AMIA Annu Symp Proc 2014:1596.

Facilitating Reconciliation of Inter-Annotator Disagreements

Johann Stan, PhD, Dina Demner-Fushman, MD, PhD, Kin Wah Fung, MD, MS, Olivier Bodenreider, MD, PhD,
Lister Hill National Center for Biomedical Communications, U.S. National Library of Medicine, National Institutes of Health, DHHS, Bethesda, MD

Abstract

Development and evaluation of Natural Language Processing methods often requires text annotation. To gauge the difficulty of the task and increase the reliability and quality of annotations, researchers often recruit at least two annotators. The discrepancies in annotations by multiple annotators need to be identified and reconciled. We present a tool that identifies and helps reconciling and validating annotations in a widely used annotation tool Brat.

Introduction

In the process of annotating a corpus of drug package inserts for drug-drug interactions, we were faced with a problem of reconciling differences in fairly complex annotations of interactions between drugs, drug classes and substances. Our goal was to annotate interactions for training supervised machine learning (ML) algorithms and evaluating the results. Annotated corpora are most useful for training ML tools if they are consistent. To ensure consistency, two experts annotated the interactions and two senior annotators adjudicated the disagreements. To facilitate annotation, we used Brat[1] that is fairly convenient for annotation, but does not provide mechanisms for reconciliation of disagreements. Therefore we have developed functions that allow reconciling disagreements and ensure consistency of annotation across similar interactions mentioned multiple times in the drug package inserts.

Methods

The *Disagreement Reconciliation module* compares annotations by two annotators and uses line numbers in the files loaded to Brat for annotation to indicate the location of disagreements. The following details about the disagreements are printed to a text file: 1) absence of an annotation, for example, *"119 MISSING FROM FILE 1 Type: Drug Span: Amplodipine"* indicates that the first annotator did not highlight Amlodipine as drug in the 119th line of the package insert; 2) different labels assigned to the same span, for example, if one of the annotators labeled *alcohol* as substance, and the other one as drug. The *Sentence Validation module* identifies similar sentences and checks if they have been annotated consistently. Sometimes different package inserts for drugs in the same drug class or different sections of an insert contain almost identical sentences. These sentences can be used to quantify intra-annotator agreement over the entire annotation process, as well as consistency in reconciling disagreements. Sentence similarity was computed using an implementation of the *Jaccard* similarity measure (threshold 0.75). Annotated biomedical entities as well as those identified by Metamap were replaced with standard names, e.g. "drug" in order to capture similar syntactic constructions used for different representatives of drug classes with similar pharmacodynamic and pharmacokinetic properties.

Results

We tested the annotation reconciliation tools on 176 manually annotated package inserts (8081 biomedical entities and 4841 interactions). The tools identified 2584 discrepancies, of which 1200 were in entity annotation and 1384 in interaction annotations. 320 similar sentences were annotated inconsistently (e.g. different types attributed to same drugs or different interaction types).

Conclusion

The tools helped us improve annotation guidelines and detect and reconcile discrepancies. To the best of our knowledge, this is the first publicly available extension for assisting with discrepancies and assuring consistency of annotations using Brat. The toolkit can be downloaded from http://lhce-brat.nlm.nih.gov/disagreementAnalyzer.htm

References

1. Stenetorp P, Topic G, Ohta T, Ananiadou S, Tsujii J. Brat: a Web-based Tool for NLP-Assisted Text Annotation, In Proceedings of the Demonstrations Session at EACL 2012, 2012.