Evaluation for Partial Event Coreference

Jun Araki Eduard Hovy Teruko Mitamura
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
junaraki@cs.cmu.edu, hovy@cmu.edu, teruko@cs.cmu.edu

Abstract

This paper proposes an evaluation scheme to measure the performance of a system that detects hierarchical event structure for event coreference resolution. We show that each system output is represented as a forest of unordered trees, and introduce the notion of conceptual event hierarchy to simplify the evaluation process. We enumerate the desiderata for a similarity metric to measure the system performance. We examine three metrics along with the desiderata, and show that metrics extended from MUC and BLANC are more adequate than a metric based on Simple Tree Matching.

1 Introduction

Event coreference resolution is the task to determine whether two event mentions refer to the same event. This task is important since resolved event coreference is useful in various tasks such as topic detection and tracking, information extraction, question answering, textual entailment, and contradiction detection.

A key challenge for event coreference resolution is that one can define several relations between two events, where some of them exhibit subtle deviation from perfect event identity. For clarification, we refer to perfect event identity as *full (event) coreference* in this paper. To address the subtlety in event identity, Hovy et al. (2013) focused on two types of partial event identity: *subevent* and *membership*. Subevent relations form a stereotypical sequence of events, or a script (Schank and Abelson, 1977; Chambers and Jurafsky, 2008). Membership relations represent instances of an event collection. We refer to both as *partial (event) coreference* in this paper. Figure 1 shows some examples of the subevent and membership relations in the illustrative text below, taken from the Intelligence Community domain of violent events. Unlike full coreference, partial coreference is a directed relation, and forms hierarchical event structure, as shown in Figure 1. Detecting partial coreference itself is an important task because the resulting event structures are beneficial to text comprehension. In addition, such structures are also useful as background knowledge information to resolve event coreference.

> A car bomb that police said was set by Shining Path guerrillas **ripped off**(E4) the front of a Lima police station before dawn Thursday, **wounding**(E5) 25 people. The **attack**(E6) marked the return to the spotlight of the feared Maoist group, recently overshadowed by a smaller rival band of rebels. The pre-dawn **bombing**(E7) **destroyed**(E8) part of the police station and a municipal office in Lima's industrial suburb of Ate-Vitarte, **wounding**(E9) 8 police officers, one seriously, Interior Minister Cesar Saucedo told reporters. The bomb **collapsed**(E11) the roof of a neighboring hospital, **injuring**(E12) 15, and **blew out**(E13) windows and doors in a public market, **wounding**(E14) two guards.

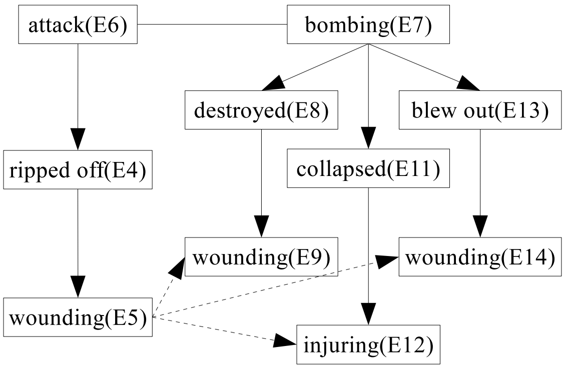


Figure 1: Examples of subevent and membership relations. Solid and dashed arrows represent subevent and membership relations respectively, with the direction from a parent to its subevent or member. For example, we say that E4 is a subevent of E6. Solid lines without any arrow heads represent full coreference.

In this paper, we address the problem of evalu-

Proceedings of the 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 68–76,
Baltimore, Maryland, USA, June 22-27, 2014. ©2014 Association for Computational Linguistics

ating the performance of a system that detects partial coreference in the context of event coreference resolution. This problem is important because, as with other tasks, a good evaluation method for partial coreference will facilitate future research on the task in a consistent and comparable manner. When one introduces a certain evaluation metric to such a new complex task as partial event coreference, it is often unclear what metric is suitable to what evaluation scheme for the task under what assumptions. It is also obscure how effectively and readily existing algorithms or tools, if any, can be used in a practical setting of the evaluation. In order to resolve these sub-problems for partial coreference evaluation, we need to formulate an evaluation scheme that defines assumptions to be made regarding the evaluation, specifies some desiderata that an ideal metric should satisfy for the task, and examines how adequately particular metrics can satisfy them. For this purpose, we specifically investigate three existing algorithms MUC, BLANC, and Simple Tree Matching (STM).

The contributions of this work are as follows:

- We introduce a conceptual tree hierarchy that simplifies the evaluation process for partial event coreference.
- We present a way to extend MUC, BLANC, and STM for the case of unordered trees. Those metrics are generic and flexible enough to be used in evaluations involving data structures based on unordered trees.
- Our experimental results indicate that the extended MUC and BLANC are better than Simple Tree Matching for evaluating partial coreference.

2 Related Work

Recent studies on both entity and event coreference resolution use several metrics to evaluate system performance (Bejan and Harabagiu, 2010; Lee et al., 2012; Durrett et al., 2013; Lassalle and Denis, 2013) since there is no agreement on a single metric. Currently, five metrics are widely used: MUC (Vilain et al., 1995), B-CUBED (Bagga and Baldwin, 1998), two CEAF metrics CEAF-ϕ_3 and CEAF-ϕ_4 (Luo, 2005), and BLANC (Recasens and Hovy, 2011). We can divide these metrics into two groups: cluster-based metrics, e.g., B-CUBED and CEAF, and link-based metrics, e.g., MUC and BLANC. The former group is not applicable to evaluate partial coreference because it is unclear how to define a cluster. The latter is not readily applicable to the evaluation because it is unclear how to penalize incorrect directions of links. We discuss these aspects in Section 4.1 and Section 4.2.

Tree Edit Distance (TED) is one of the traditional algorithms for measuring tree similarity. It has a long history of theoretical studies (Tai, 1979; Zhang and Shasha, 1989; Klein, 1998; Bille, 2005; Demaine et al., 2009; Pawlik and Augsten, 2011). It is also widely studied in many applications, including Natural Language Processing (NLP) tasks (Mehdad, 2009; Wang and Manning, 2010; Heilman and Smith, 2010; Yao et al., 2013). However, TED has a disadvantage: we need to predefine appropriate costs for basic tree-edit operations. In addition, an implementation of TED for unordered trees is fairly complex.

Another tree similarity metric is Simple Tree Matching (STM) (Yang, 1991). STM measures the similarity of two trees by counting the maximum match with dynamic programming. Although this algorithm was also originally developed for ordered trees, the underlying idea of the algorithm is simple, making it relatively easy to extend the algorithm for unordered trees.

Tree kernels have been also widely studied and applied to NLP tasks, more specifically, to capture the similarity between parse trees (Collins and Duffy, 2001; Moschitti et al., 2008) or between dependency trees (Croce et al., 2011; Srivastava et al., 2013). This method is based on a supervised learning model with training data; hence we need a number of pairs of trees and associated numeric similarity values between these trees as input. Thus, it is not appropriate for an evaluation setting.

3 Evaluation Scheme

When one formulates an evaluation scheme for a new task, it is important to define assumptions for the evaluation and desiderata that an ideal metric should satisfy. In this section, we first describe assumptions for partial coreference evaluation, and introduce the notion of conceptual event hierarchy to address the challenge posed by one of the assumptions. We then enumerate the desiderata for a metric.

3.1 Assumptions on Partial Coreference

We make the following three assumptions to evaluate partial coreference.

Twinless mentions: Twinless mentions (Stoyanov et al., 2009) are the mentions that exist in the gold standard but do not in a system response, or vice versa. In reality, twinless mentions often happen since an end-to-end system might produce them in the process of detecting mentions. The assumption regarding twinless mentions has been investigated in research on entity coreference resolution. Cluster-based metrics such as B-CUBED and CEAF assume that a system is given true mentions without any twinless mentions in the gold standard, and then resolves full coreference on them. Researchers have made different assumptions about this issue. Early work such as (Ji et al., 2005) and (Bengtson and Roth, 2008) simply ignored such mentions. Rahman and Ng (2009) removed twinless mentions that are singletons in a system response. Cai and Strube (2010) proposed two variants of B-CUBED and CEAF that can deal with twinless mentions in order to make the evaluation of end-to-end coreference resolution system consistent.

In evaluation of partial coreference where twinless mentions can also exist, we believe that the value of making evaluation consistent and comparable is the most important, and hypothesize that it is possible to effectively create a metric to measure the performance of partial coreference while dealing with twinless mentions. A potential problem of making a single metric handle twinless mentions is that the metric would not be informative enough to show whether a system is good at identifying coreference links but poor at identifying mentions, or vice versa (Recasens and Hovy, 2011). However, our intuition is that the problem is avoidable by showing the performance of mention identification with metrics such as precision, recall, and the F-measure simultaneously with the performance of link identification. In this work, therefore, we assume that a metric for partial coreference should be able to handle twinless mentions.

Intransitivity: As described earlier, partial coreference is a directed relation. We assume that partial coreference is not transitive. To illustrate the intransitivity, let $e_i \xrightarrow{s} e_j$ denote a subevent relation that e_j is a subevent of e_i. In Figure 1, we have $E7 \xrightarrow{s} E8$ and $E8 \xrightarrow{s} E9$. In this case, E9 is not a subevent of E7 due to the intransitivity of subevent relations. One could argue that the event 'wounding(E9)' is one of stereotypical events triggered by the event 'bombing(E7)', and thus $E7 \xrightarrow{s} E9$. However, if we allow transitivity of partial coreference, then we have to measure all implicit partial coreference links (e.g., the one between E7 and E9) from hierarchical event structures. Consequently, this evaluation policy could result in an unfair scoring scheme biased toward large event hierarchy.

Link propagation: We assume that partial coreference links can be propagated due to a combination of full coreference links with them. To illustrate the phenomenon, let $e_i \Leftrightarrow e_j$ denote full coreference between e_i and e_j. In Figure 1, we have $E6 \Leftrightarrow E7$ and $E7 \xrightarrow{s} E8$. In this case, E8 is also a subevent of E6, i.e., $E6 \xrightarrow{s} E8$. The rationale behind this assumption is that if a system identifies $E6 \xrightarrow{s} E8$ instead of $E7 \xrightarrow{s} E8$, then there is no reason to argue that the identified subevent relation is incorrect given that $E6 \Leftrightarrow E7$ and $E7 \xrightarrow{s} E8$. The discussion here also applies to membership relations.

3.2 Conceptual Event Hierarchy

The assumption of link propagation poses a challenge in measuring the performance of partial coreference. We illustrate the challenge with the example in the discussion on link propagation above. We focus only on subevent relations to describe our idea, but one can apply the same discussion to membership relations. Suppose that a system detects a subevent link $E7 \xrightarrow{s} E8$, but not $E6 \xrightarrow{s} E8$. Then, is it reasonable to give the system a double reward for two links $E7 \xrightarrow{s} E8$ and $E6 \xrightarrow{s} E8$ due to link propagation, or should one require a system to perform such link propagation and detect $E7 \xrightarrow{s} E8$ as well for the system to achieve the double reward? In the evaluation scheme based on event trees whose nodes represent event mentions, we need to predefine how to deal with link propagation of full and partial coreference in evaluation. In particular, we must pay attention to the potential risk of overcounting partial corefrence links due to link propagation.

To address the complexity of link propagation, we introduce a conceptual event tree where each node represents a conceptual event rather than an event mention. Figure 2 shows an example of a conceptual subevent tree constructed from full coreference and subevent relations in Figure 1. Using set notation, each node of the tree represents an abstract event. For instance, node {E6, E7} represents an "attacking" event which both event mentions E6 and E7 refer to.

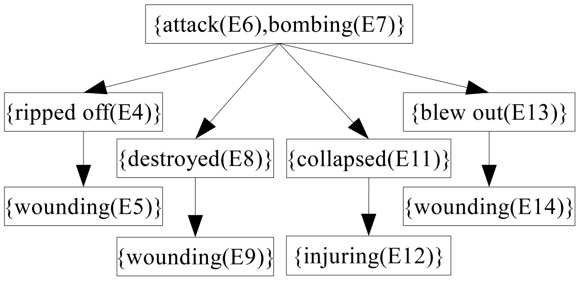


Figure 2: A conceptual subevent tree constructed from the full coreference and subevent relations in Figure 1.

The notion of a conceptual event tree obviates the need to cope with link propagation, thereby simplifying the evaluation for partial coreference. Given a conceptual event tree, an evaluation metric is basically just required to measure how many links in the tree a system successfully detects. When comparing two conceptual event trees, a link in a tree is identical to one in the other tree if there is at least one event mention shared in parent nodes of those links and at least one shared in child nodes of those links. For example, suppose that system A identifies $E6 \xrightarrow{s} E8$, system B $E7 \xrightarrow{s} E8$, system C both, and all the systems identify $E6 \Leftrightarrow E7$ in Figure 1. In this case, they gain the same score since the subevent links that they identify correspond to one correct subevent link $\{E6, E7\} \xrightarrow{s} \{E8\}$ in Figure 2. It is possible to construct the conceptual event hierarchy for membership relations in the same way as described above. This means that the conceptual event hierarchy allows us to show the performance of a system on each type of partial coreference separately, which leads to more informative evaluation output.

One additional note is that the conceptual event tree representing partial coreference is an unordered tree, as illustrated in Figure 2. Although we could represent a subevent tree with an ordered tree because of the stereotypical sequence of subevents given by definition, partial coreference is in general represented with a forest of unordered trees[1].

3.3 Desiderata for Metrics

In general, a system output of partial event coreference in a document is represented not by a single tree but by a forest, i.e., a set of disjoint trees whose nodes are event mentions that appear in the document. Let T be a tree, and let F be a forest $F = \{T_i\}$. Let $sim(F_g, F_r) \in [0, 1]$ denote a similarity score between the gold standard forest F_g and a system response forest F_r. We define the following properties that an ideal evaluation metric for partial event coreference should satisfy.

P1. *Identity*: $sim(F_1, F_1) = 1$.

P2. *Symmetricity*: $sim(F_1, F_2) = sim(F_2, F_1)$.

P3. *Zero*: $sim(F_1, F_2) = 0$ if F_1 and F_2 are totally different forests.

P4. *Monotonicity*: The metric score should increase from 0 to 1 monotonically as two totally different forests approach the identical one.

P5. *Linearity*: The metric score should increase linearly as each single individual correct piece of information is added to a system response.

The first three properties are relatively intuitive. P4 is important because otherwise a higher score by the metric does not necessarily mean higher quality of partial event coreference output. In P5, a correct piece of information is the addition of one correct link or the deletion of one incorrect link. This property is useful for tracking performance progress over a certain period of time. If the metric score increases nonlinearly, then it is difficult to compare performance progress such as a 0.1 gain last year and a 0.1 gain this year, for example.

In addition, one can think of another property with respect to structural consistency. The motivation for the property is that one might want to give more reward to partial coreference links that form hierarchical structures, since they implicitly form sibling relations among child nodes. For instance, suppose that system A detects two links $\{E6, E7\} \xrightarrow{s} \{E8\}$ and $\{E6, E7\} \xrightarrow{s} \{E11\}$, and system B two links $\{E8\} \xrightarrow{s} \{E9\}$ and $\{E11\} \xrightarrow{s} \{E12\}$ in Figure 2. We can think that system A performs better since the system successfully detects an implicit subevent sibling relation between $\{E8\}$ and $\{E11\}$ as well. Due to space limitations, however, we do not explore the property in this work, and leave it for future work.

[1] For example, it is impossible to intuitively define a sequence of child nodes in a membership event tree in Figure 1.

4 Evaluation Metrics

In this section, we examine three evaluation metrics based on MUC, BLANC, and STM respectively under the evaluation scheme described in Section 3.

4.1 B-CUBED and CEAF

B-CUBED regards a coreference chain as a set of mentions, and examines the presence and absence of mentions in a system response that are relative to each of their corresponding mentions in the gold standard (Bagga and Baldwin, 1998). Let us call such set a mention cluster. A problem in applying B-CUBED to partial coreference is that it is difficult to properly form a mention cluster for partial coreference. In Figure 2, for example, we could form a gold standard cluster containing all nodes in the tree. We could then form a system response cluster, given a certain system output. The problem is that B-CUBED's way of counting mentions overlapped in those clusters cannot capture parent-child relations between the mentions in a cluster. It is also difficult to extend the counting algorithm to incorporate such relations in an intuitive manner. Therefore, we observe that B-CUBED is not appropriate for evaluating partial coreference.

We see the basically same reason for the inadequacy of CEAF. It also regards a coreference chain as a set of mentions, and measures how many mentions two clusters share using two similarity metrics $\phi_3(R, S) = |R \cap S|$ and $\phi_4(R, S) = \frac{2|R \cap S|}{|R|+|S|}$, given two clusters R and S. One can extend CEAF for partial coreference by selecting the most appropriate tree similarity algorithm for ϕ that works well with the algorithm to compute maximum bipartite matching in CEAF. However, that is another line of work, and due to space limitations we leave it for future work.

4.2 Extension to MUC and BLANC

MUC relies on the minimum number of links needed when mapping a system response to the gold standard (Vilain et al., 1995). Given a set of key entities K and a set of response entities R, precision of MUC is defined as the number of common links between entities in K and R divided by the number of links in R, whereas recall of MUC is defined as the number of common links between entities in K and R divided by the number of links in K. After finding a set of mention clusters by resolving full coreference, we can compute the number of correct links by counting all links spanning in those mention clusters that matched the gold standard. It is possible to apply the idea of MUC to the case of partial coreference simply by changing the definition of a correct link. In the partial coreference case, we define a correct link as a link matched with the gold standard including its direction. Let MUC$_p$ denote such extension to MUC for partial coreference.

Similarly, it is also possible to define an extension to BLANC. Let BLANC$_p$ denote the extension. BLANC computes precision, recall, and F1 scores for both coreference and non-coreference links, and average them for the final score (Recasens and Hovy, 2011). As with MUC$_p$, BLANC$_p$ defines a correct link as a link matched with the gold standard including its direction. Another difference between BLANC and BLANC$_p$ is the total number of mention pairs, denoted as L. In the original BLANC, $L = N(N-1)/2$ where N is the total number of mentions in a document. We use $L_p = N(N-1)$ instead for BLANC$_p$ since we consider two directed links in partial coreference with respect to each undirected link in full coreference.

4.3 Extension to Simple Tree Matching

The underlying idea of STM is that if two trees have more node-matching, then they are more similar. The original STM uses a dynamic programming approach to perform recursive node-level matching in a top-down fashion. In the case of partial coreference, we cannot readily use the approach because partial coreference is represented with unordered trees, and thus time complexity of node-matching is the exponential order with respect to the number of child nodes. However, partial event coreference is normally given in a small hierarchy with three levels or less. Taking advantage of this fact and assuming that each event mention is uniquely identified in a tree, we extend STM for the case of unordered trees by using greedy search. Algorithm 1 shows an extension to the STM algorithm for unordered trees.

We can also naturally extend STM to take forests as input. Figure 3 shows how one can convert a forest into a single tree whose subtrees are the trees in the forest by introducing an additional dummy root node on top of those tree. The resulting tree is also an unordered tree, and thus we can apply Algorithm 1 to that tree to measure the sim-

Algorithm 1 Extended simple tree matching for unordered trees.

```
Input: two unordered trees $A$ and $B$
Output: score
procedure SimpleTreeMatching($A$, $B$)
    if the roots of $A$ and $B$ have different elements then
        return 0
    else
        $s := 1$        ▷ The initial score for the root match.
        $m :=$ the number of first-level sub-trees of $A$
        $n :=$ the number of first-level sub-trees of $B$
        for $i = 1 \rightarrow m$ do
            for $j = 1 \rightarrow n$ do
                if $A_i$ and $B_j$ have the same element then
                    s = s + SimpleTreeMatching($A_i$, $B_j$)
```

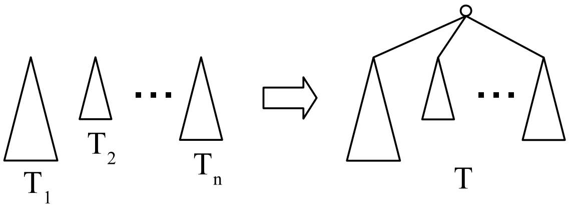


Figure 3: Conversion from a forest to a single tree with an additional dummy root.

ilarity of two forests comprising unordered trees. Let STM$_p$ denote the extended STM. Finally, we normalize STM$_p$. Let NSTM$_p$ be a normalized version of STM$_p$ as follows: $NSTM_p(F_1, F_2) = STM_p(F_1, F_2)/max(|F_1|, |F_2|)$ where $|F|$ denotes the number of nodes in F.

4.4 Flexibility of Metrics

Making assumptions on evaluation for a particular task and defining desiderata for a metric determine what evaluation scheme we are going to formulate. However, this kind of effort tends to make resulting evaluation metrics too restrictive to be reusable in other tasks. Such metrics might be adequate for that task, but we also value the flexibility of a metric that can be directly used or be easily extended to other tasks. To investigate the flexibility of MUC$_p$, BLANC$_p$ and STM$_p$, we will examine these metrics without making the assumptions of twinless mentions and intransitivity of partial coreference against each metric. We consider that the assumption of link propagation is more fundamental and regard it as a basic premise, and thus we will continue to make that assumption.

MUC was originally designed to deal with response links spanning mentions that even key links do not reach. Thus, it is able to handle twinless mentions. If we do not assume intransitivity of partial coreference, we do not see any difficulty in changing the definition of correct links in MUC$_p$ and making it capture transitive relations. Therefore, MUC$_p$ does not require both assumptions of twinless mentions and intransitivity.

In contrast, BLANC was originally designed to handle true mentions in the gold standard. Since BLANC$_p$ does not make any modifications on this aspect, it cannot deal with twinless mentions either. As for intransitivity, it is possible to easily change the definition of correct and incorrect links in BLANC$_p$ to detect transitive relations. Thus, BLANC$_p$ does not require intransitivity but does require the assumption of no twinless mentions.

Since STM$_p$ simply matches elements in nodes as shown in Algorithm 1, it does not require the assumption of twinless mentions. With respect to intransitivity, we can extend STM$_p$ by adding extra edges from a parent to grandchild nodes or others and applying Algorithm 1 to the modified trees. Hence, it does not require the assumption of intransitivity.

5 Experiments

To empirically examine the three metrics described in Section 4.2 and Section 4.3, we conducted an experiment using the artificial data shown in Table 1. Since BLANC$_p$ cannot handle twinless mentions, we removed twinless mentions. We first created the gold standard shown in the first row of the table. It contains fifty events, twenty one singleton events, and seven event trees with three levels or less. We believe this distribution of partial coreference is representative of that of real data. We then created several system responses that are ordered toward two extremes. One extreme is all singletons in which they do not have correct links. The other is a single big tree that merges all event trees including singletons in the gold standard.

Figure 4 shows how the three metrics behave in two cases: (a) we increase the number of correct links from all singletons to the perfect output (equal to the gold standard), and (b) we increase the incorrect links from the perfect output to a single tree merging all trees in the gold standard. In the former case, we started with System 3 in Table 1. Next we added one correct link $28 \xrightarrow{s} 29$ shown in System 2. This way, we added correct links up to the perfect output one by one in a bottom-up fashion. In the latter case, we started

Response	Output
Gold standard	**(1(2(6))(3(7))(4)(5)) (8(9(11)(12))(10)) (13(14)(15)(16)(17)(18)) (19(20(21))(22)) (23(24)(25)) (26(27)) (28(29))** (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50)
System 1	**(1(4)(5)(2(6))(3(7))) (8(9(11)(12))(10)) (13(18)(14)(15)(16)(17)) (19(22)(20(21))) (23(24)(25)) (26(27)) (28(29))** (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) **(49(50))**
System 2	(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) **(28(29))** (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50)
System 3	(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50)

Table 1: Examples of a system response against a gold standard partial coreference. Each event tree is shown in the bold font and in the Newick standard format with parentheses.

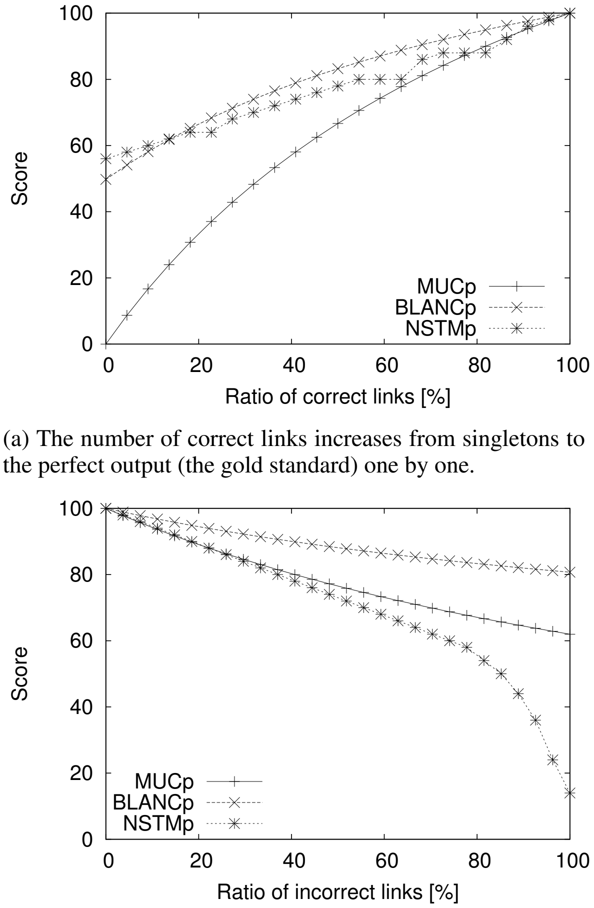
with the perfect output, and then added one incorrect link $49 \xrightarrow{s} 50$ shown in System 1. In a manner similar to case (a), we added incorrect links up to the merged tree one by one in a bottom-up fashion.

The results indicate that MUC$_p$ and BLANC$_p$ meet the desiderata defined in Section 3.3 more adequately than NSTM$_p$. The curve of MUC$_p$ and BLANC$_p$ in Figure 4 are close to the linearity, which is practically useful as a metric. In contrast, NSTM$_p$ fails to meet P4 and P5 in case (a), and fails to meet P5 in case (b). This is because STM first checks whether root nodes of two trees have the same element, and if the root nodes have different elements, STM stops searching the rest of nodes in the trees.

6 Discussion

In Section 4.4, we observed that MUC$_p$ and STM$_p$ are more flexible than BLANC$_p$ because they can measure the performance coreference in the case of twinless mentions as well. The experimental results in Section 5 show that MUC$_p$ and BLANC$_p$ more adequate in terms of the five properties defined in Section 3.3. Putting these together, MUC$_p$ seems the best metric for partial event coreference. However, MUC has two disadvantages that (1) it prefers systems that have more mentions per entity (event), and (2) it ignores recall for singletons (Pradhan et al., 2011). MUC$_p$ also has these disadvantages. Thus, BLANC$_p$ might be the best choice for partial coreference if we could assume that a system is given true mentions in the gold standard.

Although STM$_p$ fails to satisfy P4 and P5, it has potential power to capture structural proper-



(a) The number of correct links increases from singletons to the perfect output (the gold standard) one by one.

(b) The number of incorrect links increases from the perfect output to a single tree merging all trees one by one.

Figure 4: Score comparison among MUC$_p$, BLANC$_p$, and NSTM$_p$.

ties of partial coreference described in Section 3.3. This is because STM's recursive fashion of node-counting can be easily extend to counting the number of correct sibling relations.

7 Conclusion

We proposed an evaluation scheme for partial event coreference with conceptual event hierarchy constructed from mention-based event trees. We discussed possible assumptions that one can make, and examined extensions to three existing metrics. Our experimental results indicate that the extensions to MUC and BLANC are more adequate than the extension to STM. To our knowledge, this is the first work to argue an evaluation scheme for partial event coreference. Nevertheless, we believe that our scheme is generic and flexible enough to be applicable to other directed relations of events (e.g., causality and entailment) or other related tasks to compare hierarchical data based on unordered trees (e.g., ontology comparison). One future work is to improve the metrics by incorporating structural consistency of event trees as an additional property and implementing the metrics from the perspective of broad contexts beyond local evaluation by link-based counting.

8 Acknowledgements

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA or the US government. We would like to thank anonymous reviewers for their helpful comments.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for Scoring Coreference Chains. In *Proceedings of LREC 1998 Workshop on Linguistics Coreference*, pages 563–566.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised Event Coreference Resolution with Rich Linguistic Features. In *Proceedings of ACL 2010*, pages 1412–1422.

Eric Bengtson and Dan Roth. 2008. Understanding the Value of Features for Coreference Resolution. In *Proceedings of EMNLP 2008*, pages 294–303.

Philip Bille. 2005. A Survey on Tree Edit Distance and Related Problems. *Theoretical Computer Science*, 337(1-3):217–239.

Jie Cai and Michael Strube. 2010. Evaluation Metrics For End-to-End Coreference Resolution Systems. In *Proceedings of SIGDIAL 2010*, pages 28–36.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *Proceedings of ACL-HLT 2008*, pages 789–797.

Michael Collins and Nigel Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of NIPS 2001*, pages 625–632.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured Lexical Similarity via Convolution Kernels on Dependency Trees. In *Proceedings of EMNLP 2011*, pages 1034–1046.

Erik D. Demaine, Shay Mozes, Benjamin Rossman, and Oren Weimann. 2009. An Optimal Decomposition Algorithm for Tree Edit Distance. *ACM Transactions on Algorithms (TALG)*, 6(1):2:1–2:19.

Greg Durrett, David Hall, and Dan Klein. 2013. Decentralized Entity-Level Modeling for Coreference Resolution. In *Proceedings of ACL 2013*, pages 114–124.

Michael Heilman and Noah A. Smith. 2010. Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions. In *Proceedings of NAACL-HLT 2013*, pages 1011–1019.

Eduard Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. 2013. Events are Not Simple: Identity, Non-Identity, and Quasi-Identity. In *Proceedings of NAACL-HLT 2013 Workshop on Events: Definition, Detection, Coreference, and Representation*, pages 21–28.

Heng Ji, David Westbrook, and Ralph Grishman. 2005. Using Semantic Relations to Refine Coreference Decisions. In *Proceedings of EMNLP/HLT 2005*, pages 17–24.

Philip N. Klein. 1998. Computing the Edit-Distance Between Unrooted Ordered Trees. In *Proceedings of the 6th European Symposium on Algorithms (ESA)*, pages 91–102.

Emmanuel Lassalle and Pascal Denis. 2013. Improving pairwise coreference models through feature space hierarchy learning. In *Proceedings of ACL 2013*, pages 497–506.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 489–500.

Xiaoqiang Luo. 2005. On Coreference Resolution Performance Metrics. In *Proceedings of EMNLP 2005*, pages 25–32.

Yashar Mehdad. 2009. Automatic Cost Estimation for Tree Edit Distance Using Particle Swarm Optimization. In *Proceedings of ACL-IJCNLP 2009*, pages 289–292.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, 34(2):193–224.

Mateusz Pawlik and Nikolaus Augsten. 2011. RTED: A Robust Algorithm for the Tree Edit Distance. *Proceedings of the VLDB Endowment (PVLDB)*, 5(4):334–345.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of CoNLL Shared Task 2011*, pages 1–27.

Altaf Rahman and Vincent Ng. 2009. Supervised Models for Coreference Resolution. In *Proceedings of EMNLP 2009*, pages 968–977.

Marta Recasens and Eduard Hovy. 2011. BLANC: Implementing the Rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates.

Shashank Srivastava, Dirk Hovy, and Eduard Hovy. 2013. A Walk-Based Semantically Enriched Tree Kernel Over Distributed Word Representations. In *Proceedings of EMNLP 2013*, pages 1411–1416.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the-Art. In *Proceedings of ACL/IJCNLP 2009*, pages 656–664.

Kuo-Chung Tai. 1979. The Tree-to-Tree Correction Problem. *Journal of the ACM (JACM)*, 26(3):422–433.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A Model-Theoretic Coreference Scoring Scheme. In *Proceedings of the 6th Message Understanding Conference (MUC)*, pages 45–52.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In *Proceedings of COLING 2010*, pages 1164–1172.

Wuu Yang. 1991. Identifying Syntactic Differences Between Two Programs. *Software: Practice and Experience*, 21(7):739–755.

Xuchen Yao, Benjamin Van Durme, Chris Callison-burch, and Peter Clark. 2013. Answer Extraction as Sequence Tagging with Tree Edit Distance. In *Proceedings of NAACL-HLT 2013*, pages 858–867.

Kaizhong Zhang and Dennis Shasha. 1989. Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. *SIAM J. Comput.*, 18(6):1245–1262.