Managing a Multilingual Treebank Project

Milan Souček **Timo Järvinen** **Adam LaMontagne**
Lionbridge
Finland
{milan.soucek,timo.jarvinen,adam.lamontagne}@lionbridge.com

Abstract

This paper describes the work process for a Multilingual Treebank Annotation Project executed for Google and coordinated by a small core team supervising the linguistic work conducted by linguists working online in various locations across the globe. The task is to review an output of a dependency-syntactic parser, including the POS types, dependency types and relations between the tokens, fix errors in output and prepare the data to a shape that can be used for further training of the parser engine. In this paper we focus on the implemented Quality Assurance processes and methodology that are used to monitor the output of the four language teams engaged in the project. On the quantitative side we monitor the throughput to spot any issues in particular language that would require intervention or improving the process. This is combined with a qualitative analysis that is performed primarily by comparing the incoming parsed data, the reviewed data after the first round and after the final cross-review using snapshots to compile and compare statistics. In addition, the possible inconsistencies in the annotations are checked and corrected automatically, where possible, in appropriate stages of the process to minimize the manual work.

1 Introduction

Multilingual dependency parsing has become an important part of dependency parsing tasks, mainly due to growing needs of the cross-language sources for supporting machine translation, search and retrieval and other natural language applications. Different approaches to processing multilingual data have been investigated in recent years and their outputs compared in a series of CoNLL shared tasks on multilingual dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007; Surdeanu et al., 2008; Hajič et al., 2009). One of the possibilities for building multilingual parsers is training parsers from annotated data that was presented e.g. in models developed by McDonald et al. (2005) and Nivre et al. (2006). Preparing an annotated Treebank for training purposes is a resource-intensive task. For that reason, such tasks have to be planned and coordinated in such a way that the work is processed efficiently and unnecessary costs are eliminated. One manner of achieving efficiency is to prepare annotated data for multiple languages at the same time, which allows the data annotation provider to establish a consistent environment for creating and maintaining cross-language annotation guidelines and processes. In our current project, we are working with Google to review and prepare annotated data for training a multilingual parser using the Stanford typed dependencies model – a simple model represented by part of speech and dependency relation types recognizable across languages (de Marneffe and Manning, 2008).

The scope of this project covers manual review of 15 000 parsed sentences for each of four involved languages – German, French, Spanish and Brazilian Portuguese. For German, Spanish and French, a supervised training model is used for parsing the data before annotation (Zhang and Nivre, 2011). For Brazilian Portuguese, a cross-lingual parser is used (McDonald et al., 2011), where delexicalized model is trained on Spanish and French data with assistance of the part of speech tagger (Das and Petrov, 2011). Data corpus used for parsing is domain-based, the current scope of the project does not target representativeness. For German, French and Spanish, Wikipedia texts were used as the main data source, for Brazilian Portuguese, mainly news texts were included. Brazilian Portuguese also follows a different timeline and for that reason, we don´t present any results gained for this language in the current paper. Data are batched in groups of 100-500 sentences per file. The parsing system performs the tokenization of the data that separates punctuation as individual tokens. The pre-parsed data contains three levels of annotation: part-of-speech (POS) labels, binary dependencies between the tokens and dependen-

Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pages 292–297,
Prague, August 27–30, 2013.

cy relation labels (deprel). All these levels are reviewed and corrected in the process.
The number of dependency relations varies slightly between the languages. Some 51 to 57 labels are used. However, one of the targets of the project is to review the inventory of the relations to ensure uniform representation across the languages and some adjustments to the inventory of the dependency labels were made in the initial phase of the project.
Files are processed in PML format (Pajas and Štěpánek, 2006) using the Tree Editor TrEd 2.0 (Hajič et al., 2001) and CoNLL2009 stylesheet extension. In order to see the parser engine improvement and to be most efficient with the manual annotation, the data is processed in sprints made up of around 1500 sentences. Each sprint follows the cycle described in section 5 in this paper and its output is used for training the parser. Parser performance improves with each training and each manual annotation round requires less effort and time.

2 Initiating the project

A fundamental step for successfully running a multilingual language technology process is the planning phase. On the technical side, we concentrated on creating a consistent environment for processing the data, where all tools would be easily manageable by both the core team and the distributed linguists. A virtualized desktop environment is currently the most efficient environment for handling this kind of project. On the resourcing side, we built a multilingual team of linguists with expertise in the field of syntax. Our requirements for annotation experts emphasized candidates' target language knowledge (though, for syntax analyses, native knowledge is not strictly required) and linguistic studies background with special stress on studies in syntax and previous experience with the review and annotation work. Finally, for the actual work process, we created a model of cross-review annotation work, where two annotators work on manual annotation of the same set of data in two phases. Manual annotation is supported by automated validation tools that report statistics for evaluation of output data quality and annotators' throughput.

3 Maintaining consistency

In order to ensure technical and annotation consistency in the team that consists of several annotators per language, we developed processes that allow the team to work in a consistent environment and in a real-time online collaboration. Technical consistency is achieved by the use of a virtual machine for all project work. The possibility of instant communication allows the team to discuss actual problems related to annotation decisions and guidelines interpretation. This dynamic process is further supported by a secure online interface that contains all project data and documentation and that allows all project participants (including annotators, internal support team and product owner) to have full visibility of the production cycle and provide feedback on the tools, annotation process, output quality or any other specific aspect of the project.

3.1 Technical consistency

All work on the Multilingual Treebank Annotation Project is done in a virtual machine (Cloud), where annotators connect from their own distributed work stations via remote desktop client. This allows the core team to centrally manage the tools and support, allowing the participants to focus on the linguistic tasks. Also all project data is stored in secure data shares and is accessible at all times to the core team who can then make any necessary manipulations to the data and also easily manage the workflow progress.
In the cross-review work model, where multiple annotators are working on the same file, one of the challenges is to maintain version control. We use Tortoise SVN Version Control system to manage file versions. SVN is a powerful tool that helps to track the latest version of all files that are being worked on. It also has functions to compare different file versions and resolve conflicts between them. Finally, we use a centralized progress tracker, which is a macro-driven tool that collects statistics about processed data from annotators' individual tracking reports which are kept on the virtual server. The individual tracking reports calculate throughput and other statistics by task based on data input by the linguists. The centralized progress tracker collects the statistics from the individual reports (see example in Table 1). This data is used further for evaluation of the annotation progress for each language, as well as, with connection to quality check results, for evaluation of individual performance of each annotator.

French	All	Ann 1	Ann 2	Ann 3
Total A	1653	470	833	350
Total R	2135	150	1985	0
A Through	14,03	19,58	13,04	9,48
R Through	91,19	75	107,39	0
Total Hours	164	26	101	37

Table 1: Example of throughput statistics. The figures are the number of sentences processed by individual annotators (Ann 1, 2, 3) in the first annotation round (A) and in the review round (R). Throughput is expressed by amount of sentences processed per hour.

3.2 Annotation consistency

To ensure a high level of annotation consistency, we use a dynamic work model that includes introductory hands-on trainings for annotators, general guidelines for handling cross-language annotation scenarios, language specific annotation guidelines and a centralized team communication portal, where annotators discuss annotation problems and decisions (see also Figure 1). At the initial stage of the project, each annotator reviews language specific annotation guidelines and a sample Gold Standard annotations data. Hands-on training follows, where annotators work on actual parser output, collect questions and problematic cases and discuss them with the other team members in the discussion portal with specific reference to the language guidelines documentation. At this stage, problematic areas in dependency guidelines are reviewed and the master guidelines are updated with clarifications and annotation examples. Annotators also review each other's work and provide further feedback to the team about annotation errors observed. This annotation-review model complies with the double-review model described in section 5.

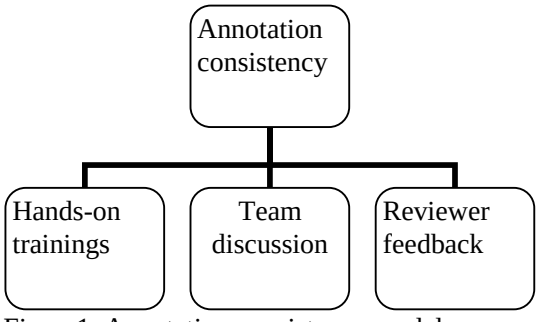

Figure1. Annotation consistency model.

During the initial project stage, usually after each annotator has processed about 500 sentences, the lead annotator for each language is identified, based on previous experience, work quality output and throughput speed. With guidance from the core team, the lead annotator is then responsible for coordinating team discussions and updating master guidelines documents. Lead annotators also review language specific guidelines for other languages and compare annotation decisions applied in their language with decisions for same or similar patterns used by other language teams. Result of this cross-language guidelines review opens up a dialog across teams and helps to further improve cross-language consistency. In the double-review work model, the lead annotator is mainly engaged in the review phase of annotation work, where data manually annotated during the first review round is reviewed for remaining consistency and human errors. This work model helps the team to achieve a consistent approach to annotating the data and minimizes inconsistency in the final output. In addition to dependency guidelines, annotators use a project-specific online discussion board, where they post questions and suggestions for preferred handling of annotation cases. Annotators can review also other languages' postings and comment on them as well as suggest general patterns handling for cross-language consistency.

4 Data validation

In addition to manual review, all files are validated using manual and automated validation tools. Automatic data validation consists of two parts – technical validation that is handled by an xml schema attached to each annotation file and linguistic validation that uses POS representation of deprel parent/child participants.

4.1 XML validation

For xml validation, we implemented lists of possible POS and deprel type labels for each language to xml schema that is attached to each pml file. POS and deprel labels appear as drop-down selections in the TrEd application, which eliminates a risk of incorrect label input, such as may occur if the labels were manually entered by the annotator. In case an invalid annotation label appears in the pml file, the TrEd application reports an error and the annotator can find and fix the label accordingly.

4.2 POS vs. deprel validation

Automated linguistic validation is based on possible combinations of POS participants in dependency relations (see Table 2 for example of validation tool settings). If the predefined POS representation for a deprel is violated in the data, the validation tool flags the item for review. The annotator can compare the validation results with annotated data to evaluate whether the error is valid and should be fixed or whether the error can be ignored for being an uncommon or exceptional combination of POS types in the particular deprel. POS representation for deprels is not absolute and we try not to over-define the representation in validation tool settings, in order not to miss obvious errors in output that might be hidden by too benevolent POS vs. deprel definitions. The validation tool results are therefore only approximate, but they can nevertheless show pattern errors or misinterpretation of deprel usage that might not be visible to annotators during manual review.

deprel	parent POS	current POS
amod	NOUN, PNOUN, ADJ	ADJ
aux	VERB	AUX

Table 2. Example of POS vs. deprel set-up for the linguistic validation tool. "parent POS" means the head of deprel for the current token, "current POS" means the POS of current token being validated.

We use linguistic validation tool results also as a source for relative statistical evaluation of data quality improvement during the annotation and review work on the data and for evaluating annotators' output quality. Assuming that the POS vs. deprel settings are consistent throughout the project, we can easily compare reported error rate between different stages of annotation work as well as between different annotators for each deprel label. For data improvement statistics, we take a snapshot upon receiving the data from the parser for a baseline reading, and then again after the manual annotation and finally after the annotation review. In addition to expected improvement in error rate between each stage, we can also compare results for each stage between different data batches to see e.g. improvement in parser output or expected error rate for the final annotated and reviewed file. Comparison of error rate for each annotator gives information about annotators' individual performance, in addition to their throughput speed. In terms of the data, this information helps to identify potential quality gaps, possible improvement areas in the guidelines and may give insight into the improvement of the parser in general. In terms of the language teams, statistical results provide insight about which annotators are good candidates for lead annotator and also give feedback about possible unwanted under-performance.

5 Data process cycle

We process the data in sprints of about 1500 sentences. Each sprint lasts about 2 weeks and all 4 languages follow the same sprint schedule, so that consistent amount of sentences can be delivered for all languages at the same time. The overall duration of the project was initially estimated for 22 weeks (11 sprints), including trainings, dependency guidelines development and manual annotation tasks. For the first three languages – French, German and Spanish, the project was also completed within the planned schedule, with the last batch of annotated data delivered at the end of 22nd week. For each project sprint, input data (natural sentences) are first parsed by the original version of the parser and converted to pml format. Parsed data is then pre-processed – the xml schema for each pml file is updated with an up-to-date list of annotation labels and systematic errors in parsed data are fixed by automated scripts. Systematic errors fixed in pre-processing are e.g. invalid and obsolete labels that were removed during the cross-language guidelines harmonization in the initial stages of the project. After automated fixes are done, the first linguistic validation snapshot is taken and statistics are reported. At this stage, pattern errors generated by the parser can be observed from validation results. The file is then assigned to an annotator for manual annotation with information about pattern errors present in the file. The annotator reviews the tree structure and syntactic labels (POS and deprel labels) and updates them, where necessary. Once manual annotation is done, a second validation snapshot is taken and statistics are reported. Validation results are shared with the reviewer, who works on their analysis and on fixing remaining errors in the input file. At this stage, quality feedback for annotator's work also is

collected – both snapshot statistics and manually collected feedback from the reviewer serve as data for personal performance evaluation. After manual review of the input file, the third validation snapshot is taken and statistics are reported. Validation results are reviewed for any remaining errors, mainly for invalid annotation labels. Finally, text content of input sentences is compared between the original parsed and final reviewed file to fix any possible changes in content that are not allowed in this project. Final files are used for training the parser and another set of input data is parsed with the re-trained parser for further annotation work.

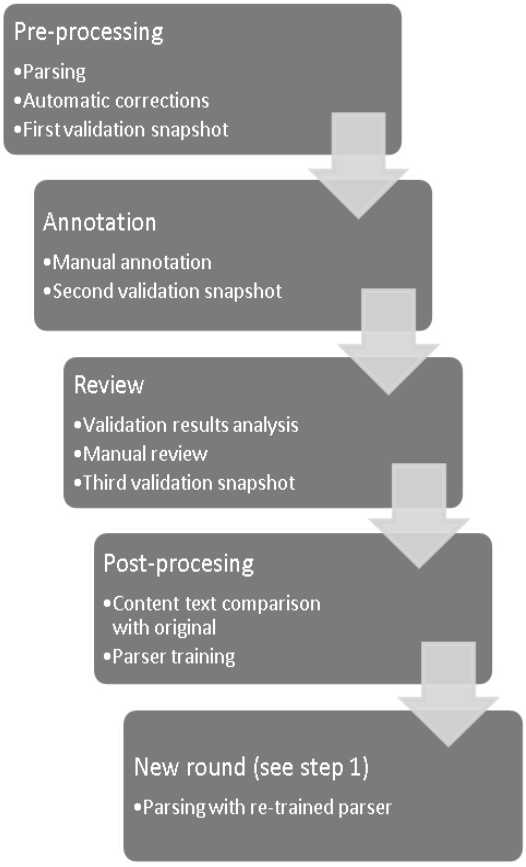

Figure 2. The parser training process cycle.

The goal of this cycle is to train the parser with batches of annotated data that improve the performance of the parser before it is used for parsing the next batch of sentences for annotation. Time, throughput and efficiency for manual annotation should improve with every new training and in every new annotation round. As a result of the training and improved parser output, a greater amount of data can be processed in the next sprint. Table 3 shows comparison of annotation throughput development for French, Spanish and German (since Brazilian Portuguese follows a different timeline in this project, comparable data was not available yet for this language). Batches represent sets of sentences received from pre-parsing. Batches marked with asterisk were pre-parsed with re-trained parser, which also reflects in throughput improvement. Other batches were pre-parsed altogether with the preceding batch and the throughput improvement reflects rather the learning curve of annotators´ individual performance. The drop in French and Spanish in Batch 6 was caused by addition of some targeted data for patterns that the parser did not treat correctly at that time (question sentences).

	French	Spanish	German
Batch1	13.00	8.91	13.94
Batch2*	23.00	14.97	23.63
Batch3*	26.50	20.88	24.05
Batch4	26.70	22.55	24.80
Batch5*	28.38	27.49	27.47
Batch6	25.52	25.56	29.37
Batch7*	30.32	27.56	43.52
Batch8	31.37	28.74	43.95

Table 3. Throughput development comparison for French, Spanish and German.

Summary

In the current paper, we presented an example of a workflow for a Multilingual Treebank Annotation Project work. We aim to provide consistency both in technical and linguistic output for the annotated data and to bring efficiency to manual processing of parsed input data. Validation and throughput tracking tools used in the project are examples of control tools for maintaining consistency, quality and efficiency in manual annotation work. As result of our initial tools and processes testing, we aim to improve our tools further by implementing e.g. lexical check tools for evaluating validity of POS annotation and more precise validation of POS vs. deprel representation. The project now continues with extended scope of languages and tasks. In our future reports, we plan to concentrate on presenting further results of the parser training process and comparison of language specific dependency guidelines.

References

Sabine Buchholz and Erwin Marsi. 2006. *CoNLL X shared task on multilingual dependency parsing*. In Proceedings of CoNLL.

Dipanjan Das and Slav Petrov. 2011. *Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections.* In Proceedings of ACL.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antónia Martí, Lluís Márquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, Yi Zhang. 2009. *The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages*. In Proceedings of CoNLL.

Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas. 2001. *The Prague Dependency Treebank: Annotation Structure and Support.* In Proceedings of the IRCS Workshop on Linguistic Databases, pages 105–114, Philadelphia, USA. University of Pennsylvania.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. *Stanford typed dependencies manual.*

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. *Online large-margin training of dependency parsers*. In Proceedings. of ACL.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. *Multi-source transfer of delexicalized dependency parsers*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 62–72, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. *Maltparser: A data-driven parser-generator for dependency parsing*. In Proceedings of LREC.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. *The CoNLL 2007 shared task on dependency parsing*. In Proceedings of EMNLPCoNLL.

Petr Pajas and Jan Štěpánek. 2006. *XML-based representation of multi-layered annotation in the PDT 2*.0. In Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006).

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. *The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies*. In Proceedings of CoNLL.

Yue Zhang and Joakim Nivre. 2011. *Transition-based dependency parsing with rich non-local features*. In Proceedings of ACL-HLT.