Inter-Annotator Agreement for ERE Annotation

Seth Kulick and **Ann Bies** and **Justin Mott**
Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA 19104
{skulick,bies,jmott}@ldc.upenn.edu

Abstract

This paper describes a system for inter-annotator agreement analysis of ERE annotation, focusing on entity mentions and how the higher-order annotations such as EVENTS are dependent on those entity mentions. The goal of this approach is to provide both (1) quantitative scores for the various levels of annotation, and (2) information about the types of annotation inconsistencies that might exist. While primarily designed for inter-annotator agreement, it can also be considered a system for evaluation of ERE annotation.

1 Introduction

In this paper we describe a system for analyzing dually human-annotated files of Entities, Relations, and Events (ERE) annotation for consistency between the two files. This is an important aspect of training new annotators, to evaluate the consistency of their annotation with a "gold" file, or to evaluate the agreement between two annotators. We refer to both cases here as the task of "inter-annotator agreement" (IAA).

The light ERE annotation task was defined as part of the DARPA DEFT program (LDC, 2014a; LDC, 2014b; LDC, 2014c) as a simpler version of tasks like ACE (Doddington et al., 2004) to allow quick annotation of a simplified ontology of entities, relations, and events, along with identity coreference. The ENTITIES consist of coreferenced entity mentions, which refer to a span of text in the source file. The entity mentions are also used as part of the annotation of RELATIONS and EVENTS, as a stand in for the whole ENTITY.

The ACE program had a scoring metric described in (Doddington et al., 2004). However, our emphasis for IAA evaluation is somewhat different than that of scoring annotation files for accuracy with regard to a gold standard. The IAA system aims to produce output to help an annotation manager understand the sorts of errors occurring, and the general range of possible problems. Nevertheless, the approach to IAA evaluation described here can be used for scoring as well. This approach is inspired by the IAA work for treebanks in Kulick et al. (2013).

Because the entity mentions in ERE are the fundamental units used for the ENTITY, EVENT and RELATION annotations, they are also the fundamental units upon which the IAA evaluation is based. The description of the system therefore begins with a focus on the evaluation of the consistency of the entity mention annotations. We derive a mapping between the entity mentions between the two files (henceforth called File A and File B). We then move on to ENTITIES, RELATIONS, and EVENTS, pointing out the differences between them for purposes of evaluation, but also their similarities.[1]

This is a first towards a more accurate use of the full ENTITIES in the comparison and scoring of ENTITIES and EVENTS annotations. Work to expand in this direction is in progress. When a more complete system is in place it will be more appropriate to report corpus-based results.

2 Entity Mentions

There are two main aspects to the system's handling of entity mentions. First we describe the mapping of entity mentions between the two annotators. As in Doddington et al. (2004), the possibility of overlapping mentions can make this a complex problem. Second, we describe how our system's output categorizes possible errors.

[1] This short paper focuses on the design of the IAA system, rather than reporting on the results for a specific dataset. The IAA system has been run on dually annotated ERE data, however, which was the source for the examples in this paper.

Proceedings of the 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 21–25,
Baltimore, Maryland, USA, June 22-27, 2014. ©2014 Association for Computational Linguistics

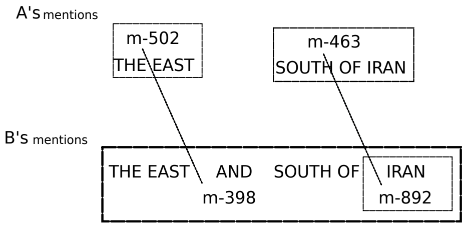

Figure 1: Case of ambiguous Entity Mention mapping disambiguated by another unambiguous mapping

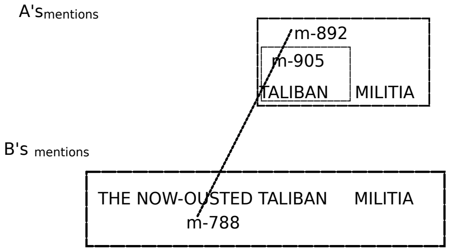

Figure 2: Case of Entity Mention mapping resolved by maximum overlap

2.1 Mapping

As mentioned in the introduction, our system derives a mapping between the entity mentions in Files A and B, as the basis for all further evaluation of the ERE annotations. Entity mentions in Files A and B which have exactly the same location (offset and length) are trivially mapped to each other. We refer to these as "exact" matches.

The remaining cases fall into two categories. One is the case of when an entity mention in one file overlaps with one and only one entity mention in the other file. We refer to these as the "unambiguous" overlapping matches. It is also possible for an entity mention in one file to overlap with more than one entity mention in the other file. We refer to these as the "ambiguous" overlapping matches, and these patterns can get quite complex if multiple ambiguous overlapping matches are involved.

2.1.1 Disambiguation by separate unambiguous mapping

Here an ambiguous overlapping is disambiguated by the presence of an unambiguous mapping, and the choice for mapping the ambiguous case is decided by the desire to maximize the number of mapped entity mentions.

Figure 1 shows such a case. File A has two entity mentions annotations (m-502 and m-463) and File B has two entity mention annotations (m-398 and m-892). These all refer to the same span of text, so m-502 (THE EAST) and m-463 (SOUTH OF IRAN) both overlap with m-398 in File B (THE EAST AND SOUTH OF IRAN). m-463 in addition overlaps with m-892 (IRAN).

We approach the mapping from the perspective of File A. If we assign the mapping for m-463 to be m-398, it will leave m-502 without a match, since m-398 will already be used in the mapping. Therefore, we assign m-502 and m-398 to map to each other, while m-463 and m-892 are mapped to each other. The goal is to match as many mentions as possible, which this accomplishes.

2.1.2 Disambiguation by maximum overlap

The other case is shown in Figure 2. Here there are two mentions in File A, m-892 (TALIBAN MILITIA) and m-905 (TALIBAN), both overlapping with one mention in File B, m-788 (THE NOW-OUSTED TALIBAN MILITIA), so it is not possible to have a matching of all the mentions. We choose the mapping with greatest overlap, in terms of characters, and so m-892 and m-788 are taken to match, while m-905 is left without a match.

For such cases of disambiguation by maximum overlap, it may be possible that a different matching, the one with less overlap, might be a better fit for one of the higher levels of annotation. This issue will be resolved in the future by using ENTITIES rather than ENTITY MENTIONS as the units to compare for the RELATION and EVENT levels.

2.2 Categorization of annotation inconsistencies

Our system produces an entity mention report that lists the number of exact matches, the number of overlap matches, and for Files A and B how many entity mentions each had that did not have a corresponding match in the other annotator's file.

Entity mentions can overlap in different ways, some of which are more "serious" than other. We categorize each overlapping entity mention based on the nature of the edge differences in the non-exact match, such as the presence or absence of a determiner or punctuation, or other material.

In addition, both exact and overlap mentions can match based on location, but be different as far as the entity mention level (NAMed, NOMinal, and PROnominal). The software also outputs all such mismatches for each match.

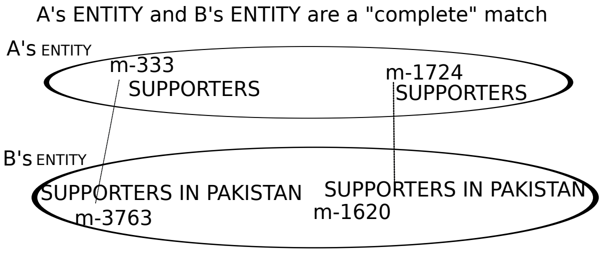

Figure 3: Complete match between File A and File B ENTITIES despite overlapping mentions

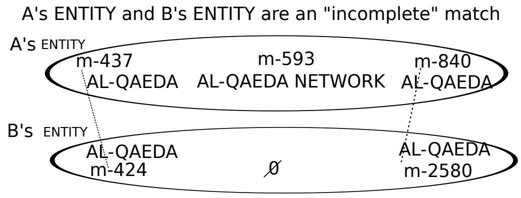

Figure 4: Incomplete match between File A and File B ENTITIES, because File B does not have a mention corresponding to m-593 in File A

3 Entities

An ENTITY is a group of coreferenced entity mentions. We use the entity mention mapping discussed in Section 2 to categorize matches between the ENTITIES as follows:

Complete match: This means that for some ENTITY x in File A and ENTITY y in File B, there is a 1-1 correspondence between the mentions of these two ENTITIES. For purposes of this categorization, we do not distinguish between exact and overlap mapping but include both as corresponding mention instances, because this distinction was already reported as part of the mention mapping.

Figure 3 shows an example of a complete match. File A has two mentions, m-333 (SUPPORTERS) and m-1724 (another instance of SUPPORTERS). These are co-referenced together to form a single ENTITY. In File B there are two mentions, m-3763 (SUPPORTERS IN PAKISTAN) an m-1620 (another instance of SUPPORTERS IN PAKISTAN). It was determined by the algorithm for entity mention mapping in Section 2.1 that m-333 and m-3763 are mapped to each other, as are m-1724 and m-1620, although each pair of mentions is an overlapping match, not an exact match. At the ENTITY level of coreferences mentions, there is a 1-1 mapping between the mentions of A's ENTITY and B's ENTITY. Therefore these two ENTITIES are categorized as having a complete mapping between them.

Incomplete match: This means that for some ENTITY x in file A and ENTITY y in file B, there may be some mentions that are part of x in A that have no match in File B, but all the mentions that are part of x map to mentions that are part of ENTITY y in File B, and vice-versa. Figure 4 shows an example of an incomplete match. File A has three entity mentions, m-437 (AL-QAEDA), m-593 (AL-QAEDA NETWORK), and m-840 (AL-QAEDA again), coreferenced together as a single ENTITY. File B has two entity mentions, m-424 (AL-QAEDA) and m-2580 (AL-QAEDA again), coreferenced together as a single ENTITY. While m-437 maps to m-424 and m-840 maps to m-2580, m-593 does not have a match in File B, causing this to be categorized as an incomplete match.

No match: It is possible that some ENTITIES may not map to an ENTITY in the other file, if the conditions for neither type of match exist. For example, if in Figure 4 m-593 mapped to a mention in File B that was part of a different ENTITY than m-424 and m-2580, then there would not be even an incomplete match between the two ENTITIES.

Similar to the mentions, ENTITIES as a whole can match as complete or incomplete, but still differ on the entity type (ORGanization, PERson, etc.). We output such type mismatches as separate information for the ENTITY matching.

4 Relations

A RELATION is defined as having:

1) Two RELATION arguments, each of which is an ENTITY.
2) An optional "trigger", a span of text.
3) A type and subtype. (e.g., "Physical.Located")

For this preliminary stage of the system, we match RELATIONS in a similar way as we do the ENTITIES, by matching the corresponding entity mentions, as stand-ins for the ENTITY arguments for the RELATION. We use the previously-established mapping of mentions as basis of the RELATION mapping.[2]

We report four types of RELATION matching:[3]
1) exact match - This is the same as the complete

[2] This is a stricter mapping requirement than is ultimately necessary, and future work will adjust the basis of RELATION mapping to be full ENTITIES.

[3] Because of space reasons and because RELATIONS are so similar to EVENTS, we do not show here an illustration of RELATION mapping.

match for ENTITIES, except in addition checking for a trigger match and type/subtype.
2) types different - a match for the arguments, although the type or subtypes of the RELATIONS do not match. (The triggers may or may not be different for this case.)
3) triggers different - a match for the arguments and type/subtype, although with different triggers.
4) no match - the arguments for a RELATION in one file do not map to arguments for any one single RELATION in the other file.

5 Events

The structure of an EVENT is similar to that of a RELATION. Its components are:

1) One or more EVENT arguments. Each EVENT argument is an ENTITY or a date.
2) An obligatory trigger argument.
3) A type and subtype (e.g., "Life.MARRY")

In contrast to RELATIONS, the trigger argument is obligatory. There must be at least one ENTITY argument (or a date argument) in order for the EVENT to qualify for annotation, although it does not need to be exactly two, as with RELATIONS.

The mapping between EVENTS works essentially as for ENTITIES and RELATIONS, once again based on the already-established mapping of the entity mentions.[4] There are two slight twists, however. It is possible for the only EVENT argument to be a date, which is not an entity mention, and so we must also establish a mapping for EVENT date arguments, as we did for the entity mentions. Because the trigger is obligatory, we treat it with the same level of importance as the arguments, and establish a mapping between EVENT triggers as well. We report three types of EVENT matching:[5]

1) exact match - all arguments match, as does the trigger, as well as the type/subtype.
2) types different - a match for the arguments and trigger, although the type or subtypes of the EVENTS do not match.
3) no match - either the arguments for a EVENT in one file do not map to arguments for any one single EVENT in the other file, or the triggers do not map.

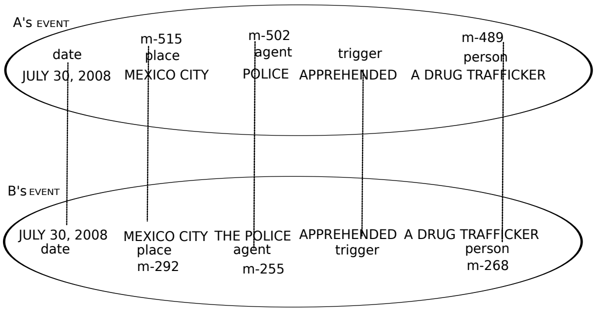

Figure 5: EVENT match

Figure 5 shows an example of an exact match for two EVENTS, one each in File A and B. All of the arguments in one EVENT map to an argument in the other EVENT, as does the trigger. Note that the argument m-502 (an entity mention, POLICE) in File A maps to argument m-255 (an entity mention, THE POLICE) in File B as an overlap match, although the EVENTS are considered an exact match.

6 Future work

We did these comparisons based on the lowest entity mention level in order to develop a preliminary system. However, the arguments for EVENTS and RELATIONS are ENTITIES, not entity mentions, and the system be adjusted to do the correct comparison. Work to adjust the system in this direction is in progress. When the full system is in place in this way, we will report results as well. In future work we will be developing a quantitative scoring metric based on the work described here.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory and Defense Advance Research Projects Agency under agreement number FA8750-13-2-0045. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory and Defense Advanced Research Projects Agency or the U.S. Government.

[4] As with relations, this is a stricter mapping than necessary, and future work will adjust to use ENTITIES as EVENT arguments.

[5] Currently, if an EVENT argument does not map to any mention in the other file, we consider the EVENT to be a "no match". In the future we will modify this (and likewise for RELATIONS) to be more forgiving, along the lines of the "incomplete match" for ENTITIES.

References

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. Automatic content extraction (ACE) program - task definitions and performance measures. In *LREC 2004: 4th International Conference on Language Resources and Evaluation*.

Seth Kulick, Ann Bies, Justin Mott, Mohamed Maamouri, Beatrice Santorini, and Anthony Kroch. 2013. Using derivation trees for informative treebank inter-annotator agreement evaluation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 550–555, Atlanta, Georgia, June. Association for Computational Linguistics.

LDC. 2014a. DEFT ERE Annotation Guidelines: Entities v1.6. Technical report, Linguistic Data Consortium.

LDC. 2014b. DEFT ERE Annotation Guidelines: Events v1.3. Technical report, Linguistic Data Consortium.

LDC. 2014c. DEFT ERE Annotation Guidelines: Relations v1.3. Technical report, Linguistic Data Consortium.