* and *System 2*). Mentions *d* and *e* are the twinless system mentions erroneously resolved and *c* a twinless key mention. *System 1* is supposed to be slightly better with respect to precision, because *System 2* produces one more spurious resolution (i.e. for mention *e*). However, B^3_0 computes exactly the same numbers for both systems. Hence, there is no penalty for erroneous coreference relations in B^3_0, if the mentions do not appear in the key, e.g. putting mentions *d* or *e* in *Set 1* does not count as precision errors. — B^3_0 is too lenient by only evaluating the correctly extacted mentions.

[2] Our discussion of B^3_0 and B^3_{all} is based on the analysis of the source code available at http://www.cs.utah.edu/nlp/reconcile/.

		Set 1	Singletons	
System 1	key response	{a b c} {a b d}		
		P	R	F
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key response	{a b c} {a b d}	 {c}	
		P	R	F
B^3_{all}		0.667	0.556	0.606
$B^3_{r\&n}$		0.667	0.556	0.606
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572

Table 2: Problems of B^3_{all} (1)

		Set 1	Singletons	
System 1	key response	{a b} {a b d}		
		P	R	F
B^3_{all}		0.556	1.0	0.715
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800
System 2	key response	{a b} {a b d}	 {i} {j} {k}	
		P	R	F
B^3_{all}		0.778	1.0	0.875
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800

Table 3: Problems of B^3_{all} (2)

B^3_{all} deals well with the problem illustrated in Table 1, the figures reported correspond to intuition. However, B^3_{all} can output different results for identical coreference resolutions when exposed to different mention taggers as shown in Tables 2 and 3. B^3_{all} manages to penalize erroneous resolutions for twinless system mentions, however, it ignores twinless key mentions when measuring precision. In Table 2, *System 1* and *System 2* generate the same outputs, except that the mention tagger in *System 2* also extracts mention *c*. Intuitively, the same numbers are expected for both systems. However, B^3_{all} gives a higher precision to *System 2,* which results in a higher F-score.

B^3_{all} retains all twinless system mentions, as can be seen in Table 3. *System 2*'s mention tagger tags more mentions (i.e. the mentions *i*, *j* and *k*), while both *System 1* and *System 2* have identical coreference resolution performance. Still, B^3_{all} outputs quite different results for precision and thus for F-score. This is due to the credit B^3_{all} takes from unresolved singleton twinless system mentions (i.e. mention *i*, *j*, *k* in *System 2*). Since the metric is expected to evaluate the end-to-end coreference system performance rather than the mention tagging quality, it is not satisfying to observe that B^3_{all}'s numbers actually fluctuate when the system is exposed to different mention taggers.

Rahman & Ng (2009) apply another variant, denoted here as $B^3_{r\&n}$. They remove only those twinless system mentions that are singletons before applying the B^3 algorithm. So, a system would not be rewarded by the the spurious mentions which are correctly identified as singletons during resolution (as has been the case with B^3_{all}'s higher precision for *System 2* as can be seen in Table 3).

We assume that Rahman & Ng apply a strategy similar to B^3_{all} after the removing step (this is not clear in Rahman & Ng (2009)). While it avoids the problem with singleton twinless system mentions, $B^3_{r\&n}$ still suffers from the problem dealing with twinless key mentions, as illustrated in Table 2.

2.2.2 B^3_{sys}

We here propose a coreference resolution evaluation metric, B^3_{sys}, which deals with system mentions more adequately (see the rows labeled B^3_{sys} in Tables 1, 2, 3, 4 and 5). We put all twinless key mentions into the response as singletons which enables B^3_{sys} to penalize non-resolved coreferent key mentions without penalizing non-resolved singleton key mentions, and also avoids the problem B^3_{all} and $B^3_{r\&n}$ have as shown in Table 2. All twinless system mentions which were deemed not coreferent (hence being singletons) are discarded. To calculate B^3_{sys} precision, all twinless system mentions which were mistakenly resolved are put into the key since they are spurious resolutions (equivalent to the assignment operations in B^3_{all}), which should be penalized by precision. Unlike B^3_{all}, B^3_{sys} does not benefit from unresolved twinless system mentions (i.e. the twinless singleton system mentions). For recall, the algorithm only goes through the original key sets, similar to B^3_{all} and $B^3_{r\&n}$. Details are given in Algorithm 1.

For example, a coreference resolution system has the following key and response:

Key : {a b c}
Response: {a b d} {i j}

To calculate the precision of B^3_{sys}, the key and response are altered to:

Key$_p$: {a b c} {d} {i} {j}
Response$_p$: {a b d} {i j} {c}

Algorithm 1 B^3_{sys}

```
Input: key sets $key$, response sets $response$
Output: precision $P$, recall $R$ and F-score $F$
Discard all the singleton twinless system mentions in $response$;
Put all the twinless annotated mentions into $response$;
if calculating precision then
    Merge all the remaining twinless system mentions with $key$ to form $key_p$;
    Use $response$ to form $response_p$
    Through $key_p$ and $response_p$;
    Calculate $B^3$ precision $P$.
end if
if calculating recall then
    Discard all the remaining twinless system mentions in $response$ to from $response_r$;
    Use $key$ to form $key_r$
    Through $key_r$ and $response_r$;
    Calculate $B^3$ recall $R$
end if
Calculate F-score $F$
```

So, the precision of B^3_{sys} is given by:

$Pr_{B^3_{sys}} = \frac{1}{6}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{2} + 1) \doteq 0.611$

The modified key and response for recall are:

Key$_r$: {a b c}
Response$_r$: {a b} {c}

The resulting recall of B^3_{sys} is:

$Rec_{B^3_{sys}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$

Thus the F-score number is calculated as:

$F_{B^3_{sys}} = 2 \times \frac{0.611 \times 0.556}{0.611 + 0.556} \doteq 0.582$

B^3_{sys} indicates more adequately the performance of end-to-end coreference resolution systems. It is not easily tricked by different mention taggers[3].

2.3 *CEAF*

Luo (2005) criticizes the B^3 algorithm for using entities more than one time, because B^3 computes precision and recall of mentions by comparing entities containing that mention. Hence Luo proposes the *CEAF* algorithm which aligns entities in key and response. *CEAF* applies a similarity metric (which could be either mention based or entity based) for each pair of entities (i.e. a set of mentions) to measure the goodness of each possible alignment. The best mapping is used for calculating *CEAF* precision, recall and F-measure.

Luo proposes two entity based similarity metrics (Equation 3 and 4) for an entity pair (K_i, R_j) originating from key, K_i, and response, R_j.

$$\phi_3(K_i, R_j) = |K_i \cap R_j| \quad (3)$$

$$\phi_4(K_i, R_j) = \frac{2|K_i \cap R_j|}{|K_i| + |R_j|} \quad (4)$$

The *CEAF* precision and recall are derived from the alignment which has the best total similarity (denoted as $\Phi(g^*)$), shown in Equations 5 and 6.

$$Precision = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \quad (5)$$

$$Recall = \frac{\Phi(g^*)}{\sum_i \phi(K_i, K_i)} \quad (6)$$

If not specified otherwise, we apply Luo's $\phi_3(\star, \star)$ in the example illustrations. We denote the original *CEAF* algorithm as *CEAF*$_{orig}$.

Detailed calculations are illustrated below:

Key : {a b c}
Response: {a b d}

The *CEAF*$_{orig}$ $\phi_3(\star, \star)$ are given by:

$\phi_3(K_1, R_1) = 2 \ (K_1 : \{abc\}; R_1 : \{abd\})$
$\phi_3(K_1, K_1) = 3$
$\phi_3(R_1, R_1) = 3$

So the *CEAF*$_{orig}$ evaluation numbers are:

$Pr_{CEAF_{orig}} = \frac{2}{3} = 0.667$
$Rec_{CEAF_{orig}} = \frac{2}{3} = 0.667$
$F_{CEAF_{orig}} = 2 \times \frac{0.667 \times 0.667}{0.667 + 0.667} = 0.667$

2.3.1 Problems of *CEAF*$_{orig}$

CEAF$_{orig}$ was intended to deal with key mentions. Its adaptation to system mentions has not been addressed explicitly. Although *CEAF*$_{orig}$ theoretically does not require to have the same number of mentions in key and response, it still cannot be directly applied to end-to-end systems, because the entity alignments are based on mention mappings.

As can be seen from Table 4, *CEAF*$_{orig}$ fails to produce intuitive results for system mentions. *System 2* outputs one more spurious entity (containing mention *i* and *j*) than *System 1* does, however, achieves a same *CEAF*$_{orig}$ precision. Since twinless system mentions do not have mappings in key, they contribute nothing to the mapping similarity. So, resolution mistakes for system mentions are not calculated, and moreover, the precision is easily skewed by the number of output entities. *CEAF*$_{orig}$ reports very low precision for system mentions (see also Stoyanov et al. (2009)).

2.3.2 Existing *CEAF* variants

Rahman & Ng (2009) briefly introduce their *CEAF* variant, which is denoted as *CEAF*$_{r\&n}$ here. They use $\phi_3(\star, \star)$, which results in equal *CEAF*$_{r\&n}$ precision and recall figures when using true mentions. Since Rahman & Ng's experiments using system mentions produce unequal precision and recall figures, we assume that, after removing

[3] Further example analyses can be found in Appendix A.

		Set 1	Set 2	Singletons
System 1	key response	{a b c} {a b}		 {c} {i} {j}
		P	R	F
CEAF$_{orig}$		0.4	0.667	0.500
B^3_{sys}		1.0	0.556	0.715
CEAF$_{sys}$		0.667	0.667	0.667
System 2	key response	{a b c} {a b}	 {i j}	 {c}
		P	R	F
CEAF$_{orig}$		0.4	0.667	0.500
B^3_{sys}		0.8	0.556	0.656
CEAF$_{sys}$		0.6	0.667	0.632

Table 4: Problems of *CEAF*$_{orig}$

		Set 1	Set 2	Set 3	Singletons
System 1	key response	{a b c} {a b}	 {i j}	 {k l}	 {c}
		P	R	F	
CEAF$_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.714	0.556	0.625	
CEAF$_{sys}$		0.571	0.667	0.615	
System 2	key response	{a b c} {a b}	 {i j k l}		 {c}
		P	R	F	
CEAF$_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.571	0.556	0.563	
CEAF$_{sys}$		0.429	0.667	0.522	

Table 5: Problems of CEAF$_{r\&n}$

twinless singleton system mentions, they do not put any twinless mentions into the other set. In the example in Table 5, *CEAF*$_{r\&n}$ does not penalize adequately the incorrectly resolved entities consisting of twinless sytem mentions. So *CEAF*$_{r\&n}$ does not tell the difference between *System 1* and *System 2*. It can be concluded from the examples that the same number of mentions in key and response is needed for computing the *CEAF* score.

2.3.3 *CEAF*$_{sys}$

We propose to adjust *CEAF* in the same way as we did for B^3_{sys}, resulting in *CEAF*$_{sys}$. We put all twinless key mentions into the response as singletons. All singleton twinless system mentions are discarded. For calculating *CEAF*$_{sys}$ precision, all twinless system mentions which were mistakenly resolved are put into the key. For computing *CEAF*$_{sys}$ recall, only the original key sets are considered. That way *CEAF*$_{sys}$ deals adequately with system mentions (see Algorithm 2 for details).

```
Algorithm 2 $CEAF_{sys}$
Input: key sets $key$, response sets $response$
Output: precision $P$, recall $R$ and F-score $F$
Discard all the singleton twinless system mentions in $response$;
Put all the twinless annotated mentions into $response$;
if calculating precision then
    Merge all the remaining twinless system mentions with $key$ to form $key_p$;
    Use $response$ to form $response_p$
    Form Map $g^\star$ between $key_p$ and $response_p$
    Calculate $CEAF$ precision $P$ using $\phi_3(\star,\star)$
end if
if calculating recall then
    Discard all the remaining twinless system mentions in $response$ to form $response_r$;
    Use $key$ to form $key_r$
    Form Map $g^\star$ between $key_r$ and $response_r$
    Calculate $CEAF$ recall $R$ using $\phi_3(\star,\star)$
end if
Calculate F-score $F$
```

Taking *System 2* in Table 4 as an example, key and response are altered for precision:

Key$_p$: {a b c} {i} {j}
Response$_p$: {a b d} {i j} {c}

So the $\phi_3(\star,\star)$ are as below, only listing the best mappings:

$$\phi_3(K_1,R_1) = 2\ (K_1 : \{abc\}; R_1 : \{abd\})$$
$$\phi_3(K_2,R_2) = 1\ (K_2 : \{i\}; R_2 : \{ij\})$$
$$\phi_3(\emptyset,R_3) = 0\ (R_3 : \{c\})$$
$$\phi_3(R_1,R_1) = 3$$
$$\phi_3(R_2,R_2) = 2$$
$$\phi_3(R_3,R_3) = 1$$

The precision is thus give by:

$$Pr_{CEAF_{sys}} = \frac{2+1+0}{3+2+1} = 0.6$$

The key and response for recall are:

Key$_r$: {a b c}
Response$_r$: {a b} {c}

The resulting $\phi_3(\star,\star)$ are:

$$\phi_3(K_1,R_1) = 2(K_1 : \{abc\}; R_1 : \{ab\})$$
$$\phi_3(\emptyset,R_2) = 0(R_2 : \{c\})$$
$$\phi_3(K_1,K_1) = 3$$
$$\phi_3(R_1,R_1) = 2$$
$$\phi_3(R_2,R_2) = 1$$

The recall and F-score are thus calculated as:

$$Rec_{CEAF_{sys}} = \frac{2}{3} = 0.667$$
$$F_{CEAF_{sys}} = 2 \times \frac{0.6 \times 0.667}{0.6 + 0.667} = 0.632$$

However, one additional complication arises with regard to the similarity metrics used by *CEAF*. It turns out that only $\phi_3(\star,\star)$ is suitable for dealing with system mentions while $\phi_4(\star,\star)$ produces uninituitive results (see Table 6).

$\phi_4(\star,\star)$ computes a normalized similarity for each entity pair using the summed number of mentions in the key and the response. *CEAF* precision then distributes that similarity evenly over the response set. Spurious system entities, such as the one with mention *i* and *j* in Table 6, are not penalized. $\phi_3(\star,\star)$ calculates unnormalized similarities. It compares the two systems in Table 6 adequately. Hence we use only $\phi_3(\star,\star)$ in *CEAF*$_{sys}$.

		Set 1	Singletons	
System 1	key	{a b c}		
	response	{a b}	{c} {i} {j}	
		P	R	F
$\phi_4(\star,\star)$		0.4	0.8	0.533
$\phi_3(\star,\star)$		0.667	0.667	0.667
System 2	key	{a b c}		
	response	{a b} {i j}	{c}	
		P	R	F
$\phi_4(\star,\star)$		0.489	0.8	0.607
$\phi_3(\star,\star)$		0.6	0.667	0.632

Table 6: Problems of $\phi_4(\star,\star)$

When normalizing the similarities by the number of entities or mentions in the key (for recall) and the response (for precision), the *CEAF* algorithm considers all entities or mentions to be equally important. Hence *CEAF* tends to compute quite low precision for system mentions which does not represent the system performance adequately. Here, we do not address this issue.

2.4 *BLANC*

Recently, a new coreference resolution evaluation algorithm, *BLANC*, has been introduced (Recasens & Hovy, 2010). This measure implements the *Rand index* (Rand, 1971) which has been originally developed to evaluate clustering methods. The *BLANC* algorithm deals correctly with singleton entities and rewards correct entities according to the number of mentions. However, a basic assumption behind *BLANC* is, that the sum of all coreferential and non-coreferential links is constant for a given set of mentions. This implies that *BLANC* assumes identical mentions in key and response. It is not clear how to adapt *BLANC* to system mentions. We do not address this issue here.

3 Experiments

While Section 2 used toy examples to motivate our metrics B^3_{sys} and *CEAF*$_{sys}$, we here report results on two larger experiments using ACE2004 data.

3.1 Data and Mention Taggers

We use the ACE2004 (Mitchell et al., 2004) English training data which we split into three sets following Bengtson & Roth (2008): Train (268 docs), Dev (76), and Test (107). We use two in-house mention taggers. The first (*SM1*) implements a heuristic aiming at high recall. The second (*SM2*) uses the *J48* decision tree classifier (Witten & Frank, 2005). The number of detected mentions, head coverage, and accuracy on testing data are shown in Table 7.

		SM1	*SM2*
training	mentions	31,370	16,081
	twin mentions	13,072	14,179
development	mentions	8,045	–
	twin mentions	3,371	–
test	mentions	8,387	4,956
	twin mentions	4,242	4,212
	head coverage	79.3%	73.3%
	accuracy	57.3%	81.2%

Table 7: Mention Taggers on ACE2004 Data

3.2 Artificial Setting

For the artificial setting we report results on the development data using the *SM1* tagger. To illustrate the stability of the evaluation metrics with respect to different mention taggers, we reduce the number of twinless system mentions in intervals of 10%, while correct (non-twinless) ones are kept untouched. The coreference resolution system used is the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The results are plotted in Figures 1 and 2.

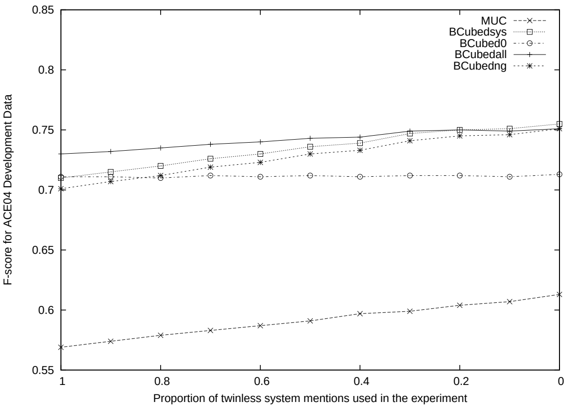

Figure 1: Artificial Setting B^3 Variants

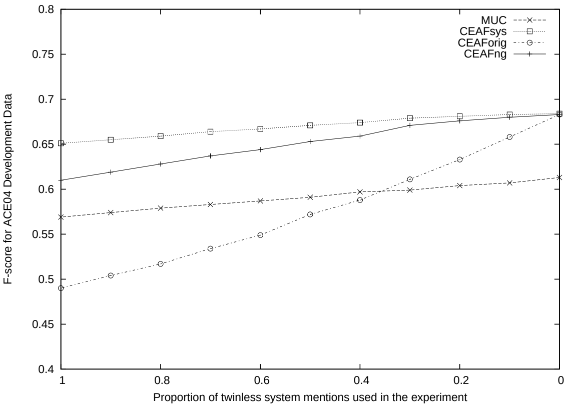

Figure 2: Artificial Setting *CEAF* Variants

	MUC		
	R	Pr	F
Soon (SM1)	51.7	53.1	52.4
Soon (SM2)	49.1	69.9	**57.7**

Table 8: Realistic Setting *MUC*

Omitting twinless system mentions from the training data while keeping the number of correct mentions constant should improve the coreference resolution performance, because a more precise coreference resolution model is obtained. As can be seen from Figures 1 and 2, the *MUC*-score, B^3_{sys} and *CEAF*$_{sys}$ follow this intuition.

B^3_0 is almost constant. It does not take twinless mentions into account. B^3_{all}'s curve, also, has a lower slope in comparison to B$^3_{sys}$ and MUC (i.e. B$^3_{all}$ computes similar numbers for worse models). This shows that the B^3_{all} score can be tricked by using a high recall mention tagger, e.g. in cases with the worse models (i.e. ones on the left side of the figures) which have much more twinless system mentions. The original *CEAF* algorithm, *CEAF*$_{orig}$, is too sensitive to the input system mentions making it less reliable. *CEAF*$_{sys}$ is parallel to B$^3_{sys}$. Thus both of our metrics exhibit the same intuition.

3.3 Realistic Setting

3.3.1 Experiment 1

For the realistic setting we compare *SM1* and *SM2* as preprocessing components for the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The coreference resolution system with the *SM2* tagger performs better, because a better coreference model is achieved from system mentions with higher accuracy.

The *MUC*, B$^3_{sys}$ and *CEAF*$_{sys}$ metrics have the same tendency when applied to systems with different mention taggers (Table 8, 9 and 10 and the bold numbers are higher with a p-value of 0.05, by a paired-t test). Since the *MUC* scorer does not evaluate singleton entities, it produces too low numbers which are not informative any more.

As shown in Table 9, B$^3_{all}$ reports counterintuitive results when a system is fed with system mentions generated by different mention taggers. B$^3_{all}$ cannot be used to evaluate two different end-to-end coreference resolution systems, because the mention tagger is likely to have bigger impact than the coreference resolution system. B$^3_0$ fails to generate the right comparison too, because it is too lenient by ignoring all twinless mentions.

	B^3_{sys}			B^3_0		
	R	Pr	F	R	Pr	F
Soon (SM2)	64.1	87.3	73.9	54.7	91.3	68.4
Bengtson	66.1	81.9	73.1	69.5	74.7	72.0

Table 11: Realistic Setting

The *CEAF*$_{orig}$ numbers in Table 10 illustrate the big influence the system mentions have on precision (e.g. the very low precision number for *Soon (SM1)*). The big improvement for *Soon (SM2)* is largely due to the system mentions it uses, rather than to different coreference models.

Both B$^3_{r\&n}$ and *CEAF*$_{r\&n}$ show no serious problems in the experimental results. However, as discussed before, they fail to penalize the spurious entities with twinless system mentions adequately.

3.3.2 Experiment 2

We compare results of Bengtson & Roth's (2008) system with our *Soon (SM2)* system. Bengtson & Roth's embedded mention tagger aims at high precision, generating half of the mentions *SM1* generates (explicit statistics are not available to us).

Bengtson & Roth report a B^3 F-score for system mentions, which is very close to the one for true mentions. Their B^3-variant does not impute errors of twinless mentions and is assumed to be quite similar to the B$^3_0$ strategy.

We integrate both the B$^3_0$ and B$^3_{sys}$ variants into their system and show results in Table 11 (we cannot report significance, because we do not have access to results for single documents in Bengtson & Roth's system). It can be seen that, when different variants of evaluation metrics are applied, the performance of the systems vary wildly.

4 Conclusions

In this paper, we address problems of commonly used evaluation metrics for coreference resolution and suggest two variants for B^3 and *CEAF*, called B^3_{sys} and *CEAF*$_{sys}$. In contrast to the variants proposed by Stoyanov et al. (2009), B^3_{sys} and *CEAF*$_{sys}$ are able to deal with end-to-end systems which do not use any gold information. The numbers produced by B^3_{sys} and $CEAF_{sys}$ are able to indicate the resolution performance of a system more adequately, without being tricked easily by twisting preprocessing components. We believe that the explicit description of evaluation metrics, as given in this paper, is a precondition for the re-

	B^3_{sys}			B^3_0			B^3_{all}			$B^3_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	65.7	76.8	70.8	57.0	91.1	**70.1**	65.1	85.8	74.0	65.1	78.7	71.2
Soon (SM2)	64.1	87.3	**73.9**	54.7	91.3	68.4	64.3	87.1	73.9	64.3	84.9	**73.2**

Table 9: Realistic Setting B^3 Variants

	CEAF$_{sys}$			*CEAF*$_{orig}$			*CEAF*$_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	66.4	61.2	63.7	62.0	39.9	48.5	62.1	59.8	60.9
Soon (SM2)	67.4	65.2	**66.3**	60.0	56.6	**58.2**	60.0	66.2	62.9

Table 10: Realistic Setting *CEAF* Variants

liabe comparison of end-to-end coreference resolution systems.

Acknowledgements. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD. scholarship. We would like to thank Éva Mújdricza-Maydt for implementing the mention taggers.

References

Bagga, Amit & Breck Baldwin (1998). Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pp. 563–566.

Bengtson, Eric & Dan Roth (2008). Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 294–303.

Culotta, Aron, Michael Wick & Andrew McCallum (2007). First-order probabilistic models for coreference resolution. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, N.Y., 22–27 April 2007, pp. 81–88.

Denis, Pascal & Jason Baldridge (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, N.Y., 22–27 April 2007, pp. 236–243.

Haghighi, Aria & Dan Klein (2007). Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 23–30 June 2007, pp. 848–855.

Luo, Xiaoqiang (2005). On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 25–32.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla & Salim Roukos (2004). A mention-synchronous coreference resolution algorithm based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, 21–26 July 2004, pp. 136–143.

Mitchell, Alexis, Stephanie Strassel, Shudong Huang & Ramez Zakhary (2004). *ACE 2004 Multilingual Training Corpus*. LDC2005T09, Philadelphia, Penn.: Linguistic Data Consortium.

Nicolae, Cristina & Gabriel Nicolae (2006). BestCut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 275–283.

Ponzetto, Simone Paolo & Michael Strube (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* New York, N.Y., 4–9 June 2006, pp. 192–199.

Rahman, Altaf & Vincent Ng (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing,* Singapore, 6-7 August 2009, pp. 968–977.

Rand, William R. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Recasens, Marta & Eduard Hovy (2010). *BLANC: Implementing the Rand index for coreference evaluation*. Submitted.

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

Stoyanov, Veselin, Nathan Gilbert, Claire Cardie & Ellen Riloff (2009). Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing,* Singapore, 2–7 August 2009, pp. 656–664.

Versley, Yannick, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang & Alessandro Moschitti (2008). BART: A modular toolkit for coreference resolution. In *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 15–20 June 2008, pp. 9–12.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly & Lynette Hirschman (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6),* pp. 45–52. San Mateo, Cal.: Morgan Kaufmann.

Witten, Ian H. & Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco, Cal.: Morgan Kaufmann.

Yang, Xiaofeng, Jian Su & Chew Lim Tan (2008). A twin-candidate model for learning-based anaphora resolution. *Computational Linguistics*, 34(3):327–356.

A B^3_{sys} Example Output

Here, we provide additional examples for analyzing the behavior of B^3_{sys} where we systematically vary system outputs. Since we proposed B^3_{sys} for dealing with end-to-end systems, we consider only examples also containing twinless mentions. The systems in Table 12 and 14 generate different twinless key mentions while keeping the twinless system mentions untouched. In Table 13 and 15, the number of twinless system mentions changes through different responses and the number of twinless key mentions is fixed.

In Table 12, B^3_{sys} recall goes up when more key mentions are resolved into the correct set. And the precision stays the same, because there is no change in the number of the erroneous resolutoins (i.e. the spurious cluster with mentions i and j). For the examples in Tables 13 and 15, B^3_{sys} gives worse precision to the outputs with more spurious resolutions, and the same recall if the systems resolve key mentions in the same way. Since the set of key mentions intersects with the set of twinless system mentions in Table 14, we do not have an intuitive explanation for the decrease in precision from response$_1$ to response$_4$. However, both the F-score and the recall still show the right tendency.

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b}	{i j}	0.857	0.280	0.422
response$_2$	{a b c}	{i j}	0.857	0.440	0.581
response$_3$	{a b c d}	{i j}	0.857	0.68	0.784
response$_4$	{a b c d e}	{i j}	0.857	1.0	0.923

Table 12: Analysis of B^3_{sys} 1

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b c}	{i j}	0.857	0.440	0.581
response$_2$	{a b c}	{i j k}	0.75	0.440	0.555
response$_3$	{a b c}	{i j k l}	0.667	0.440	0.530
response$_4$	{a b c}	{i j k l m}	0.6	0.440	0.508

Table 13: Analysis of B^3_{sys} 2

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b i j}	0.643	0.280	0.390
response$_2$	{a b c i j}	0.6	0.440	0.508
response$_3$	{a b c d i j}	0.571	0.68	0.621
response$_4$	{a b c d e i j}	0.551	1.0	0.711

Table 14: Analysis of B^3_{sys} 3

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b c i j}	0.6	0.440	0.508
response$_2$	{a b c i j k}	0.5	0.440	0.468
response$_3$	{a b c i j k l}	0.429	0.440	0.434
response$_4$	{a b c i j k l m}	0.375	0.440	0.405

Table 15: Analysis of B^3_{sys} 4