4.1 Introduction

Multiple research fields in NLP have pairs of texts as their object of study: Paraphrasing, Textual Entailment, Text Summarization, Text Simplification, Question Answering, and Machine Translation, among others. All these fields benefit from high quality corpora, annotated at different granularity levels. However, existing annotation tools have limited capabilities to process and annotate such corpora. The most popular state-of-the-art open source tools do not natively support detailed pairwise annotation and require significant adaptations and modifications of the code for such tasks.

We present the first version of WARP-Text, an open source[1] web-based annotation tool, created and designed specifically for the annotation of relationships between pairs of texts at multiple layers and at different granularity levels. Our objective was to create a tool that is functional, flexible, intuitive, and easy to use. WARP-Text was built using PHP and MySQL standard implementation.

WARP-Text is highly configurable: the administrator interface manages the number, order, and content of the different annotation layers. The pre-built layers allow for custom definitions of labels and granularity levels. The system architecture is flexible and modular, which allows for the modification of the existing layers and the addition of new ones.

The annotator interface is intuitive and easy to use. It does not require previous knowledge or extensive annotator training. The interface has already been used in the task of annotating atomic paraphrases [Kovatchev et al., 2018a] and is currently being used on two annotation tasks in Text Summarization. The learning process of the annotators was quick and the feedback was overwhelmingly positive.

The rest of this article is organized as follows. Section 4.2 presents the Related Work. Section 4.3 describes the architecture of the interface, the annotation scheme, the usage cases, and the two interfaces: administrator and annotator. Finally, Section 4.4 presents the conclusions and the future work.

4.2 Related Work

In the last several years, the NLP community has shown growing interest in tools that are web-based, open source, and multi-purpose: WebAnno [Yimam and Gurevych, 2013], Inforex [Marcińczuk et al., 2017], and Anafora [Chen and Styler, 2013]. Other popular non web-based annotation systems include GATE [Cunningham et al., 2011] and AnCoraPipe [Bertrán et al., 2008]. These systems

[1]The code is available at `https://github.com/venelink/WARP` under Creative Commons Attribution 4.0 International License.

are intended to be feature-rich and multi-purpose. However, in many tasks, it is often preferable to create a specialized annotation tool to address problems that are non-trivial to solve using the multi-purpose annotation tools. One such problem is working with multiple texts in parallel. While multi-purpose annotation tools can be adapted for such use, this often leads to a more complex annotation scheme, complicates the annotation process, requires additional annotator training and post-processing of the annotated corpora. Toledo et al. [2014] and more recently Nastase et al. [2018], Batanović et al. [2018], and Arase and Tsujii [2018] emphasize the lack of a feature-rich open-source tool for annotation of pairs of texts[2]. Some of these authors develop simple custom-made tools with limited reusability, designed for for carrying out one specific annotation task. WARP-Text aims to address this gap in the NLP toolbox by providing a feature rich system which could be used in all these annotation scenarios.

To the best of our knowledge, the only existing multi-purpose tool that is designed to work with pairs of text and allows for detailed annotation is CoCo [España Bonet et al., 2009]. It has already been used for annotations in paraphrasing [Vila et al., 2015] and plagiarism detection [Barrón-Cedeño et al., 2013]. However, CoCo is not open source and is currently not being supported or updated.

4.3 WARP-Text

By addressing various limitations of existing tools, WARP-Text fills a gap in the state-of-the-art NLP toolbox. It offers project managers and annotators a rich set of functionalities and features: the ability to work with pairs of texts simultaneously; multi-layer annotation; annotation at different granularity levels; annotation of discontinuous scope and long-distance dependencies; and the custom definition of relationships. WARP-Text consists of two separate web interfaces: annotator and administrator. In the *administrator interface* the project manager configures the annotation scheme, defines the relationships and sets all parameters for the annotation process. The annotators work in the *annotator interface*.

WARP-Text is a tool for qualitative document annotation. It provides a wide range of configuration options and can be used for fine-grained annotation. It is best suited to medium sized corpora (containing thousands of small documents) and is not fully optimized for processing, analyzing, searching, and annotating large corpora (containing millions of documents). WARP-Text has full UTF-8 support and is language independent in the sense that it can handle documents in

[2]See also the discussion about looking for tools for annotating pairs of texts in the Corpora Mailing List (May 2017): http://mailman.uib.no/public/corpora/2017-May/026526.html - http://mailman.uib.no/public/corpora/2017-May/026619.html

any UTF-8 supported natural language. So far it has been used to annotate texts in English, Bulgarian (Cyrillic), and Arabic.

WARP-Text is a multi-user system and provides two different forms of interaction between the different annotators. In the *collaborative mode*, multiple annotators work on the same text and each annotator can see and modify the annotations of the others. In the *independent mode*, the annotators perform the annotation independently from one another. The different annotations can then be compared in order to calculate inter-annotator agreement.

4.3.1 Annotation Scheme

The atomic units of the annotation scheme in WARP-Text are *relationships*. The properties of the *relationships* are *label* and *scope*. The *scope* of a *relationship* is a list of continuous or discontinuous *elements* in each of the two texts. The granularity level of the scope determines the *element* type. An *element* can be the whole text, a sentence, a phrase, a token, or can be defined manually. A *layer* in WARP-Text is a set of relationships, whose scopes belong to the same granularity level[3]. The definition of relationships and their grouping into layers is fully configurable through the administrator interface. WARP-Text supports multi-layer annotation. That is, the same pair of texts can be annotated multiple times, at different granularity levels and using different sets of relationships.

4.3.2 Administrator Interface

The administrator interface has three main modules: a) the *dataset management module*, b) the *user management module*, and c) the *layer management module*. In the *dataset management module* the project manager can: a) import a corpus, in a delimited text format, for annotation; b) monitor the current annotation status and statistics; and c) export the annotated corpus as an SQL file or an XML file. In the *user management module* the project manager creates new users and modifies existing ones. In this module the project manager also distributes the tasks (pairs) among the annotators. In the *layer management module* the project manager configures each of the layers and determines the order of the layers in the annotation process. The project manager configures for each individual layer: 1) the granularity level; 2) the relationships that belong to the layer; 3) the sub-relationships

[3]There is no one-to-one correspondence between granularity level and annotation layer. Each annotation layer is a sub-task in the main annotation task. Multiple annotation layers can work at the same granularity level. For example: at layer (1) the annotator annotates the semantic relations between the tokens in the two texts; at layer (2) the annotator annotates the scope of negation and the negation cues in the two texts. Both layer (1) and layer (2) work at the token granularity level.