Evaluation Algorithms for Event Nugget Detection : A Pilot Study

Zhengzhong Liu, Teruko Mitamura, Eduard Hovy
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
liu@cs.cmu.edu, teruko@cs.cmu.edu, hovy@cmu.edu

Abstract

Event Mention detection is the first step in textual event understanding. Proper evaluation is important for modern natural language processing tasks. In this paper, we present our evaluation algorithm and results during the Event Mention Evaluation pilot study. We analyze the problems of evaluating multiple event mention attributes and discontinuous event mention spans. In addition, we identify a few limitations in the evaluation algorithm used for the pilot task and propose some potential improvements.

1 Introduction

Textual event understanding has attracted a lot of attention in the community. Recent work has covered several areas about events, such as event mention detection(Li et al., 2013; Li et al., 2014) , event coreference (Bejan et al., 2005; Chen and Ji, 2009; Lee et al., 2012; Chen and Ng, 2013; Liu et al., 2013), and script understanding (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009). Event Mention detection is the fundamental preprocessing step for these tasks. However, downstream event researches often make minimal effort for event mention detection. For example, in event coreference work, Lee et al. (2012) do not make clear distinction between event and entity mentions. Bejan et al. (2005) and Liu et al. (2013) use oracle event mentions from human annotations. Building robust event mention detection system can help promote research in these areas and enable researchers to produce end-to-end systems. In this paper, we discuss our recent effort in providing a proper evaluation metric for event mention detection.

1.1 The Event Nugget Detection Task

As defined in Mitamura (2014), event nugget detection involves identifying semantic meaningful units (**mention span detection**) that refer to an event[1]. The task also requires a system to identify other attributes (**attribute detection**). In this pilot study, the attributes are *event type* and *realis status*.

(1) President Obama will `nominate` [realis: Other type: Personnel.Nominate] John Kerry for Secretary of State.

(2) He `carried out` the `assassination` [realis: Actual type: Life.Die] .

Example 1 shows one annotated event nugget `nominate`, which has the realis type “other” and event type “Personnel.Nominate”. Example 2 annotates one event nugget with discontinuous event span `carried out assassination`. The evaluation corpus is annotated with event nuggets that fall into 8 types of event[2]. Please refer to Mitamura (2014) for detailed definitions of the attributes.

1.2 Past Evaluation Methods

The Automatic Content Extraction 2005 evaluation task involves event extraction. The Event Detection and Recognition (VDR) task in the Automatic Content Extraction 2005 evaluation (NIST, 2005) evaluate the accuracy of event arguments and multiple other event attributes. However, event mention recognition is not directly evaluated (§3.2).

[1] This is similar to Event Trigger in ACE 2005, which is adopted in other work (Li et al., 2013; Li et al., 2014)

[2] These are *Life, Movement, Business, Conflict, Contact, Personnel, Transaction, Justice*

Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT 2015, pages 53–57,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

Li et al. (2013; 2014) evaluate event trigger detection using a mention-wise F-1 score. An event trigger is considered correct only when the span and event type are matched exactly. Errors from different sources are not separately presented.

In addition, most previous evaluations on event mention evaluation do not give partial credits to partial matches. Partial scoring is more important in the current setting because of the mention span detection task is difficult with discontinuous event nuggets.

2 The Evaluation Algorithm in Pilot Study

In this section, we describe our mention detection algorithms[3]. We will use the terms Event Nugget and Event Mention interchangeably.

2.1 Prerequisites

The main prerequisite for the evaluation is tokenization. In our pilot study, we provide a standard tokenization for all participants. System responses represent each event mention in terms of predefined token ids[4]. Discontinuous mentions can be easily represented using tokens.

2.2 Partial Span Scoring

The proposed evaluation produces a span similarity score for a pair of mentions (system and gold standard) between 0 and 1. Given a pair of mentions (G, S), we represent the span of each mention by a set of token ids (T_G , T_S). The span similarity score is defined as the Dice coefficient between the two sets (which is the same as the F-1 score).

$$\begin{aligned} Dice(T_G, T_S) &= \frac{2|T_G T_S|}{|T_G| + |T_S|} \\ &= \frac{2}{|T_G|/|T_G T_S| + |T_S|/|T_G T_S|} \\ = F1(T_G, T_S) &= \frac{2}{1/P + 1/R} \end{aligned}$$

2.3 Mention Mapping

To evaluate mention attributes, the evaluation algorithm needs to decide which system mention corresponds to a gold standard mention. We refer to this step as mention mapping. The input of our mention-mapping algorithm is the pairwise scores between all gold standard vs. system mention pair. We use the token-based Dice score (§2.2). Algorithm 1 shows our mapping algorithm to compute the mapping in one document.

Algorithm 1 Compute a mapping between system and gold standard mentions

```
Input: A list $L$ of scores $Dice(T_G, T_S)$ for all pair
    of G, S in the document
1: $M \leftarrow \varnothing; U \leftarrow \varnothing$
2: while $L \neq \varnothing$ do
3:     $G_m, S_n \leftarrow \arg\max_{(G,S)\in L} Dice(T_G, T_S)$
4:     if $S_n \notin U$ and $Dice(T_{G_m}, T_{S_n}) > 0$ then
5:         $M_{G_m} \leftarrow M_{G_m} \cup (S_n, Dice(T_{G_m}, T_{S_n}))$
6:         $U \leftarrow U \cup \{S_n\}$
Output: The mapping $M$
```

Algorithm 1 iteratively searches for the highest Dice score in all remaining mention pairs. Line 4 ensures that each system mention can only be mapped to one gold standard mention to avoid multiple counting. One gold standard mention is allowed to be mapped to multiple system mentions, which will be used in calculating attribute accuracy scores.

2.4 Overall Span Scoring

In the pilot study, we first evaluate the system's performance on span detection[5]. We use F-1 score (referred as mention level F-1 score to distinguish with the token level F-1 score in §2.2) for this task.

The definition of True Positive (TP) and False Positive (FP) for mention-level F-1 are slightly adjusted to reflect partial matching. TP values are accumulated according to Algorithm 2.

Precision, Recall, F-1 are calculated as followed:

$$P = \frac{TP}{TP + FP}; R = \frac{TP}{N_G}; F1 = \frac{2PR}{P + R}$$

N_G is the number of gold standard mentions.

In the study, we use $TP + FP$ as the denominator for Precision. We later identify a problem of this formulat. When FP is 0, even if the span range is

[3] Code base: `github.com/hunterhector/EvmEval`

[4] Some other KBP evaluations use character span evaluation, which will favor long words than short words. We argue that the difficulties in tokenizing a long word and a short word in English should be virtually the same; hence scoring these two cases differently is not fair.

[5] For simplicity, we describe our algorithm on a single document, the scorer will produce aggregate results for each metric with standard Micro and Macro average methods.

Algorithm 2 Compute TP and FP

```
Input: The set of gold standard $\mathcal{G}$; The mapping $M$ indexed by $G$; Number of system mentions $N_S$
$TP \leftarrow 0$; $FP \leftarrow 0$
for $\forall G \in \mathcal{G}$ do
    if $|M_G| = 0$ then
        $FP \leftarrow FP + 1$
    else
        $S_T \leftarrow \arg\max_{Dice}(S, Dice) \in M_G$
        $TP \leftarrow TP + Dice(G, S_T)$
Output: $TP$
```

not exactly correct, the system can still get perfect precision (though imperfect recall), which is counter-intuitive. If we calculate FP with $N_S - TP$, the precision, recall calculation will naturally resolve to:

$$P = \frac{TP}{N_S}; R = \frac{TP}{N_G}$$

The new formula is also aesthetically symmetric on precision and recall. We present the influence of this fix in §4.1.

2.5 Attribute Scoring

For each attribute and gold standard mention, we calculate the accuracy according to algorithm 3. This algorithm will give a system full credit even when the span matching is not perfect. In addition, when one system incorrectly splits one gold standard mention into two, we still give it credit as long as attributes are all predicted correctly.

Algorithm 3 Compute Attribute Accuracy for one Gold Standard Mention

```
Input: The gold standard mention G; The mapping $M$ indexed by $G$; The set $A$ indexing target attributes for all mentions;
$Accuracy \leftarrow 0$
for $S, Dice(T_S, T_G) \in M_G$ do
    if $A_S = A_G$ then
        $Accuracy \leftarrow Accuracy + 1/|M_G|$
Output: $Accuracy$
```

Gold He `carried out` the `assassination` [type: Life.Die] .

System 1 He `carried`[type: Life.Die] out the `assassination` [type: Life.Die] .

System 2 He `carried`[type: Business.MERGE] out the `assassination` [type: Life.Die].

In the above examples, there is one gold standard mention while both systems report two event mentions, and they both omit the word "out". According algorithm 3, **System 1** gets full credit while **System 2** gets 0.5. The algorithm is designed this way to prevent a system being penalized again for its span error. However, this make it difficult to find a natural way to combine span scores with attribute scores.

2.6 Combining multiple scores

Algorithm 2 and 3 are limited in that there is no one simple score for final system ranking. Furthermore, the span score only reflects the system's ability to distinguish the 8 types of event mentions from everything else, which is not a useful metric by its own.

A naive way to combine the scores is to multiply these individual scores. However, theoretically, the errors in attribute scoring and the span scoring are not independent, thus it is inappropriate to perform a simple multiplication. We propose a natural adjustment by directly augmenting attribute evaluation into F1 score calculation (Algorithm 4). Line 3 in the algorithm finds a system mention with the highest mapping score that also fits all the attributes of interest as true positive. We can choose the set $\mathcal{A}$ to contain the desired attributes we would like to evaluate on. In our implementation, we iterate all possible attribute combinations and produce all the scores (§4.2).

Algorithm 4 Compute True Postive with Attributes

```
Input: The set of gold standard mentions $\mathcal{G}$; The mapping $M$ indexed by gold standard mentions; Number of system mentions $N_S$; The set $\mathcal{A}$ indexing the attributes that will be evaluated for all mentions
1: $TP \leftarrow 0$
2: for $G \in \mathcal{G}$ do
3:     $S_{max} \leftarrow \arg\max_{Dice}(S, Dice) \in M_G$ Subject to $\mathcal{A}_{S_{max}} = \mathcal{A}_G$
4:     $TP \leftarrow TP + Dice(S_{max}, G)$
Output: $TP$
```

3 Comparison with Previous Methods

3.1 Comparison with MUC

The Message Understanding Conference provides a scoring algorithm for the information extraction task (Chinchor, 1992). Though there is no event mention evaluation, some algorithm design can still be compared with our methods.

The MUC scorer first calculates an alignment between gold standard mention and system, and then counts the number of exact matches COR, the number of partial matches PAR, the number of gold standard keys POS, the number of system responses ACT. The precision and recall are calculated as[6] :

$$P = \frac{COR + 0.5PAR}{POS}; R = \frac{COR + 0.5PAR}{ACT}$$

The MUC scorer then takes the highest F-Score from all possible alignments.

Our method makes several different decisions. First, we use a simple greedy method for choosing an alignment based on span matching instead of trying to find the best alignment.

Second, we give a partial score between 0 to 1 using the Dice Coefficient, while MUC uses a universal partial credit of 0.5. A variable partial score can reflect more subtle differences between systems.

3.2 Comparison with ACE

The Automatic Content Extraction 2005 task included an event related evaluation (NIST, 2005). The Event Mention Detection (VMD) task described in the evaluation guideline defines the event mention as a sentence or phrase. The ACE event task evaluates the systems on the attributes and arguments of a whole event (which may contains multiple event mentions). Such evaluation also requires a system to resolve event coreference. Thus, there is no direct evaluation for event nuggets in ACE 2005.

4 Experiments

We conduct evaluation on the 15 pilot study submissions using the LDC2015E3 dataset, which contains 200 documents with 6921 annotated event mentions. The results we show in this section are all micro average across these mentions.

4.1 Fixing the Precision Formula

The simple fix on precision calculation (§2.4) does not affect the overall trend of the evaluation. The scores of the participant systems only change by a very small value, and the span-based ordering remains the same. We argue that this fix is both more theoretically sound and mathematically pleasing.

4.2 Combining Multiple Scores

As discussed in §2.6, scoring each metric individually will make it difficult to provide one unified score to rank all systems. This can be seen from Figure 1, which plot the evaluation results using the original scoring (sorted on Span F1). In addition, because attribute scores are only calculated on the gold standard mentions, the false alarms on the rest of the predicted mentions are not penalized.

Figure 2 shows the results using multiplicity combination. We observe that the resulting scores will soon become too small after multiplication, which are less interpretable.

Figure 3 presents the results after applying Algorithm 4. The combined score of all attributes now falls into a more reasonable range (bounded by the performance of the hardest attribute, namely realis status). We also observe that all performances decrease monotonically.

We can also use the results from Figure 3 to understand the performance bottleneck of the systems. For example, in system 7, there is a big gap between the mention type F1 score and the span F1. This indicates that the type detection accuracy is low and should be improved. In system 5, the mention span F1 and mention type F1 are very close. Therefore the bottleneck might be in event span identification. This information is not immediately clear from the other figures.

5 Conclusions

In this paper we describe our proposed evaluation metric for event nugget task and identify two problems in evaluation design. We propose solutions to these problems and find out that the new methods produce more interpretable results.

Acknowledgments

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program.

[6] We simplified the discussion by assuming there is no optional gold standard key, which will be removed by the MUC scorer if exists but not aligned

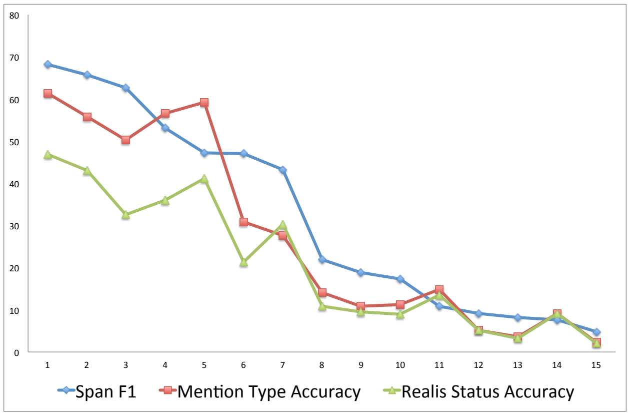

Figure 1: System results sorted by Span F1 score

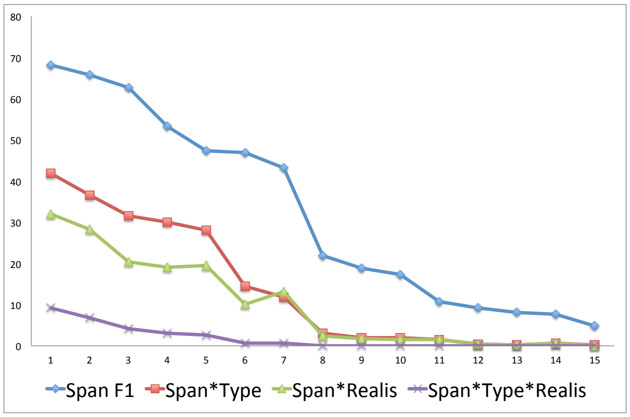

Figure 2: Combining scores with multiplicity (sorted on combined score)

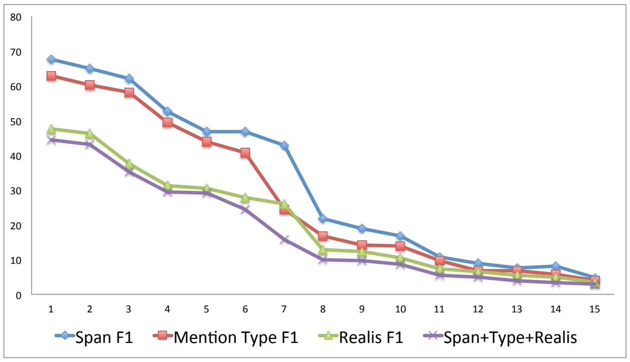

Figure 3: Attribute augmented scoring (sorted on combined score)

References

Cosmin Adrian Bejan, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. 2005. Nonparametric Bayesian Models for Unsupervised Event Coreference Resolution. In Y Bengio, D Schuurmans, J Lafferty, C K I Williams, and A Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1–9.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL '08 Meeting of the Association for Computational Linguistics*, pages 789–797.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610. Association for Computational Linguistics.

Zheng Chen and H Ji. 2009. Graph-based event coreference resolution. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57.

Chen Chen and Vincent Ng. 2013. Chinese Event Coreference Resolution: Understanding the State of the Art. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 822–828.

Nancy Chinchor. 1992. Muc-5 evaluation metric. In *Proceedings of the 5th Conference on Message Understanding*, pages 69–78.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing Information Networks Using One Single Model. In *Proceedings the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP2014)*.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2013. Supervised Within-Document Event Coreference using Information Propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4539–4544, Reykjavik, Iceland. European Language Resources Association (ELRA).

Teruko Mitamura. 2014. TAC KBP event detection annotation guidelines, v1.7. Technical report, Carnegie Mellon University, September.

NIST. 2005. The ACE 2005 (ACE05) Evaluation Plan: Evaluation of the Detection and Recognition of ACE. Technical report, National Institute of Standards and Technology.