Annotating Discourse Relations with The PDTB Annotator

Alan Lee[1], **Rashmi Prasad**[2], **Bonnie Webber**[3], **Aravind Joshi**[1]
[1]Department of Computer and Information Science, University of Pennsylvania
{aleewk,joshi}@seas.upenn.edu
[2]Department of Health Informatics and Administration, University of Wisconsin-Milwaukee
prasadr@uwm.edu
[3]School of Informatics, University of Edinburgh
Bonnie.Webber@ed.ac.uk

Abstract

The PDTB Annotator is a tool for annotating and adjudicating discourse relations based on the annotation framework of the Penn Discourse TreeBank (PDTB). This demo describes the benefits of using the PDTB Annotator, gives an overview of the PDTB Framework and discusses the tool's features, setup requirements and how it can also be used for adjudication.

1 Introduction

In recent years, discourse relations have become a topic of some interest and there has in effect been a rise in the number of corpora annotated for discourse relations. Following the release of the Penn Discourse TreeBank (PDTB) in 2008 (Prasad et al., 2008), a number of comparable corpora have since adapted the PDTB framework (Prasad et al., 2014), including the Hindi Discourse Relation Bank (Oza et al., 2009), the Leeds Arabic Discourse TreeBank (Al-Saif and Markert, 2010), the Biomedical Discourse Relation Bank (Prasad et al., 2011), the Chinese Discourse TreeBank (Zhou and Xue, 2012), the Turkish Discourse Bank (Zeyrek et al., 2013), the discourse layer of the Prague Dependency Treebank 3.0 (Bejček et al, 2013) and the TED-Multilingual Discourse Bank (TED-MDB) (Zeyrek et al., 2016).

Groups starting new discourse annotation projects have sought an openly available resource to support their work. To address this for annotation in the PDTB framework, we have packaged an updated version of our annotation tool - the PDTB Annotator - for use by the research community. Some of the potential benefits of using the PDTB Annotator include the following: i) the tool is Java-based and therefore works on a number of platforms; ii) it requires little external setup or preprocessing, minimally a set of (Unicode-encoded) text files; iii) with the use of Unicode text files, the tool works with a variety of writing systems and caters for a wide number of languages; iv) it lets a project define its own sense hierarchy; v) it doubles up as an adjudication tool - a pair of annotated file sets may be combined into an "adjudicator view" and the resulting adjudications saved into gold files; vi) each annotation is stored as a simple pipe-delimited text entry, allowing for easy retrieval or processing.

We briefly describe the PDTB annotation framework (Section 2), discuss existing annotation tools (Section 3), give a tour of the PDTB Annotator (Section 4), explain the kind of set-up and configuration for getting started (Section 5) and show how the tool can also be used for adjudication (Section 6).

2 The PDTB Annotation Framework

The PDTB follows a lexically-grounded approach for annotating discourse relations. Discourse relations can be realized explicitly in the text by *discourse connectives*. For example, the **Result** relation in (1) is annotated by marking the discourse connective *as a result* as the expression of the *Explicit* relation.

(1) *Despite the economic slowdown, there are few clear signs that growth is coming to a halt.* As a result, **Fed officials may be divided over whether to ease credit**. (0072)

This work has been supported by the National Science Foundation under grants RI 1422186 and RI 1421067.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 121–125, Osaka, Japan, December 11-17 2016.

All relations are taken to have two arguments - Arg1 (shown in italics) and Arg2 (in bold). As per the revised argument-naming conventions in recent ongoing work on PDTB enrichment (Webber et al., 2016), the Arg2 in syntactically coordinated relations follows (i.e. is to the right of) Arg1, while the Arg2 in syntactically subordinated relations is (syntactically) subordinate to Arg1, regardless of textual order.

Discourse relations are not always realized as Explicit connectives. In such cases, a connective is left to be inferred by the annotator, who lexically encodes this inferred relation. This is shown in (2), where a **Reason** relation between the two adjacent sentences is annotated with *because* as the *Implicit* connective:

(2) *Also unlike Mr. Ruder, Mr. Breeden appears to be in a position to get somewhere with his agenda*. Implicit=because, **As a former White House [...], he is savvy in the ways of Washington.**. (0955)

Aside from Explicit vs Implicit relations, the PDTB framework allows for two other types of relations: *AltLex* for cases where the insertion of an Implicit connective to express an inferred relation leads to a redundancy due to the relation being alternatively lexicalized by some non-connective expression; *EntRel* for cases where only an entity-based coherence relation could be perceived between the sentences. A *NoRel* type is allowed for cases where no discourse relation or entity-based relation could be perceived between the sentences (Prasad et al., 2008).

Senses are annotated for Explicit, Implicit and AltLex relations. An annotator can also infer more than one sense between two arguments of a discourse relation. The tagset of senses is organized hierarchically into three levels. (See (Webber et al., 2016) for the latest PDTB sense hierarchy, which contains a number of refinements and improvements over the version used in PDTB 2.0.) Level 1, which contains four classes - **Temporal**, **Contingency**, **Comparison** and **Expansion**; a Level 2 subclass which further subcategorizes the Level 1 classes, and a Level 3 type, which conveys information about the *directionality* of Level 2 relations which are asymmetric. As an example, conditional relations are encoded as **Contingency** at Level 1, **Condition** at Level 2 and then either **Arg1-as-cond** (3) or **Arg2-as-cond** (4) at Level 3, depending on which argument of the relation serves as the antecedent of the conditional:

(3) *Call Jim Wright's office in downtown Fort Worth, Texas, these days* and **the receptionist still answers the phone "Speaker Wright's office**.

(4) *Insurance companies will offer a good rate* if **no one is sick**

The PDTB framework does not seek to establish links between discourse relations and makes no assumptions regarding higher-level discourse structures (e.g. as trees or graphs). Corpora annotated in the framework present a shallow representation of discourse structure and are well-suited as training material for the task of shallow discourse parsing (Xue et al., 2015).

3 Existing Annotation Tools

There does not exist at present a suitable tool for the annotation of discourse relations according to the PDTB framework. There are tools for annotating relations in the framework of Rhetorical Structure Theory (Mann and Thompson, 1988), like the ISI RST Annotation Tool (Marcu, n.d.), but these tools follow a different theoretical framework and a different set of assumptions - pre-segmentation of the text is required and all relations must be recursively structured into a single hierarchical tree. More recently, the Tree Editor (TrEd) for the Prague Dependency Treebank (PDT) (Bejček et al, 2013) was extended to allow for the annotation of discourse relations (Mírovský et. al., 2015) and was indeed used for developing the discourse layer of the PDT. However, while the discourse annotation in the PDT is inspired by the lexicalized approach of the PDTB, the discourse layer is overlaid on top of the existing tectogrammatical layer and does not stand off from the raw text.

There are more general-purpose text annotation tools which might conceivably be adapted for PDTB-style annotations, provided they allow for the free annotation of segments of text and then for customized linkings between these elements (e.g. MMAX2 (Müller and Strube, 2006), PALinkA (Orăsan, 2003)). However, general-purpose tools understandably require considerable customization and their output representations, typically in XML, often require more technical post-processing before in-depth analysis can

proceed. In our experience, many annotation projects using the PDTB framework start off as pilots or prototypes with quick turnaround time requirements and cannot afford the disproportionate effort needed to customize complex multi-purpose tools.

4 The PDTB Annotator: A Brief Tour

The PDTB Annotator is a Java-based tool released as a runnable jar file and has been successfully used by Mac, Windows and Linux-based users running at least the 1.6 version of the Java Runtime Environment. The jar file is used in conjunction with a preconfigured file (called Options.cfg) which controls the sense tags as well as Implicit connectives available to the annotator (see Section 5).

The main window of the PDTB Annotator contains three sub-panels, as discussed below and shown in respective left-to-right order in Fig. 1.

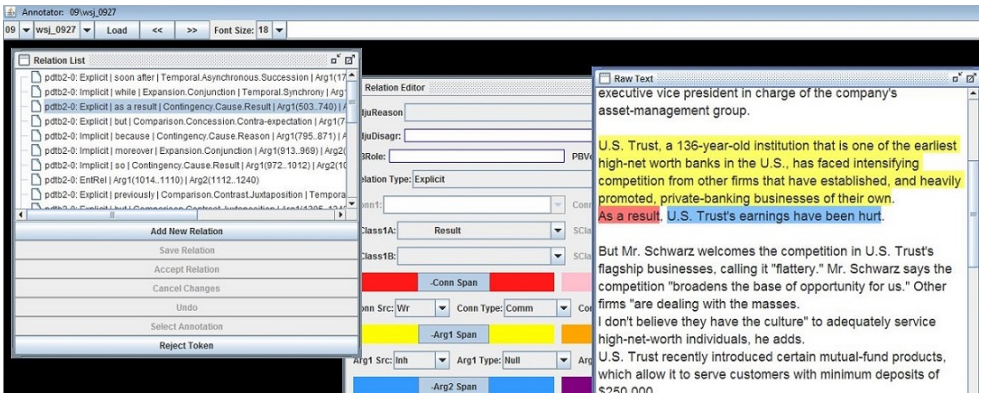

Figure 1: A view of the PDTB Annotator. Users may rearrange the subpanels as needed.

- The *Relation List panel* lists all annotation tokens for a particular text. Users also add new tokens, undo changes or reject a token here.
- The *Relation Editor panel* provides functionality for annotating the various features of a discourse relation - the relation type, the arguments of a relation (Arg1 and Arg2), the sense(s) of a relation, etc. Comments are also added on this panel.
- The *Raw Text panel* shows the actual text to be annotated. The annotator selects relevant portions of text using the mouse (discontinuous text spans are possible) and then switches to the Relation Editor to add the discourse relation features of the selected span.

Tokens are saved into a simple annotation file format, with one annotation file corresponding to each raw text file. Each token is represented by a pipe-delimited line of text and there are presently 34 fields in use. A description of each field can be found in The PDTB Group (2016). While most fields encode discourse relation features, a few additional ones are also used for adjudication and project management purposes (see Section 6). [1]

The original file format developed for PDTB 2.0 was designed for easy translation into Backus-Naur Form for use with the tools and APIs of the time. The difficulty of working with the older format led to the development of the simpler current format, where the pipe-delimited text entries can be easily processed by text-processing tools or imported into a spreadsheet.

5 Setup and Configuration

The PDTB Annotator makes use of three sets of files: i) text files; ii) annotation files; iii) comment files. Of these, only text files are obligatory. The resources needed for setting up an annotation project using the PDTB Annotator are minimal:

[1]Two fields - PB Role and PB Verb are specific to the PDTB. These were created to indicate links between certain PDTB tokens and semantic roles in the PropBank (Palmer, 2005). These fields can be left empty for other purposes.

- A set of text files. These should be raw text files and UTF-8 encoded.[2] A simple directory structure is assumed, consisting of a single base directory containing one or more sub-directories. The text files are distributed into these sub-directories. Naming conventions are up to the user.

- A base directory for annotation files. If an annotator is to annotate from scratch, this directory is left empty. An annotation file corresponding to each text file will be created dynamically as the annotation proceeds, mirroring the directory structure and file-naming convention of the text files.

The basic requirements aside, some common additional configuration or preprocessing steps include: i) defining a base directory for **comment files**, which lets the annotator comment on a token; ii) providing annotators with a set of **pre-annotated files**. For example, a set of explicit connectives might be pre-identified for annotation, automatically extracted from the raw texts and imported into the PDTB file format; iii) **Customizing the sense hierarchy**. This is done by simply modifying the text-based hierarchy provided in Options.cfg; iv) **Updating the list of implicit connectives** from the dropdown menu in the Relation Editor panel. This is also done by modifying Options.cfg, which contains by default a list of English connectives. A project might want to show connectives in a different language, for example.

6 Adjudication

The PDTB Annotator also doubles up as an adjudication tool. Using the tool this way, an adjudicator can evaluate corresponding tokens from two annotators.[3] For each pair of corresponding tokens, the adjudicator selects and potentially edits one of the tokens as the gold entry, then saves it into a gold file.

There is no additional setup needed to use the PDTB Annotator as an adjudication tool beyond specifying, upon launching the tool, the locations of the two sets of annotated files to be adjudicated. Figure 2 shows the "adjudicator view" of the Relation List panel introduced in Figure 1. Here, a list of gold tokens is shown and each node in the list can be expanded to show the pair of annotations being adjudicated, as shown for the third and fourth tokens. An unadjudicated gold token is displayed in red along with an agreement report - either "Annotators agree" (token #3), or "Disagreements" (tokens #2 and #4). Disagreements are reported for mismatches in sense, relation type or argument span. Adjudicated tokens are shown in black (token #1).

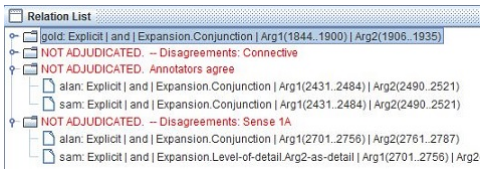

Figure 2: Adjudicator view of the Relation List panel

For each adjudicated token, the adjudicator can also specify the *reason* for the adjudication decision - e.g. due to annotator agreement, or the adjudicator agreed with one of the annotators, or corrections were made by the adjudicator, etc. For disagreeing tokens, the adjudicator can also specify the *type* of disagreement - e.g. a sense disagreement, a mismatch in argument spans, etc. There are fields reserved for these adjudication features in the annotation file format.

7 Conclusion

The latest version of the PDTB Annotator is designed to be convenient by serving the purposes of both annotation and adjudication. It can be run on many platforms and supports several writing systems and languages. By having control over the elements of some features, such as the sense hierarchy and implicit connectives, users can explore the suitability of other/additional senses or connectives for their corpus.

[2]By supporting UTF-8 encoded files, the PDTB Annotator works for a number of languages and writing systems. Romanized writing systems particularly benefit from the dynamic tokenization of the raw texts (by whitespace/punctuation), which makes it easier to select text spans using a mouse. Such tokenizations can be turned off for other writing systems.

[3]The current tool assumes at most two annotators, as agreement reports are based on a pair of annotators.

New features to record more fine-grained analyses during adjudication, such as the reason and type of disagreement, can be directly used to study task complexity in greater depth. The tool can be found at `http://www.seas.upenn.edu/~pdtb/annotator.html`. Any questions can be directed to the first author.

References

Al-Saif, A. and K. Markert. 2010. *The Leeds Arabic Discourse Treebank: Annotating discourse connectives for Arabic*. Proceedings of the 7th International Conference on Language Resources and Evaluation, Valletta.

Bejček, B., Hajičová, E., Hajič, J., Jínová, P., Kettnerová, V., Kolářová, V., Mikulová, M., Mírovský, J., Nedoluzhko, A., Panevová, J., Poláková, L., Ševčíková, M., Štěpánek, J. and Zikánová, S. 2013 *Prague Dependency Treebank 3.0*. Available at <http://ufal.mff.cuni.cz/pdt3.0/>

Mann, W. and S. Thompson 1988. *Rhetorical Structure Theory: A Theory of Text Organization*. Text 8(3):243281.

Marcu, D. n.d. Available at <http://www.isi.edu/licensedsw/RSTTool/index.html>

Mírovský, J., Jínová, P. and Poláková, L. *Discourse Relations in the Prague Dependency Treebank 3.0* Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, pp. 34-38

Müller, C. and M. Strube *Multi-Level Annotation of Linguistic Data with MMAX2*. In: Sabine Braun, Kurt Kohn, Joybrato Mukherjee (Eds.): Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods. Frankfurt: Peter Lang, pp. 197-214. (English Corpus Linguistics, Vol.3).

Orăsan, C. *PALinkA: A highly customisable tool for discourse annotation*. In Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue

Oza, U., R. Prasad, S. Kolachina, S. Meena, D. M. Sharma, and A. Joshi. 2009. *Experiments with annotating discourse relations in the Hindi Discourse Relation Bank*. Proceedings of the 7th International Conference on Natural Language Processing (ICON), Hyderabad.

Palmer M., P. Kingsbury P, D. Gildea 2005 *The Proposition Bank: An Annotated Corpus of Semantic Roles*. Computational Linguistics. 31 (1): 71106.

Prasad. R, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. *The Penn Discourse Treebank 2.0*. Proceedings of the 6th International Conference on Language Resources and Evaluation, Marrackech.

Prasad, R., S. McRoy, N. Frid, A. Joshi, and H. Yu. 2011. *The Biomedical Discourse Relation Bank*. The Biomedical Discourse Relation Bank. BMC Bioinformatics, 12(188):118.

The PDTB Group. 2016. *The PDTB Annotator*. <http://www.seas.upenn.edu/~pdtb/annotator.html>

Prasad, R., B. Webber and A. Joshi 2014. *Reflections on the Penn Discourse TreeBank, Comparable Corpora, and Complementary Annotation*. Computational Linguistics 49(4). pp. 921-950.

Webber, B., R. Prasad, A. Lee and A. Joshi 2016. *A Discourse-Annotated Corpus of Conjoined VPs*. Proceedings of the Tenth Linguistic Annotation Workshop (LAW). Berlin.

Xue, N., H. Ng, S. Pradhan, R. Prasad, C. Bryant and A. Rutherford 2015. *The CoNLL-2015 Shared Task on Shallow Discourse Parsing*. Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task. Beijing.

Zeyrek, D., I. Demirşahin, A. Sevdik-Çallı and R. Çakıcı 2016. *Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language*. Dialogue and Discourse, 4(2):174184.

Zeyrek, D., A. Mendes, S. Gibbon, Y. Grishina, M. Ogrodniczuk 2016. *TED-Multilingual Discourse Bank (TED-MDB): TED Talks annotated in the PDTB style.*. (in preparation).

Zhou, Y. and N. Xue 2012. *PDTB-style discourse annotation of Chinese text*. Proceedings of the 50th Annual Meeting of the ACL, Jeju Island.