**Event Mention Detection scoring**

**Overall workflow**

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

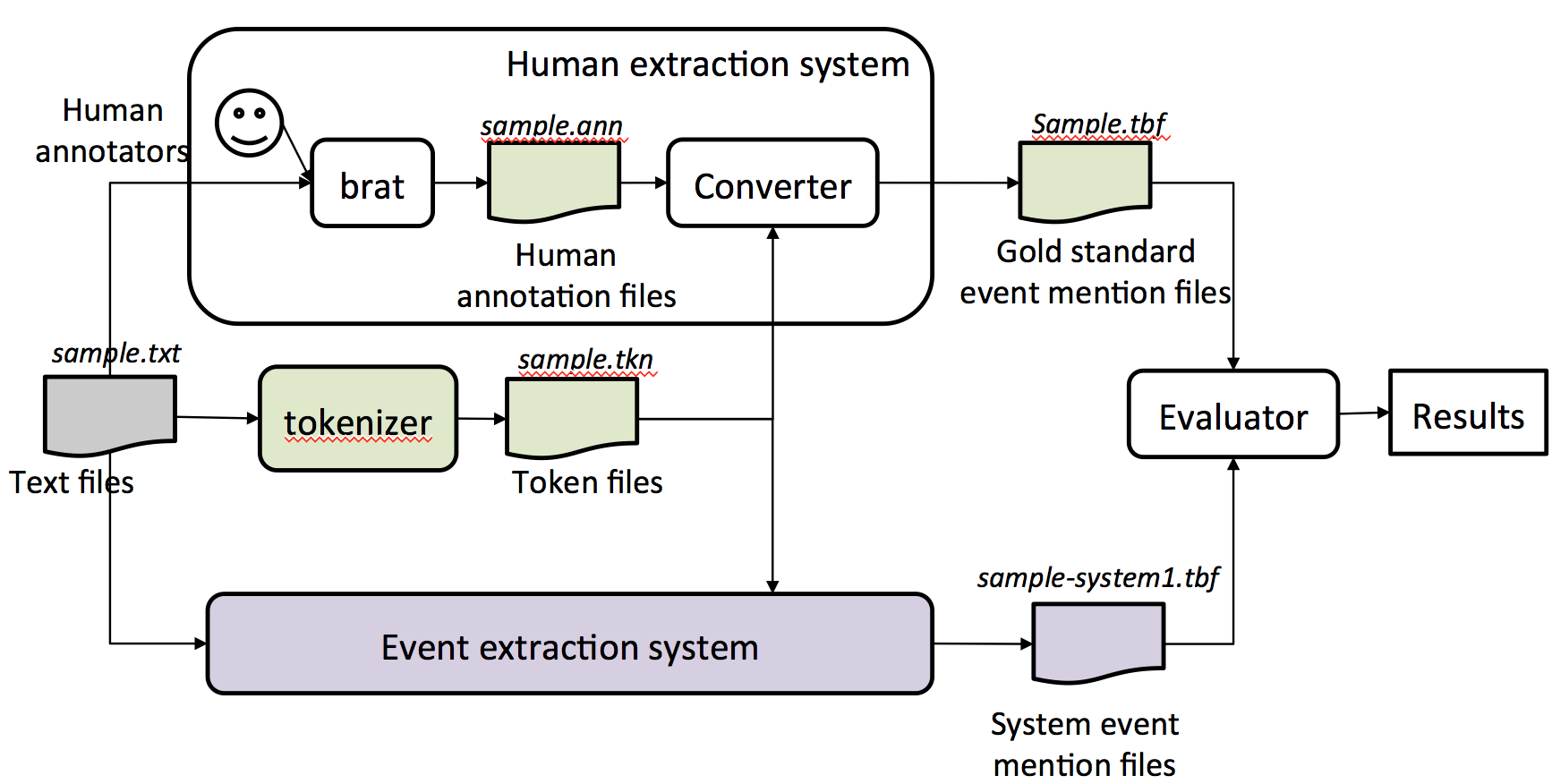


Figure 1: An overall workflow of event mention evaluation.

For each text file, human annotators use the **brat** rapid annotation tool to create a gold standard annotation file. We convert a brat annotation file to our evaluation file format. We assume event extraction systems output system event mentions in the same file format. The evaluator (scorer) then reads the output of event mention detection systems and compares them to the gold standard.

The scorer reads the output of event mention detection systems and compares them to the gold standard.

**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, one file per document.

**Output of Scorer:**

1. System output annotation as item 2 in Input, with addition of a score for each mention appended to each line.

2. Overall performance report for system, as described in “Scoring” section.

**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
   1. Header := #BeginOfDocument<s><doc ID>
3. A footer will indicate the end of a document
   1. Footer := #EndOfDocument
4. Different event mentions should not include the same token

For each mention line, we follow the following format,

**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><score1><TAB> <score2><TAB><score3>

**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 (,)

<mention> := the actual character string of the mention

<event-type> := the ACE hierarchy type

<realis status> := the REALIS label

<score1> := any score (confidence, etc.) the system wants to assign (ignored)

<score2> := score assigned in the evaluation

<score3> := additional possible score assigned by human

<TAB> := tab character

**Pseudo-code for Scoring:**

Let mappingScores = {}

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

FOR each system mention S := {S\_mid, S\_tokens} (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**

FOR each gold mention G:= {G\_mid, G\_tokens}

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 overlap := OVERLAP(S\_tokens, G\_tokens) **#See NOTE2**

IF overlap > 0

mappingScores := mappingScores + (G, S, overlap)

END IF

END FOR

END FOR

#STEP2: After the calculation of all pairs, we can find the best mapping between

#System Mention and Gold Standard Mentions

Sort mappingScores based on overlap

Mapping = {} # create a empty mapping set to hold mappings

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 overlap than this one

IF G has not been mapped and S has not been mapped

THEN Mapping := Mapping + {G,S, overlap}

END WHILE

#Append system score to the gold standard file

FOR each gold mention G:= {G\_mid, G\_tokens}:

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

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:

Precision := TP / (TP+FP)

Recall := TP / #GoldStandardMentions

**Subroutine OVERLAP1(G,S): #See NOTE2**

IF G == S, THEN score := 1.0

IF G∧S == {}, THEN score := 0.0

ELSE

IF |G| > |S|, THEN score := (|S∧G|)/|G|

IF |G| < |S|, THEN score := (|S∧G|)/|S|

RETURN score

End Subroutine

**Subroutine OVERLAP2(G,S): #See NOTE2**

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

**Note 1**: Invisible words are ignored in scoring. They include: determiners {the, a, an}, pronouns {I, you, he, she, we, they, his, her, my, your, mine, yours, our, ours}, relative pronouns {who, what, where, when}, …?

Note that “it” and “that” are removed from the list because they can occasionally be resolved as nominal event mentions.

**Note 2**: Overlap can use either Overlap1 or Overlap2, which are alternative methods to calculate the overlap between system and gold standard mentions. We call this mention-level score.

* Overlap1 computes simple accuracy and normalizes by dividing the intersection of G and S mentions by their maximum.
* Overlap2 computes traditional Recall and Precision, and combines them into an F-score for each mention (mention level average).
* Overlap2 actually provides an upper bound of the score computed by Overlap1, because Overlap1 is always smaller than either Precision or Recall.
* Both Overlap1 and Overlap2 calculate a score for each mention separately, and then require a subsequent combination of these scores for a document level F-score of the document as a whole.
* Mapping between gold standard and system mentions are based on the overlap score calculated, however, in normal cases these two overlap gives identical mapping.

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 can ignore "a" in evaluation
* [make a quick decision] ==> it is okay for annotators to annotate the whole phrase; we can ignore "a" and include "quick" in evaluation

# Appendix: 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 | 17 | advice | Communicate | Other | 1 |
| sue | sample | E2 | 19 | reassurance | Communicate | Other | 1 |
| sue | sample | E3 | 33 | came | Transport-Person | Actual | 1 |
| sue | sample | E4 | 52 | 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 | 52 | going | Transport-Person | Actual | 1 |
| gold | sample | E2 | 33 | came | Transport-Person | Actual | 1 |
| gold | sample | E3 | 87 | got | Transport-Person | Actual | 1 |
| gold | sample | E4 | 14,17,18,19 | offer advice or reassurance | Communicate | Other | 1 |

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

There are no invisible words, so no removal will be done

Compute the “mappingScore” table as followed:

|  |  |  |  |
| --- | --- | --- | --- |
| Gold Mention | System Mention | Overlap1 | Overlap2 |
| (E1, [52]) | (E4, [52]) | 1 | 1 |
| (E2, [33]) | (E3, [33]) | 1 | 1 |
| (E4, [14,17,18,19]) | (E1, [17]) | ¼ (See #1) | 2/5 (See #2) |

#1 : Overlap\_1(G\_E4,S\_E1) = (|E1 ^ E4|) / max(|E1| + |E4|) = 1/ max(1,4) = ¼

#2:

Prec(G\_E4,S\_E1) = (|E1 ^ E4|) / |E1| = 1/1 = 1;

Recall(G\_E4,S\_E1) = (|E1 ^ E4|) / |E4| = ¼ = ¼;

Overlap\_2(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

Let’s use overlap 1 as an example

Sort the “mappingScore” table based on overlap 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Gold Mention | System Mention | Overlap1 | Overlap2 |
| (E1, [52]) | (E4, [52]) | 1 | 1 |
| (E2, [33]) | (E3, [33]) | 1 | 1 |
| (E4, [14,17,18,19]) | (E1, [17]) | ¼ (See #1) | 2/5 (See #2) |

## STEP3.1: Compute document level errors and corrects

S{E1,E3,E4} are contained in the mappingScore table, so

TP = mappingScore[S\_E1].Overlap + mappingScore[S\_E3].Overlap + mappingScore[S\_E4].Overlap = 1 + 1 + 1/4 = 2.25

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

FP = 1

## STEP3.2: Compute document level precision, recall:

Precision := TP / (TP+FP) = 2.25 / (2.25+1) = 0. 6923

Recall := TP / #GoldStandardMentions = 2.25/4 = 0.5625

F1 := 2\*Precision\*Recall/ (Precision+Recall) = 2\*0. 6923\*0.5625/ (0.6923+0. 5625) = 0.6207

In addition, 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 | Overlap |
| gold | sample | E1 | 52 | going | Transport-Person | Actual | 1 | 1 |
| gold | sample | E2 | 33 | came | Transport-Person | Actual | 1 | 1 |
| gold | sample | E3 | 87 | got | Transport-Person | Actual | 1 | - |
| gold | sample | E4 | 14,17,18,19 | offer advice or reassurance | Communicate | Other | 1 | 0.25 |