Midas Loop: Prioritized Human-in-the-Loop Annotation for Large Scale Multilayer Data

Luke Gessler, Lauren Levine, Amir Zeldes
Georgetown University
Department of Linguistics
{lg876, lel76, amir.zeldes}@georgetown.edu

Abstract

Large scale annotation of rich multilayer corpus data is expensive and time consuming, motivating approaches that integrate high quality automatic tools with active learning in order to prioritize human labeling of hard cases. A related challenge in such scenarios is the concurrent management of automatically annotated data and human annotated data, particularly where different subsets of the data have been corrected for different types of annotation and with different levels of confidence. In this paper we present Midas Loop, a collaborative, version-controlled online annotation environment for multilayer corpus data which includes integrated provenance and confidence metadata for each piece of information at the document, sentence, token and annotation level. We present a case study on improving annotation quality in an existing multilayer parse bank of English called AMALGUM, focusing on active learning in corpus preprocessing, at the level of sentence segmentation, which remains surprisingly challenging for automated systems. Our results show improvements to state-of-the-art sentence segmentation and a promising workflow for getting "silver" data to approach gold standard quality.

Keywords: corpus, annotation, collaborative, active learning, multilayer, sentence segmentation, human in the loop

1. Introduction

Multilayer corpora (Ide et al., 2010; Santos and Mota, 2010; Zeldes, 2018) are richly annotated language resources that contain information about a variety of linguistic phenomena in parallel, such as morpho-syntactic analyses, named entity recognition, semantic role labeling or 'PropBanking' (Palmer et al., 2005), coreference resolution and more. While they are highly valuable for both linguistic studies and computational applications, such datasets can be challenging to maintain: the existence of multiple annotations for each text means that different annotations may be aligned or interconnected, that segmentations such as word tokenization and sentence splitting will often need to match across layers (Krause et al., 2012), and that correcting one part of a corpus may have complex consequences for another (Peng and Zeldes, 2018). These challenges can more easily be overcome for small, hand-curated datasets, but may become unmanageable for larger corpora, especially if iterative improvement and corrections to the data are envisioned.

In this paper we present a new, open-source, production-ready system for iterative correction of large-scale multilayer data. The system, called Midas Loop, integrates with retrainable NLP models to provide confidence metadata for CoNLL-U annotations. This confidence metadata allows for both the targeting of low confidence areas of the data for manual review, as well as harnessing higher confidence areas of the data in order to curate subsets that can be used for tasks that have specific requirements regarding annotation quality.

We use the freely available AMALGUM corpus (Gessler et al., 2020) as a case study, containing 4M tokens in 8 English genres, automatically annotated for high quality Universal Dependencies (UD) parses (incl. enhanced dependencies); document structure using TEI p5 XML tags (Burnard and Bauman, 2008); typed and nested named and non-named entity recognition; normalized time expressions; coreference resolution; and discourse parses in Rhetorical Structure Theory (Mann and Thompson, 1988). Of these tasks, our system currently handles sentence segmentation (at the document level), as well as structural tasks which are edited at the sentence level, including POS tagging, lemmatization, dependency syntax corrections etc. These capabilities thus encompass the standard UD/CoNLL-U format column annotations[1], and in the future we plan to add extensible support for other kinds of annotations expressed in the MISC column or metadata lines of the CoNLL-U format, such as annotations for entities, coreference and discourse parses.

Since the substantial size of the data curated by the system makes comprehensive manual correction unfeasible, we adopt an active learning strategy, which allows users to query the system for likely errors based on NLP model output probabilities, which are then highlighted in context and presented to annotators.

We evaluate the effectiveness of our strategy on the surprisingly tricky task of automated sentence splitting in multiple genres, by iteratively retraining tools on high-priority corrected data in a synergistic cycle of manual and automated correction. The resulting data contains mixed gold and silver quality annotations, which necessitate facilities for keeping track of version controlled annotation provenance, as well as qualitative and quantitative quality estimates at the document, sentence, token and annotation levels.

The main contributions of this paper are:

[1] `https://universaldependencies.org/format.html`; see below for more details

1. We present an open source annotation system for large scale multilayer corpus correction incorporating active learning across a broad range of tasks, which highlights uncertain NLP outputs prioritized for annotator correction and tracks annotation quality through metadata.
2. We also present a new and improved version of this work's test case corpus, AMALGUM, with very high quality automatic and some manually corrected NLP output.
3. We evaluate the effectiveness of active learning for sentence splitting and achieve a substantially improved SOTA score for English sentence splitting on the genre-diverse gold standard GUM dataset, which includes both spoken and written data, as well as challenging unedited user generated content from the Web. (Sanguinetti et al., 2022)

2. Previous work

2.1. Multilayer annotation

Because of their complex structure and potential interdependencies between layers, multilayer corpora can be particularly challenging to annotate and to maintain. While an initial focus on correcting treebanking (Lai and Bird, 2004) allowed the use of single tools without many cross-checks, subsequent work on integrating frame semantics, prosody and pragmatics led to multilayer data with intertwined syntactic, phonological, semantic and pragmatic graphs that pushed single interface tools to their limit, as in the SALSA project (Burchardt et al., 2008) or the NXT Switchboard Corpus (Calhoun et al., 2010). Later corpora such as MASC (Ide et al., 2010) and OntoNotes (Weischedel et al., 2012) added increasingly many levels of annotation, such as concurrent word senses, semantic role labeling, coreference resolution and named entity recognition, in addition to morpho-syntactic analyses, with the result that separate tools were often used for editing each layer.

Many single-task annotation interfaces exist for the layers handled by our system, including Arborator (Gerdes, 2013) and UD Annotatrix (Tyers et al., 2018) for dependency trees, and CorefAnnotator (Reiter, 2018) for coreference annotation. There also exist widely used generic web based tools, such as WebAnno (Eckart de Castilho et al., 2016) and INCEpTION (Klie et al., 2018), which target the annotation of typed spans and relations. Such tools are highly effective for individual annotation types. However, they are not designed to simultaneously handle the full spectrum of annotation types found in multilayer corpora, nor do they interact well with concurrent editing of segmentation and sentence-level annotations, or preserve versioned provenance information during iterative improvements to documents.

There are also a few examples of annotation tools tailored to multilayer editing, including FoLiA (van Gompel and Reynaert, 2014) and Atomic (Druskat et al., 2014), which were built from the ground up to support diverse, possibly interdependent, annotations in a single graph data model. Our approach follows these in that we use a single data model to support multiple layers, though we maintain a closer workflow to annotation of corpora such as OntoNotes, in that each annotation task interface is specialized and separate, exposing only necessary facets of the data and simplifying user interactions by limiting the amount of training required for each task. However, this inevitably means that our API must keep track of single layer changes which have meaningful consequences for other layers, which we manage in a non-destructive and version controlled way during updates (see Section 3).

2.2. Active learning

Active learning (AL), initially called 'uncertainty sampling' (Lewis and Gale, 1994) has a long history in NLP as a technique to reduce the amount of data required to learn a task: by targeting uncertain outputs from a large pool of automatically labeled data, human annotators can focus effort on resolving cases that algorithms find particularly challenging. AL continues to be applied successfully in recent papers for sentence classification (Ein-Dor et al., 2020), Named Entity Recognition (NER) (Shen et al., 2017), paraphrase detection (Bai et al., 2020), sentiment analysis (Ashrafi Asli et al., 2020) and much more.

We observe two trends in previous work on annotation systems for AL: 1. they typically target a single, specific task and/or domain (e.g. NER output for biomedical data) and typically only support relatively simple structures, such as non-overlapping span annotations (Searle et al., 2019; Lin et al., 2019) or document classification (Wiechmann et al., 2021); 2. they often simplify tasks by presenting specific questions to annotators: for example, a system might present a pair of mentions with questionable coreference status to an annotator for validation, substantially simplifying the interaction and interface requirements (Li et al., 2020).

Such systems can be highly valuable for targeted needs, however they fall short when the goal is to iteratively upgrade large-scale, silver-quality data into a gold-standard-near multilayer resource, with comprehensive linguistic annotations. Probably the closest existing tool to Midas Loop in implementing these goals is *prodigy*[2], which allows annotation with AL for customizable spans, as well as some graph annotations; however it is a non-freely available commercial tool, is tied to the SpaCy NLP platform,[3] which does not support some of our annotation workflows, and cannot handle discourse trees, which are relevant to our work with AMALGUM.

Finally, although AL is generally expected to improve NLP tool accuracy, care must be taken to prevent a focus on skewed outlier data, which can result

[2] https://prodi.gy/
[3] https://spacy.io/

if AL-selected examples outnumber 'normal' common examples, or substantially alter their relative likelihood (Baldridge and Osborne, 2004; Karamcheti et al., 2021). In our experiment in Section 4 we therefore focus on choosing entire documents with high levels of uncertainty (which presumably also contain 'common' cases), rather than just individual sentences from all documents, but the risk of data skewing nevertheless remains. To assess the practical impact of AL in the context of the present project, Section 4.2 evaluates the gains from targeted data selection for one early and very important task in the compilation of multilayer corpora: sentence splitting.

3. System Architecture

Midas Loop can be divided into two parts. The core system is a web server which maintains the state of the data and allows changes to be made to the data via an HTTP API. The frontend system is a web browser application which provides a graphical user interface with multiple annotation components for making changes to data. Guidance from machine learning models on which annotations are most dubious (and therefore most in need of manual review) is stored in order to be visually indicated in the interface.

Our frontend system's functionality enables the human-in-the-loop workflow described in this paper and enables editing of most annotations in the popular CoNLL-U format adopted by UD. However, the core system's API is agnostic regarding the frontend interface, and as such it is also possible to interact with the core system in other ways: for example, another web browser frontend could be created, or a crowdsourcing study on Amazon Mechanical Turk could send updates to the core system, which is an independent component.

3.1. Core System[4]

Overview The core system is a web server implemented in Clojure[5] which provides an HTTP API for clients to create, read, update, and delete CoNLL-U annotations. Token-based authentication restricts access to only authorized users, and it is possible to import and export data both via the HTTP API and the command-line. The core system is distributed as a single standalone .jar file and works on any platform with a Java Virtual Machine implementation. The core system contacts NLP services via HTTP and is therefore completely decoupled from them, allowing services to be implemented ad hoc in another programming language, such as Python. A full description of the API is included in the system's repository.

Data Model Internally, CoNLL-U file strings are deserialized and represented as a graph. Each document, sentence, metadata line, and "token" (i.e., 10-column row) is represented as a node. Additionally, each annotation within a token is represented as a node: for each token, there is a separate node for its FORM column, and for fields with multiple annotations like morphological features (FEATS) and MISC annotations,[6] each key-value pair is represented as a separate node. This proliferation of graph structure is needed in order to easily keep track of which annotations are human-verified "gold" annotations, and which annotations are NLP system-provided "silver" annotations: some tokens may have e.g. a gold part of speech annotation but silver syntactic head and dependency relation annotations.

Database The immutable graph database XTDB[7] is used to store and process this representation. We additionally note that XTDB stores the full history of all past database states. This functionality is not used by the core system at the moment, but it could be used in the future in order to allow access to **all** past versions of a certain document or sentence.

NLP Integration In order for active learning support to be available for a certain kind of annotation, an NLP system must be available which can provide annotation probabilities. This functionality is entirely "opt-in" and may be configured for as many or as few annotation kinds as desired. It is required that NLP systems are reachable via HTTP and can handle a few standardized API calls, and we anticipate that users will find it most convenient to take existing NLP models and wrap them in an implementation of this HTTP protocol using a Python web framework such as Flask.

NLP services are consulted at a sentence-level resolution: every time any element of a sentence changes, all registered NLP services are notified, and have the opportunity to provide new annotations and probability distributions for the layer in that sentence. Annotations from NLP services will overwrite existing annotations, unless an existing annotation is "gold" (i.e. manually added by an annotator), in which case the existing annotation will not be overwritten. For example, if sentence segmentation is altered, we assume that an automatic parser should be called to parse the resulting, newly formed sentences.

Supported Data The core system provides full support for reading, editing, importing, and serializing core datatypes in a standard CoNLL-U file. This includes changes to the 10 standard columns, as well as changes to sentence splits. Multiword and empty tokens as specified in the CoNLL-U format are fully supported. Changes to tokenization, changes to metadata lines, enhanced dependency editing,[8] and creation of new textual data other than via import of a CoNLL-U string are cur-

[4] https://github.com/gucorpling/midas-loop-ui.git

[5] https://clojure.org/

[6] See https://universaldependencies.org/format.html

[7] https://xtdb.com/

[8] For English, as in other UD data, we currently propagate corrected enhanced dependencies automatically based on corrected un-enhanced morphosyntax.

rently unsupported, but are planned for future releases.

Future Supported Data Additionally, although our system does not yet support editing of annotations not natively expressed in CoNLL-U, such as those for entities, coreference, and discourse, we plan to support these eventually using a configuration which will tell the system how to read them from the MISC column or metadata lines. We also plan to support representations proposed by the recent Universal Anaphora project (`http://universalanaphora.org/`). These extensions will allow the system to continue working with just CoNLL-U while allowing it to process arbitrary annotations.

3.1.1. Layer Interdependencies

As some annotation layers have dependencies on others, a word on how layer dependencies are handled in our system is warranted. For instance, head attachments in a dependency syntax layer are constrained by token and sentence annotations: in UD, valid heads must be tokens within the same sentence as the child token. This complicates the process of programmatically applying changes to multilayer data: for example, if an existing sentence is split, any head attachments that span the new sentence boundary must be removed, or else some tokens will have invalid heads.

For issues such as this, where a change in a "lower" layer could render existing annotations in "higher" layers ill-formed, our general approach is to perform the smallest number of adjustments necessary in order to arrive at a valid state. For example, in the situation just described where a dependency syntax layer is affected by a sentence split, we choose to nullify any head attachments which span the new sentence boundary, ensuring that the tree will remain valid, albeit incomplete. (Note however that if an NLP service is registered for dependency syntax, the new sentences will soon receive new parses from the service.) Analogous operations are implemented for other layer interactions which ensure that data in the system will avoid invalid states.

3.2. Frontend System[9]

Our frontend system provides a UI for performing our active learning workflow on a subset of CoNLL-U annotation types. Specifically, we support read/write as well as active learning support for sentence boundaries, HEAD/DEPREL, XPOS, and UPOS and currently read/write only support for LEMMA. We also support querying and ordering documents according to the number of probable annotation errors in a document, as identified by proportion of gold annotations (Figure 3) or NLP model output probabilities for a given type of annotation. Specifically, with regard to the output probabilities, given a document D with tokens $t_1, \ldots, t_n$ and annotations $a_1, \ldots, a_n$ for a given layer, and given a probability distribution over possible annotations on

[9] `https://github.com/gucorpling/midas-loop`

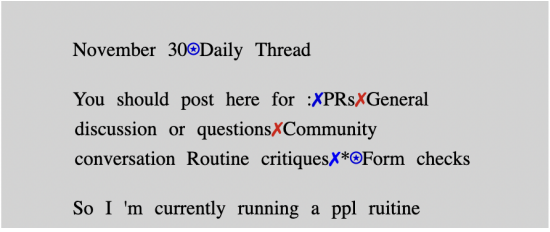

Figure 1: Segmentation interface: ✗ indicates a sentence split; ⊙ indicates that a space is not a sentence split. Red indicate a suspicious position for annotator inspection, while blue indicates edits by the user.

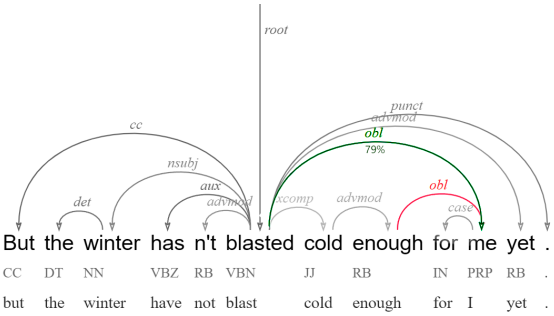

Figure 2: The syntax interface showing a suspicious annotation in red, and a high-confidence corrected annotation in green. Each suspicious annotation is shown to the user, who can determine which annotation to keep.

that layer at each position i, $\mathrm{P}(A_i = a_i|D)$, we compute $\frac{1}{n}\sum_1^n \max_{a_i} \mathrm{P}(A_i = a_i|D)$, i.e. the average probability of the most likely label at each position for the entire document. This information is used in aggregate, and there is currently no functionality for querying for documents with specific likely error types.

We have two different interfaces at the document level: one interface is for handling segmentation boundaries (Figure 1), and the other handles all remaining supported annotation types, i.e. tree by tree editing of UD data (Figure 2).[10] A third annotation UI for entities and coreference is currently being developed.

4. Evaluation

4.1. Data and setup

In order to evaluate the effectiveness of the prioritized sentence split corrections completed in Midas Loop for this case study, we used data corrected for sentence splits from the AMALGUM corpus to supplement the training data of GUM (Georgetown University Multilayer corpus (Zeldes, 2017)), the smaller human annotated English web corpus on which AMALGUM is based. The auto-annotated AMALGUM corpus itself is considered silver data, while the sentence split corrections completed in Midas Loop are considered gold

[10] We would like to credit Gerdes (2013) for the look and feel of the dependencies interface, which re-implements the graphical style of the annotation Arborator tool.

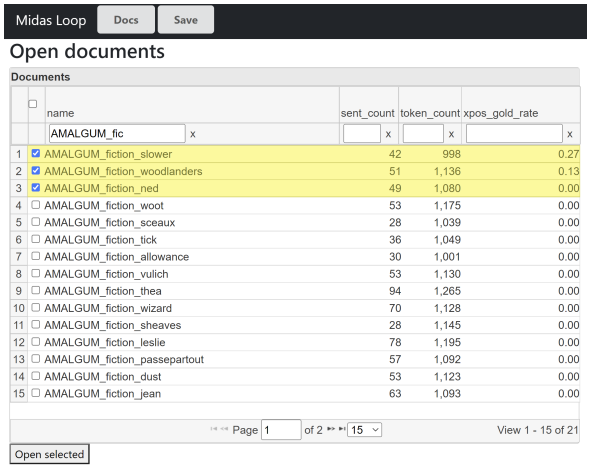

Figure 3: The document selection interface, which is used to query documents for annotation correction.

data. GUM is entirely human annotated and is thus considered to be composed of entirely gold data.

While sentence splitting has not enjoyed as much attention as syntactic or semantic analysis, and is sometimes regarded as an easy or solved task, even recent results on its accuracy in unseen data indicate that it is highly challenging, with f-scores on the GUM test set ranging from 86.35 (Stanza, (Qi et al., 2020)), to 91.60 (Trankit, (Nguyen et al., 2021)) to 93.5 (GumDrop, (Yu et al., 2019)).[11] At the same time, incorrectly split sentences by definition result in incorrect syntax trees, malsegmented discourse parses and potentially cut off entities or mentions for coreference resolution, meaning that it is a high priority to start the multilayer annotation process with high accuracy splitting.

Within the AMALGUM corpus, 10 documents of the highest priority for correction were chosen from each of the 8 genres included in the corpus: academic, biography, fiction, forum, how-to, interview, news, and travel. To determine the documents most in need of correction, each document of the AMALGUM corpus was run through a transformer based, shingled sentence splitter, which applies tokenwise binary classification to overlapping spans of 20 tokens in an attempt to find split points. The splitter is implemented using flair (Akbik et al., 2019) as an LSTM-based sequence tagger fed by transformer word embeddings encoded by the pretrained English `bert-base-cased` model.

The splitter's confidence score (0–1) on whether or not there was a sentence split at the proceeding space was recorded for each token: we say that a space needs to be examined by a human annotator if it precedes a token with a recorded confidence threshold of under 0.9. The document with the highest count of instances in need of human inspection, normalized by the token length of the document, is designated as the document of highest priority for correction. The 80 AMALGUM documents identified by prioritization, containing approximately 68K tokens, were divided amongst three human annotators and their sentence splits were corrected.

[11] These numbers are not perfectly comparable, since different papers have used different release versions of the UD dataset, but they give an idea of the challenging nature of the task.

Metric	ALL	POS.
Raw agreement	0.9965	0.9660
F_1 score	0.9827	0.9827
Cohen's κ	0.9808	—

Table 1: Microaveraged agreement for 8 documents, considering either ALL tokens or only the positive split class (POS., no credit for correct negatives)

We assess the quality of our gold sentence split annotations by double-annotating one document out of the 10 for each genre, for a total of 8 double-annotated documents. Sentence split annotation is treated as a binary sequence tagging task, where the token at the beginning of each sentence is given the positive label ("B") and all other tokens are given the negative label ("O"). We report our scores in Table 1, including the measures for raw tokenwise agreement (% tokens where both annotators made the sae decision), mutual F1 score (the F1 score, taking one annotator as gold and the other as the prediction), as well as Cohen's Kappa. Overall, our agreement measures indicate very high consistency in our gold sentence split annotations.

4.2. Results

Due to non-deterministic GPU behavior, we report 5-run averages for splitting scores on each genre (as is common practice, we use positive class F1, with no credit for the very common correct negative class), as well as the cross-genre macro average and the instance-based micro average, which can differ since some genres have substantially more splits per document, as well as different distributions of longer or shorter sentences per document. Results are broken down into several scenarios: first, we compare the use of just the gold standard GUM corpus as training data versus adding the AMALGUM data from the active learning corrections to training. Second, because AMALGUM is a multilayer corpus which includes information about TEI XML tags in the source data, such as paragraphs, headings, bulleted lists and more, this information can easily be used to improve sentence splitting accuracy (Gessler et al., 2020) – for example, sentences are usually assumed not to cross paragraph boundaries, or run on from headings into subsequent text. We therefore compare the effect of adding data both 'ex situ' in splitting from plain tokenized text, and 'in situ', with access to the XML tags, which represents a more realistic but also easier scenario for our use case.

Figure 4 shows the results, with boxplots for the spread of scores across genres, without active learning data (red) and with it (teal). We note that in the

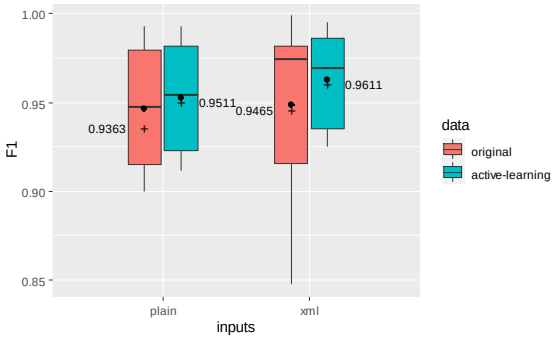

Figure 4: Sentence splitting results for 8 genres in GUM's test set when training on GUM, with and without added AMALGUM data from active learning. In the XML scenario, XML tags are used to prevent sentences crossing paragraphs and other block elements. Crosses and their labels indicate micro-averages and dots mark the 8-genre macro-averages.

plain text scenario, micro-averaged accuracy improves by ∼1.5%, which is substantial when scores are already in the mid-90s, corresponding to a 23% reduction in errors. Adding XML block information, which prevents sentences crossing paragraphs, headings, etc., improves both scenarios almost exactly by 1%, leading to a realistic sentence splitting accuracy score of 96.11%, an extremely high score compared to scores reported by systems on past versions of GUM (to the best of our knowledge, the plain text score, too, constitutes a new SOTA result on any version of GUM). Although scores are relatively close, the difference is highly significant for all contrasts ($p < 0.01$) across all 5 runs (the added XML or AL scenarios never underperform scenarios without them, in any of the five runs).

We also note that the active learning-enhanced data leads to increased stability across genres in both scenarios (less variance), with a noticeable instability in the unenhanced XML scenario. Qualitative analysis shows that the instability is caused by unfortunate split decisions across block elements in both Reddit and news data, whose elimination by the XML boundaries creates extremely long unsplit sentences. These contexts result from the noisiness and lack of punctuation in user-generated content on Reddit, and oddities of headline syntax, captions and other 'news-speak', which are common in the news genre (Bostan et al., 2020). It appears that the active learning data, which was selected to reflect contexts that models were uncertain about in each genre, prevents some of these errors and leads to more consistent scores, across the 5 runs on average.

We also review the annotator corrections made to the AMALGUM data in order to determine how effectively we identified documents that were of high priority for annotators to review. Table 2 shows the proportions of token boundaries flagged for review as well as proportions of boundaries that were changed by annotators during review. As 46.78% of flagged sentence splits were identified as false positives by the annotators reviewing the documents, we note that the cases highlighted for review were truly non-obvious cases that the splitter could not reliably predict and as such needed to be reviewed by a human annotator. We also note that nearly all of the necessary changes in the documents were correctly flagged for review, as only 1.82% of non-flagged sentence splits were additionally identified by annotators as false positives. Looking at Table 2, we see a similar picture on a smaller scale when we look at the non-sentence split spaces flagged as possible false negatives for review.

Splits	
Splits flagged	29.85%
Flagged splits merged	46.78%
Non flagged splits merged	1.82%
Spaces	
Spaces flagged	0.89%
Flagged spaces split	24.14%
Non flagged spaces split	0.47%

Table 2: Proportions of token boundaries that were flagged for review and proportions of changes that were made by annotators during review.

5. Conclusion

In this paper we presented Midas Loop, a collaborative multilayer corpus annotation system built specifically for active-learning-guided, iterative correction of automatically annotated data analyzed across different and interdependent annotation types representable in the CoNLL-U format. By using the system, we were able to improve annotation quality for the challenging and fundamental task of sentence splitting, whose accuracy is a prerequisite for subsequent annotation layers affected by sentence level decisions, such as dependency annotation, NER, coreference resolution and discourse parsing.

Our results on sentence splitting indicated that the system was effective in suggesting documents which were likely to contain many errors, and that the potential error positions identified by the system were indeed likely to require correction (about half of the time) and contained almost all positions requiring correction (over 98% in this case). Re-training our sentence splitter using the added AL-selected data proved highly effective, resulting in new SOTA scores on sentence splitting with and without XML tag information, and bringing substantial error reductions and cross-genre stability in every scenario tested.

Our future plans for the system include adding more annotation functionality, and especially support for discourse level annotations covered by the AMALGUM corpus, such as coreference resolution and the annotation of associated mentioned entities, as well as support for full document discourse parsing. We plan to leverage the existing, separate annotation tools used to annotate the original GUM corpus, but which do not currently offer good integration for multilayer interactions and

active learning. These include the GitDox (Zhang and Zeldes, 2017) editor's Spannotator widget[12] and the discourse annotation interface of rstWeb (Zeldes, 2016).[13]

6. Bibliographical References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Ashrafi Asli, S. A., Sabeti, B., Majdabadi, Z., Golazizian, P., Fahmi, R., and Momenzadeh, O. (2020). Optimizing annotation effort using active learning strategies: A sentiment analysis case study in Persian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2855–2861, Marseille, France. European Language Resources Association.

Bai, G., He, S., Liu, K., Zhao, J., and Nie, Z. (2020). Pre-trained language model based active learning for sentence matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1495–1504, Barcelona, Spain (Online).

Baldridge, J. and Osborne, M. (2004). Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 9–16, Barcelona, Spain.

Bostan, L. A. M., Kim, E., and Klinger, R. (2020). GoodNewsEveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1554–1566, Marseille, France. European Language Resources Association.

Burchardt, A., Padó, S., Spohr, D., Frank, A., and Heid, U. (2008). Formalising multi-layer corpora in OWL DL - lexicon modelling, querying and consistency control. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 389–396, Hyderabad, India.

Burnard, L. and Bauman, S., (2008). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*.

Calhoun, S., Carletta, J., Brenier, J., Mayo, N., Jurafsky, D., Steedman, M., and Beaver, D. (2010). The NXT-format Switchboard Corpus: A rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4):387–419.

Druskat, S., Bierkandt, L., Gast, V., Rzymski, C., and Zipser, F. (2014). Atomic: An open-source software platform for multi-layer corpus annotation. In Josef Ruppenhofer et al., editors, *Proceedings of KONVENS 2014*, pages 228–234, Hildesheim, Germany.

Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., and Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Ein-Dor, L., Halfon, A., Gera, A., Shnarch, E., Dankin, L., Choshen, L., Danilevsky, M., Aharonov, R., Katz, Y., and Slonim, N. (2020). Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online.

Gerdes, K. (2013). Collaborative dependency annotation. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 88–97, Prague, Czech Republic, August. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Gessler, L., Peng, S., Liu, Y., Zhu, Y., Behzad, S., and Zeldes, A. (2020). AMALGUM - a free, balanced, multilayer English web corpus. In *Proceedings of LREC 2020*, pages 5267–5275, Marseille, France.

Ide, N., Baker, C., Fellbaum, C., and Passonneau, R. (2010). The Manually Annotated Sub-Corpus: A community resource for and by the people. In *Proceedings of ACL 2010*, pages 68–73, Uppsala, Sweden.

Karamcheti, S., Krishna, R., Fei-Fei, L., and Manning, C. (2021). Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online.

Klie, J.-C., Bugert, M., Boullosa, B., de Castilho, R. E., and Gurevych, I. (2018). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics, June. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Krause, T., Lüdeling, A., Odebrecht, C., and Zeldes, A. (2012). Multiple tokenizations in a diachronic corpus. In *Exploring Ancient Languages through Corpora*, Oslo.

Lai, C. and Bird, S. (2004). Querying and updating treebanks: A critical survey and requirements analysis. In *Proceedings of the Australasian Language Technology Workshop*, pages 139–146, Sydney.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of SIGIR '94*, Dublin.

[12] See `https://corpling.uis.georgetown.edu/gitdox/spannotator.html`

[13] `https://corpling.uis.georgetown.edu/rstweb/info/`

Li, B. Z., Stanovsky, G., and Zettlemoyer, L. (2020). Active learning for coreference resolution using discrete annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8320–8331, Online.

Lin, B. Y., Lee, D.-H., Xu, F. F., Lan, O., and Ren, X. (2019). AlpacaTag: An active learning-based crowd annotation framework for sequence tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 58–63, Florence, Italy.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Nguyen, M. V., Lai, V., Veyseh, A. P. B., and Nguyen, T. H. (2021). Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1):71–106.

Peng, S. and Zeldes, A. (2018). Validating and merging a growing multilayer corpus – the case of GUM. In *14th American Association of Corpus Linguistics Conference (AACL 2018)*, Atlanta, GA.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Reiter, N. (2018). CorefAnnotator - A New Annotation Tool for Entity References. In *Abstracts of EADH: Data in the Digital Humanities*, December.

Sanguinetti, M., Cassidy, L., Bosco, C., Özlem Çetinoğlu, Cignarella, A. T., Lynn, T., Rehbein, I., Ruppenhofer, J., Seddah, D., and Zeldes, A. (2022). Treebanking user-generated content: a UD based overview of guidelines, corpora and unified recommendations. *Language Resources and Evaluation*.

Santos, D. and Mota, C. (2010). Experiments in human-computer cooperation for the semantic annotation of Portuguese corpora. In *Proceedings of LREC 2010*, pages 1437–1444, Valletta, Malta.

Searle, T., Kraljevic, Z., Bendayan, R., Bean, D., and Dobson, R. (2019). MedCATTrainer: A biomedical free text annotation interface with active learning and research use case specific customisation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 139–144, Hong Kong, China.

Shen, Y., Yun, H., Lipton, Z., Kronrod, Y., and Anandkumar, A. (2017). Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada.

Tyers, F. M., Sheyanova, M., and Washington, J. N. (2018). UD annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 10–17.

van Gompel, M. and Reynaert, M. (2014). Folia: A practical XML format for linguistic annotation - a descriptive and comparative study. In *Proceedings of CLIN 2014*.

Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M., Xue, N., Palmer, M., Hwang, J. D., Bonial, C., Choi, J., Mansouri, A., Foster, M., aati Hawwary, A., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., and Houston, A. (2012). OntoNotes release 5.0. Technical report, Linguistic Data Consortium, Philadelphia.

Wiechmann, M., Yimam, S. M., and Biemann, C. (2021). ActiveAnno: General-purpose document-level annotation tool with active learning integration. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 99–105, Online.

Yu, Y., Zhu, Y., Liu, Y., Liu, Y., Peng, S., Gong, M., and Zeldes, A. (2019). GumDrop at the DISRPT2019 shared task: A model stacking approach to discourse unit segmentation and connective detection. In *Proceedings of Discourse Relation Treebanking and Parsing (DISRPT 2019)*, pages 133–143, Minneapolis, MN.

Zeldes, A. (2016). rstWeb - a browser-based annotation interface for Rhetorical Structure Theory and discourse relations. In *Proceedings of NAACL-HLT 2016 System Demonstrations*, pages 1–5.

Zeldes, A. (2017). The GUM corpus: Creating multilayer resources in the classroom. *LREC*, 51(3):581–612.

Zeldes, A. (2018). *Multilayer Corpus Studies*. Routledge Advances in Corpus Linguistics 22. Routledge, London.

Zhang, S. and Zeldes, A. (2017). GitDOX: A linked version controlled online XML editor for manuscript transcription. In *Proceedings of FLAIRS-30*, pages 619–623, Marco Island, FL.