Event Mention Detection Scoring

Feb 12, 2015
Language Technologies Institute
Carnegie Mellon University

1 Overall workflow

We show an overall workflow of evaluation for event mention detection in Figure 1.

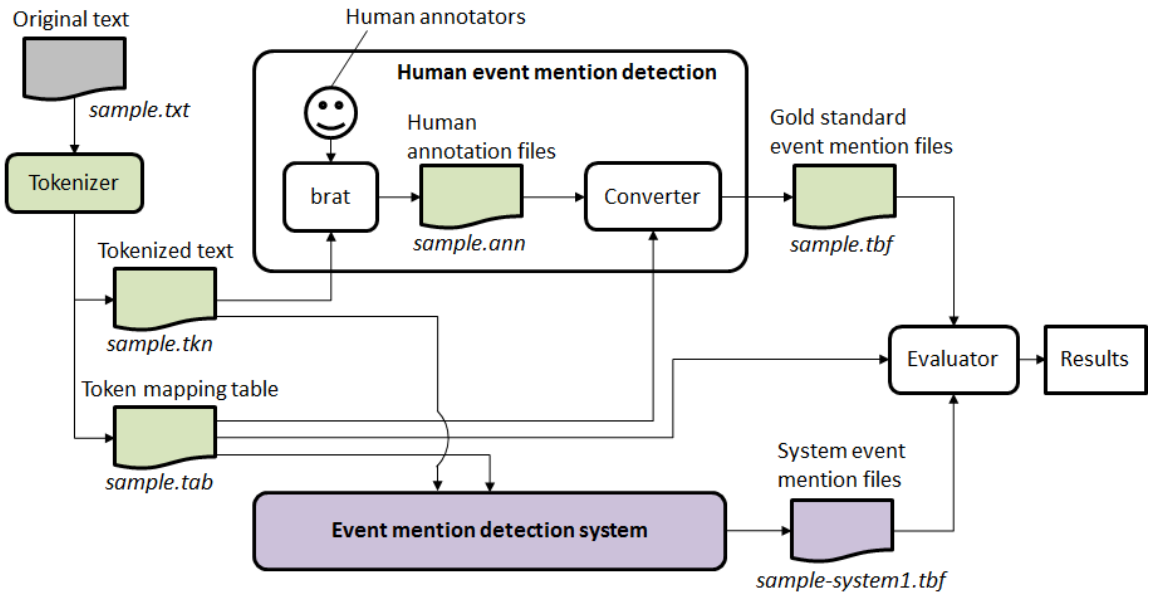

Figure 1: An overall workflow of evaluation for event mention detection.

For each text file, we first tokenize text. The tokenized text is the same input for human annotators and event mention detection systems. Given the tokenized text, human annotators use the **brat** rapid annotation tool[1] to create a gold standard annotation file. We convert the brat annotation file to our evaluation file format. An event mention detection system is given two files as input: (1) the tokenized text and (2) a token mapping table. The latter specifies token ID for every token. The ID information is used in the output of the system. Let us refer to the output of the system as a system event mention file. We require a system event mention file to be given in the same file format as the gold standard file. The evaluator (scorer) takes the gold standard file and a system event mention file as input, and compares them to give a score for the system.

As shown in Figure 1, we carry out tokenization in our evaluation mechanism. We call it **pre-tokenization** since it is done before evaluation. The reasons for pre-

[1] http://brat.nlplab.org/

tokenization are two-fold. First, modern language technologies on English are normally based on tokens, not strings. Mostly, a token is the smallest unit to be used by a complex natural language processing system, and may consist of one or more separated strings. Specifically in the case of event mention detection, if we do not incorporate tokens in evaluation at all, then a system's gaining or missing one correct token ends up with different scores according to the length of the string (i.e., the number of characters in the string). This is unfair from the perspective of evaluation. Therefore, it is necessary to evaluate the performance of event mention detection in terms of tokens.

Second, a modern English tokenizer is basically reliable enough to help human annotators correct (or rethink at least) their event mention annotation, particularly in the relatively unstable process of their creating the gold standard together. This is not only about trivial annotation errors (e.g., misselection of a span with extra whitespace in their web browser) but also about linguistically questionable event mentions.

2 Evaluation Process and Formats

General Process:

1. Run the brat2tokenFormat.py script to convert annotated source files into gold standard format (tbf files), which will be feed into the scorer
2. The participant system should produce their results in the same format of the gold standard format (tbf files).
3. Run the scoring script as described below[2]

Input of Scorer:

1. Gold standard annotation for documents, in format (one line per mention), all annotations are contained in one file only.

2. System output annotation for documents submitted by participants, in format (one line per mention), all annotations are contained in one file only.

3. Tokenization files associated with each document, "tab" extension is appended to the file extension of its corresponding source filename.

Output of Scorer:

1. Overall performance report for system, as described in "Scoring" section.
2. Visualization in text form or html form if the corresponding arguments are specified. (This is not the core functionality for scoring, please refer to the README file of the scorer for details)

Evaluation Script Options:

Please follow the README and help information in the script distribution.

[2] An example shell script is provided together with the evaluation package to run both steps

2.1 Formats

2.1.1 Tokenization file format

LDC will provide tokenized mapping files along with the tokenization files (tkn). We will use tab-delimited mapping files for evaluation purpose. These files map the tokens to their offsets[3] in the tokenized files. "tab" is appended to the file extension of its corresponding source filename. A mapping table contains 4 columns for each row, and the rows contain an ordered listing of the document's tokens. The columns are:

- token_id: A string of "t" followed by a token-number beginning at 0
- token_str: The literal string of a given-token
- tkn_begin: Index of the token's first character in the tkn file
- tkn_end: Index of the token's last character in the tkn file

2.1.2 System and gold standard annotation file format:

1. All event mention annotations for all documents in the corpus are written into one single file
2. A header will indicate the start of a new document (<s> is the space character)
    a. Header := #BeginOfDocument<s><doc ID>
3. A footer will indicate the end of a document
    a. Footer := #EndOfDocument
4. Different event mentions should not include the same token

The event mentions file is a tab-separated file. To be specific, for each mention line, we follow the following format.

2.1.3 Definition of event mention format (one per line):

event-mention := <system ID><TAB><doc ID><TAB><mention ID><TAB><token ID list><TAB><mention><TAB><event-type><TAB><realis status><TAB><score>

Explanation:

<system ID> := the name of the system
<doc ID> := the ID of the input document
<mention ID> := the ID of the mention, which should uniquely identify the mention within the current document
<token ID list> := list of IDs for the token(s) of the current mention, in ascending order, separated by commas (,) . Each ID is a string of "t" followed by a token-number beginning at 0, the same as how they appear in the tokenization files
<mention> := the actual character string of the mention
<event-type> := the ACE hierarchy type
<realis status> := the REALIS label

[3] Offsets are character offsets where the first character is 0

<score> := a score (confidence, etc.) the system wants to assign for the mention span detection, assign 0 if the system does not use a score. This score will not affect the evaluation results
<TAB> := tab character

Example:
system1 sample_document_id E6 t96 talking Contact_Meet Other 1

3 Scoring

3.1 Scoring for one document

We denote a gold standard mention with G, and a system mention with S. Overlap(G,S) is a token-based F1-score function of G and S that returns a score between 0 and 1 (see the OVERLAP subroutine in [Appendix 1: Pseudo-code for scoring one document] for details). All invisible words are already removed from G and S[4].

3.1.1 Create a mapping

To perform scoring for a document, system mentions are mapped to gold standard mentions based on the Overlap score. A system mention is always mapped to one and only one gold standard mention with which the system mention will have the highest overlap score. However, one gold standard mention can be mapped to multiple system mentions.

3.1.2 Scoring mention detection

To score mention detection, a mention-based F1 score is computed in the following way:

1. For each gold standard mention G_i, recall that G_i can be mapped to multiple system mentions, we only choose one system mention S_j that maximize $Overlap\ (G_i, S_j)$, and denote $TP_i = \max Overlap\ (G_i, S_j)$. Let J be a set that contains all such system mentions S_j across the whole document.
2. True Positive = $\sum_i TP_i$
3. False Positive = #System Mention - $|J|$
4. Precision = True Positive / (True Positive + False Positive)
5. Recall = True Positive / #Gold Mention

[4] Invisible words are ignored in scoring. They include: determiners {the, a, an}, pronouns {I, you, he, she, we, my, your, her, our}, relative pronouns {who, what, where, when}.
Note that "it" and "that" and pronouns including {his, ours, mine, yours, ours, they} are not included in the invisibles list because they can occasionally be resolved as nominal event mentions.

6. F1 = H (Precision, Recall), where H is the harmonic average function

3.1.3 Scoring realis status and mention type detection

To score realis status and mention type detection, we use the same mapping:

1. For each gold standard mention G_i, we count the number of system mentions S_j that are mapped to it as N_i,
2. We use X_i_realis and X_i_mention to define the realis status and mention type of mention X_i respectively.
3. Initialize with realis_score = 0; mention_score = 0
4. If G_i_realis = S_j_realis, realis_score = realis_score + $1/N_i$;
5. Similarly, if G_i_mention = S_j_mention, mention_score = mention_score + $1/N_i$;
6. realis_detection_accuracy = realis_score / #GoldStandardMentions
7. type_detection_accuracy = mention_score / #GoldStandardMentions

Examples:

Rule 1: do not accept prepositions but include particles

- [look] up a chimney *vs.* [look up] a dictionary
- [climb] up the ladder
- [take responsibility for]
- sing [all the way] to school
- [go] to school

Rule 2: consider the maximum extent of an event mention, but don't worry about determiners (they are invisible)

- [takes a shower] ==> it is okay for annotators to include "a" in their annotation; we ignore "a" for evaluation
- [make a quick decision] ==> it is okay for annotators to annotate the whole phrase; we ignore "a" and include "quick" in the evaluation

3.2 Summarization score

After all documents are scored, we also report scores that give a summary of performance over the whole corpus by taking the average across documents. We use the standard Micro and Macro average definition, as listed below:

Macro Average Scores (numerical average over the document scores):

Precision_macro = sum of all Precision / #document
Recall_macro = sum of all Recall / #document
F1_macro = 2* Precision_macro * Recall_macro / (Precision_macro + Recall_macro)
Type_detection_accuracy_macro = sum of all type_detection_accuracy / #document
Realis_detection_accuracy_macro = sum of all realis_detection_accuracy / #document

Micro Average Scores (sum of the individual true positives, false positives, and false negatives of each mention to calculate the overall F-Score)

Precision_micro = (sum of TP on all docs)/ (sum of TP on all docs + sum of FP on all docs)
Recall_micro = (sum of TP on all docs) / (total number of gold standard mention in all docs)
F1_micro = 2* Precision_ micro * Recall_ micro / (Precision_ micro + Recall_ micro)
Type_detection_accuracy_micro = sum of num_type_correct / (total number of gold standard mention in all docs)
Realis_detection_accuracy_micro = sum of realis_detection_score / (total number of gold standard mention in all docs)

Appendix 1: Pseudo-code for scoring one document

```
Initialize mappingScores as an empty list.

#STEP 1: Compute overlap scores for each pair of Gold/System Mention

FOR each system mention S := {S_mid, S_tokens, S_realis, S_type} (one per line)
  Let S_mid := mention id of S
  Let S_tokens := token IDs associated with S
  Let S_tokens := S_tokens – {token IDs of invisible words} #See NOTE 1
  Let S_realis := realis status of S
  Let S_type := mention type of S

  FOR each gold mention G:= {G_mid, G_tokens, G_realis, G_type}
    Let G_mid := mention id of G
    Let G_tokens := token IDs associated with G
    Let G_tokens := G_tokens – {token IDs of invisible words}
    Let G_realis := realis status of G
    Let G_type := mention type of G

    Let overlap := OVERLAP(S_tokens, G_tokens)
    IF overlap > 0
       mappingScores := mappingScores + (G, S, overlap)
    END IF
  END FOR
END FOR

#STEP2: After calculating all pairs, we find the best mapping between System
#Mentions and Gold Standard Mentions

Sort mappingScores based on overlap

Initialize Mapping as an empty list to hold mapping records

WHILE mappingScores != {}:
  (G, S, overlap) = mappingScores.pop()    #get the item with the highest overlap

  #if G and S have not been mapped, it means there are no better overlaps

  IF G has not been mapped and S has not been mapped
        THEN Mapping := Mapping + {G,S, overlap}
  ELSE IF G has been mapped but S has not been mapped
        THEN Find the record in Mapping that contains G, append S to record
  END IF
END WHILE
```

```
#Append system score to the gold standard file
FOR each gold mention G:
        Score := Mapping[G].overlap
        append Score to the end of the line of G_mid in Gold Standard, in
        position <score2>
END FOR

#STEP3.1: Compute document level errors and corrects on mention detection
TP := 0
FOR EACH System Mention S
 IF S is contained in Mapping
    TP := TP + Mapping[S].overlap
 ELSE
    FP := FP + 1
 END IF
END FOR

#STEP3.2: Compute document level precision, recall for mention detection:
Precision := TP / (TP+FP)
Recall := TP / #GoldStandardMentions
F1_Score := 2*Precision*Recall/(Precision+Recall)

#STEP3.3: Compute mention and realis type detection score:
type_correct_score := 0
realis_correct_score := 0
FOR EACH LINE (G,{S}, overlap) in Mapping
      Mapping_num:= |{S}|
      Single_score := 1/ Mapping_num
      FOR EACH LINE S in {S}
         IF G_type == S_type
            type_correct_score := type_correct_score + Single_score
         END IF
         IF G_realis == S_realis
            realis_correct_score := realis_correct_score + Single_score
         END IF
      END FOR
END FOR

Type_detection_accuracy := type_correct_score / #GoldStandardMentions
Realis_detection_accuracy:= realis_correct_score  / #GoldStandardMentions

# Return and report the following measures for this document:
Measures for this doc = {TP, FP, type_correct_score, realis_correct_score , Precision,
Recall, F1_Score, Type_detection_accuracy, Realis_detection_accuracy }
```

Subroutine OVERLAP(G,S):

```
 IF G == S, THEN  score := 1.0
 IF G⋀S == {}, THEN score := 0.0
 ELSE
        precision_m := (|S⋀G|)/|S|
        recall_m := (|S⋀G|)/|G|
        score := 2*precision_m*recall_m / (precision_m + recall_m)
  RETURN score
End Subroutine
```

Appendix 2: Example of scoring computation

Sample System output:

System Id	Doc Id	Event Mention Id	Token Id List	Mention Text	Event Type	Realis Status	System Confidence
sue	sample	E1	t17	advice	Communicate	Other	1
sue	sample	E2	t19	reassurance	Communicate	Other	1
sue	sample	E3	t33	came	Transport-Person	Actual	1
sue	sample	E4	t52	going	Transport-Person	Actual	1

Gold annotations:

System Id	Doc Id	Event Mention Id	Token Id List	Mention Text	Event Type	Realis Status	System Confidence
gold	sample	E1	t52	going	Transport-Person	Actual	1
gold	sample	E2	t33	came	Transport-Person	Actual	1
gold	sample	E3	t87	got	Transport-Person	Actual	1
gold	sample	E4	t14,t17,t18, t19	offer advice or reassurance	Communicate	Other	1

In the following tables, the "Event Type" and "Realis Status" are omitted for clarity

3.3 STEP 1: Compute overlap scores for each Gold/System Mention pair

There are no invisible words, so no removal is done

Compute the "mappingScore" table as followed:

Gold Mention	System Mention	Overlap
(E1, [52])	(E4, [52])	1
(E2, [33])	(E3, [33])	1
(E4, [14,17,18,19])	(E1, [17])	2/5 (See #)
(E4, [14,17,18,19])	(E2, [19])	2/5 (Same as above)

Example calculation of overlap:

Prec(G_E4,S_E1) = (|E1 ^ E4|) / |E1| = 1/1 = 1;

Recall(G_E4,S_E1) = (|E1 ^ E4|) / |E4| = ¼ = ¼;

Overlap(G_E4,S_E1) = 2 * Prec(G_E4,S_E1) * Recall(G_E4,S_E1) / (Prec(G_E4,S_E1) + Recall(G_E4,S_E1)) = 2 * 1 * ¼ / (1 + ¼) = 2/5

STEP2: After the calculation of all pairs, we can find the best mapping between System Mention and Gold Standard Mentions

Sort the "mappingScore" table based on overlap (ties are currently broken on their appearance in the data):

Gold Mention	System Mention	Overlap
(E1, [52])	(E4, [52])	1
(E2, [33])	(E3, [33])	1
(E4, [14,17,18,19])	(E1, [17])	2/5
(E4, [14,17,18,19])	(E2, [19])	2/5

We select mappings from the table above from top to bottom:

1. In row1, Select Gold, E1 to map to System, E4, we also record the overlap score = 1
2. In row2, Select Gold, E2 to map to System, E3, we also record the overlap score = 1
3. In row3, Select Gold, E4 to map to System, E1, we record the overlap score = 2/5
4. In row4, Select Gold, E4 to map to System, E2, since Gold E4 has already been mapped to a mention System E1, we do not record an overlap score, but we record the system mention here so we know that E4 is mapped to 2 system mention

We have the following mapping table (mappingScore table):

Gold Mention	System Mention	Overlap
(E1, [52])	(E4, [52])	1
(E2, [33])	(E3, [33])	1
(E4, [14,17,18,19])	(E1, [17]) , (E2,[19])	2/5

STEP3.1: Compute document level errors and corrects

TP is the sum of the overlap in the mappingScore table:

TP = 1 + 1 + 2/5 = 2.4

S{E2} is not contained in the mappingScore table, so

FP = 1

STEP3.2: Compute document level precision, recall:

Precision := TP / (TP+FP) = 2.4 / (2.4+1) = 0. 7059
Recall := TP / #GoldStandardMentions = 2.25/4 = 0.6
F1 := 2*Precision*Recall/ (Precision+Recall) = 2*0. 7059*0.6/ (0. 7059+0. 6) = 0.6487

STEP3.3: Compute mention type and realis status detection score:

For each row in the mapping table, we check whether the system mention(s) has/have the same realis status and mention type as the gold mention.

G_E1 – S_E4 and G_E2 – S_E3 are both one-to-one mappings, so N = 1. Both mention types and realis status are correct, giving type_score = 2, realis_score = 2.

G_E4 is mapped to 2 mentions {S_E1, S_E2}, so N = 2. Both mention types and realis status are correct, giving type_score = ½ + ½ = 1, realis_score = ½ + ½ = 1.

The sum of type score is 2 + 1 = 3, and the total realis score 2+1 = 3. This gives the accuracy as:

Type_detection_accuracy := 3 / #GoldStandardMentions = 0.75
Realis_detection_accuracy:= 3 / #GoldStandardMentions = 0.75

Final Output:

Output1: The score appended gold standard file will be like the following

System Id	Doc Id	Event Mention Id	Token Id List	Mention Text	Event Type	Realis Status	System Confidence	Sue Mention score
gold	sample	E1	t52	going	Transport-Person	Actual	1	1
gold	sample	E2	t33	came	Transport-Person	Actual	1	1
gold	sample	E3	t87	got	Transport-Person	Actual	1	-
gold	sample	E4	t14,t17,t18,t19	offer advice or reassurance	Communicate	Other	1	0.4

Output2: Individual document performance and averaged performance

We only take one document as example, which make the micro and macro measures to be the same.

```
========Document results==========
TP      FP      #Gold   Prec    Recall  F1      Type    Realis  Doc Id
2.40    1.00    4       0.7059  0.6000  0.6486  0.7500  0.7500  1b268b27094ba9c5feb11192dad940ab

========Final Results=========
Precision (Micro Average): 0.7059
Recall (Micro Average):0.6000
F1 (Micro Average):0.6486
Mention type detection accuracy (Micro Average):0.7500
Mention realis status accuracy (Micro Average):0.7500
Precision (Macro Average): 0.7059
Recall (Macro Average): 0.6000
F1 (Macro Average): 0.6486
Mention type detection accuracy (Macro Average):0.7500
Mention realis status accuracy (Macro Average):0.7500
```