

TextAI: Enhancing TextAE with Intelligent Annotation Support

Maximilian Grundke, Johannes Jasper, Mariya Perchyk,
Jan Philipp Sachse, Ralf Krestel, Mariana Neves

Hasso Plattner Institute
Potsdam, Germany
mariana.neves@hpi.de

Abstract

We present TextAI, an extension to the annotation tool TextAE, that adds support for named-entity recognition and automated relation extraction based on machine learning techniques. Our learning approach is domain-independent and increases the quality of the detected relations with each added training document. We further aim at accelerating and facilitating the manual curation process for natural language documents by supporting simultaneous annotation by multiple users.

1 Introduction

Faced with rapidly growing numbers of publicly available natural language documents, it is becoming increasingly difficult to extract the underlying knowledge in a structured manner. Thus, annotation of documents for the purpose of extracting this information is an important task in many research domains today. Creating these annotations is mostly done manually, even though it is a very time consuming work and requires deep understanding and domain knowledge (Hirschman et al., 2012).

TextAI¹ is a tool developed to support annotators as a first step on the way to minimize the effort of extracting information from written texts. There exists already a wide range of annotation editors (Neves and Leser, 2012). Thus, we chose not to implement an additional standalone editor, but to build our system around the TextAE² tool, which is an existing open source editor. We extended TextAE by providing additional features to manage documents of multiple users and predicting entities and relations, which can then be

adopted by users into their set of annotations. The system learns from its users and improves prediction of relations over time based on previously annotated documents. We use the biomedical domain as use case for our system and as basis for evaluation. The developed approach, however, can be used across different domains that require manual annotation. With its automated annotation suggestions and its multi-user support, TextAI introduces capabilities that are conceptually different from other annotation tools.

2 Related Work

Automatic Annotation Suggestions. Various studies on prediction of annotations confirm that automated recommendations increase the speed and improve the quality of annotations. Lingren et al. (Lingren et al., 2014) determined that automatic annotation suggestions for named entities result in 13.83% to 21.5% time saving without reducing the inter-annotator agreement (IAA) or qualitative annotator performance. Additionally, Fort and Sagot (Fort and Sagot, 2010) and South et al. (South et al., 2014) showed a significant gain in quality. Hernandez et al. (Hernandez et al., 2014) reported an improvement of non-expert annotator performance using automated named-entity recognition. The WebAnno annotation tool (Yimam et al., 2014) includes automatic suggestions for three generic structures: spans, relations, and chains. It integrates an external machine learning tool, which requires users to configure the features themselves. While this allows for domain specific optimization, it excludes non-expert users from using this functionality. By allowing definition of custom annotation labels, WebAnno increases its flexibility, however eliminates labels as a suitable machine learning feature.

¹<https://github.com/LearningToNote>

²<http://textae.pubannotation.org/>

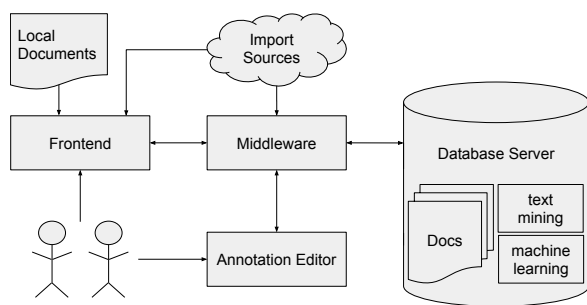


Figure 1: Components of the TextAI system.

Annotation Editors. TextAE is a browser-based annotation tool which comes without a server backend, but supports importing documents and annotations based on a simple JSON format. Being implemented using HTML and JavaScript, TextAEs functionality can be easily expanded. Furthermore, it supports both a wide range of keyboard shortcuts and usability improvements that aim to increase performance of its users. The brat rapid annotation tool (Stenetorp et al., 2012) features a client-server architecture, supports annotation of documents by multiple users simultaneously and allows comparison of different sets of annotations for one document. We chose to adapt several of these features and enhance TextAE to support them, while leaving it a simple and easy to use editor, as feature-rich tools, such as brat, quickly become complicated to use as they require complex configuration.

3 System Architecture

Annotating documents with entities and relations traditionally involves manually highlighting entities in the document and marking relations between them. Our system employs machine learning techniques to automate these steps by using the workflow depicted in Figure 1. The user interacts with a central front-end component, which is used to manage annotation tasks, documents and users, as well as importing documents from multiple sources. Users also interact with the annotation editor for editing and reviewing documents and their annotations. In the back end, we use an in-memory database (IMDB), which provides document storage, text analysis features and integration with machine learning algorithms. The middleware layer mediates between the different interfaces and the IMDB and deals with all logic concerning user and document management.

3.1 Front end

The user interacts with a central front-end component, which allows importing documents from multiple sources, such as the local file system, e.g., files in the BioC file format (Comeau et al., 2013) or by querying PubMed³, based on plain text. After importing documents into TextAI, these can be loaded from the database (cf. Back end). Besides document storage, the IMDB further offers basic text mining and analysis features that we use and expand. Domain independence is achieved through the introduction of another level of hierarchy called "tasks", which can be used to organize documents under one semantic group. This has the technical implication that each task has its own machine learning models, hence annotations within one task have no impact on predictions in another one.

3.2 Annotation Editor

Users can review and edit documents and their annotations through the annotation editor. The later also allows to trigger NER, relationship prediction, and other methods aimed at improving annotation quality and speed. We extended TextAE, which is a powerful standalone annotation tool based on JSON-formatted input, that allows loading text and creating annotations and relations. It allows adding multiple annotations to each position in the document and displaying different layers of annotations through color coding. Instead of displaying different kinds of annotations, such as POS tags and domain-specific information, e.g., medical terms, we use different colors for annotations made by different users on the same document. Because of the HTML span-tag-based implementation for annotation rendering, TextAE displays overlapping annotations in a stacked way instead of inline.

Users can create custom labels for their annotations in TextAI. However, we ask the users to map their custom labels to one of the UMLS semantic types⁴, in order to improve the ability of our system to learn based on annotations made by users and to normalize the annotations made by different users and tasks. For instance, when annotating the DDI corpus (Herrero-Zazo et al., 2013), users could create four labels, one for each drug type

³<http://www.ncbi.nlm.nih.gov/pubmed>

⁴<https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml>

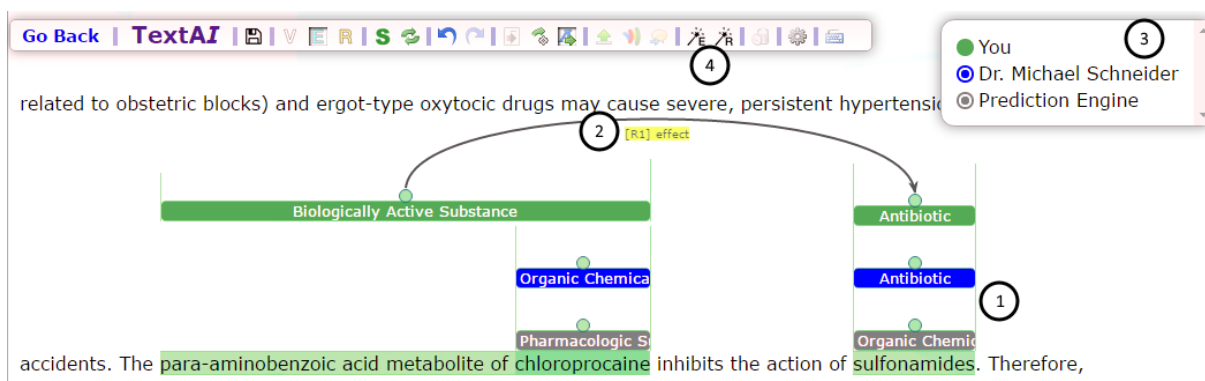


Figure 2: Screen-shot of the TextAI annotation editor showing annotations from different users (1), a drug-drug-interaction relation (2), the user-defined label (3) and the prediction functionality (4).

included in this corpus, but they are asked map every type to a UMLS semantic type, for instance, "T2000 - Clinical Drug" or "T121 - Pharmacologic Substance". Currently, we only allow mapping a label to one single UMLS semantic type.

Annotating large corpora is a task that can involve multiple experts, which requires user management to be available in a collaborative annotation system. A problem that occurs when multiple people do the same work is the creation of annotations with similar, but not identical meaning. Therefore, our system nudges users towards better annotations by asking them to map their labels to the UMLS semantic types. Further, during annotation, users can choose to hide their annotations from both other users and the machine learning part of our system, while still being able to use all of its features.

Users can correct the predictions made by the system by selectively adding them to their own annotation sets and editing them if necessary. At any point in the process, users can also manually add, remove, and alter entities and relations to their own set.

3.3 Back end

We use Rserve⁵, which provides an interface to the statistical computing language R with its extensive text mining and machine learning capabilities. Further, our system relies on the text analysis functionalities of an IMDB for entity and relation predictions. Besides the documents, task-specific metadata, such as trained models, domain specific stopword lists and NER dictionaries are stored in the database, allowing fast access to im-

⁵<https://rforge.net/Rserve/>

portant domain-dependent information.

3.4 Information Extraction

The automated suggestion of annotations involves two major steps, NER and relation prediction. For NER we apply a simple dictionary-based approach using the UMLS dictionary and part-of-speech (POS) tagging of the documents.

We train two support vector machines⁶ to detect relations between two entities: one to determine whether a relation exists between two entities, and in case of positive, one to classify the type of relation. The relations are represented by n-dimensional feature vectors. We get descriptive characteristics of the relation of two entities by combining lexical and syntactical features. In the pre-processing step, the document text is split in sentences and every word is tokenized, lemmatized and POS-tagged. Similar to the approach described in (Bui et al., 2014), we divided the sentence into three groups by adding the prefix "b" to each token appearing before the first entity, the prefix "i" to the words in between the two entities and the prefix "a" to the words after the second entity. While we remove stop words and ignore entities in the context, we do not filter out punctuation, i.e. comma, colon and semicolon, since their appearance between two entities can be a strong indicator that there is no relation between these entities. We only consider three tokens on either side of each entity in order to emphasize the near context around the annotations. The annotated types of both entities are considered as a feature as well. We also include the distance, i.e. the number of

⁶e1071 (<https://cran.r-project.org/package=e1071>), an R interface to LIBSVM with its default RBF kernel

words and the number of characters between two entities, as a feature. We avoid using trigger words as proposed by (Bui et al., 2014), since this would contradict with our domain-independence objective.

Every newly annotated document is used to re-train the models for relationship extraction, thus learning over time to improve its performance. Further, using the UMLS predefined set of types also improves the learning capabilities of our system, as entities can now be chosen from a finite set and the entity type becomes a stronger feature for relation prediction. Since training these models is only possible and reasonable given a certain amount of information, TextAI needs a set of pre-annotated documents, either by importing or manually annotating a few documents of the corpus in advance.

3.5 Middleware

As both database and front-end components have highly independent schemata and interfaces, transforming data between them is a key role of the middleware. The middleware implements a RESTful interface representing users, tasks, documents, user-documents and their content to pass on information about data objects to the browser-based front end. Additionally, user management and access rights management are handled here. An integrated user model ensures that every action taken is properly authenticated and authorized. Our annotation editor can load and export data provided in JSON format, thus the middleware is also responsible for transforming information between the different representations in our system. In addition, the middleware provides an interface to import and export documents. Internally, we use the BioC format (Comeau et al., 2013), as it is general enough to be used across different annotation domains.

4 Experiments

We focused on the medical domain and used the DDI corpus (Herrero-Zazo et al., 2013) to evaluate the performance of our NER and relation extraction procedures.

NER. In our evaluation, we consider not only exact matches of the gold-standard and the predicted entities, but also overlapping entity label markers are accepted. Our average F-1 score was 77.3% with a recall of 85.62% and a precision of

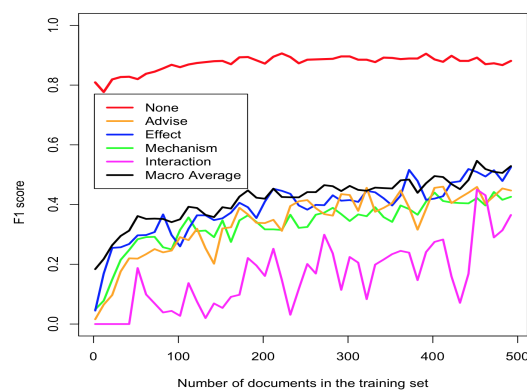


Figure 3: Influence of training set size on prediction performance.

72.53% for the DDI-Drugbank and DDI-Medline data set.

Relation Extraction. In total, the DDI corpus names five different relation types: non-relation, mechanism, effect, advise and a general interaction. We achieved a precision of 72.95% on the test set of the DDI corpus, averaged over 10 iterations. Other researchers who performed relation prediction on the DDI corpus achieve comparable performance results of 60.9% and 62.99% macro averaged F-score (Thomas et al., 2013a).

Training Set Size. When users annotate new relations, the systems performance increases since the SVM models are retrained with new user input. We measure the impact of this input on the efficiency of our classifier with a setup as follows. Starting with only one annotated document, we train both SVM classifiers on the given data and evaluate on 20 randomly selected documents. We then select 10 more documents for training and again test on 20 test documents. This cycle repeats until 500 documents are included in the training set. Figure 3 illustrates the F-score for each of the DDI interaction types averaged over 10 runs with varying training documents.

In comparison to our performance for other relations, the "interaction" type obtained lower results and these have oscillated over our experiments. This relation type is under-represented in the DDI corpus as it constitute only 6% of the annotated DDI relations. As discussed in previous work (Thomas et al., 2013b), this resulted in lower performance of the systems for this relation type on the test set. We believe that the oscillation on the results occurs for those test sets which con-

tained more or less instances of the "interaction" relation type that could not be correctly detected by our system.

5 Conclusions and Future Work

We have presented a prototype that extends the annotation editor TextAE with multi-user functionality and annotation prediction. This was achieved by creating a concept of per-user annotation sets and tasks, as well as a NER framework and relation prediction algorithm. Our system provides users with functionality for annotations prediction without interfering with their day-to-day annotation work.

As future work, we plan on conducting a user study on annotation speed and quality. Further, we also want to explore NER algorithm based on machine learning and on the labels which are normalized to the UMLS semantic types and not only on the current dictionary-based approach. Finally, semi-supervised learning approaches, such as active learning to leverage user feedback, could improve NER and relation extraction even further.

References

- Quoc-Chinh Bui, Peter M.A. Sloot, Erik M. van Muligen, and Jan A. Kors. 2014. A novel feature-based approach to extract drug-drug interactions from biomedical text. *Bioinformatics*.
- Donald C. Comeau, Rezarta Islamaj Doan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, Alfonso Valencia, Karin Verspoor, Thomas C. Wieggers, Cathy H. Wu, and W. John Wilbur. 2013. Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, 2013.
- Karèn Fort and Benoît Sagot. 2010. Influence of pre-annotation on pos-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop, LAW IV '10*, pages 56–63, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Andres M Hernandez, Harry S Hochheiser, John R Horn, Rebecca S Crowley, and Richard D Boyce. 2014. Testing pre-annotation to help non-experts identify drug-drug interactions mentioned in drug product labeling. In *Second AAAI Conference on Human Computation and Crowdsourcing*.
- María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics (JBI)*, 45(5):914–920, 10.
- Lynette Hirschman, Gully A. P. C Burns, Martin Krallinger, Cecilia Arighi, K. Bretonnel Cohen, Alfonso Valencia, Cathy H. Wu, Andrew Chatr-Aryamontri, Karen G. Dowell, Eva Huala, Anlia Loureno, Robert Nash, Anne-Lise Veuthey, Thomas Wieggers, and Andrew G. Winter. 2012. Text mining for the biocuration workflow. *Database*, 2012.
- Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. 2014. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association*, 21(3):406–413.
- Mariana Neves and Ulf Leser. 2012. A survey on annotation tools for the biomedical literature. *Briefings in bioinformatics*, page bbs084.
- Brett R. South, Danielle Mowery, Ying Suo, Jianwei Leng, scar Ferrndez, Stephane M. Meystre, and Wendy W. Chapman. 2014. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *Journal of Biomedical Informatics*, 50:162 – 172. Special Issue on Informatics Methods in Medical Privacy.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April. Association for Computational Linguistics.
- Philippe Thomas, Mariana Neves, Tim Rocktäschel, and Ulf Leser. 2013a. Wbi-ddi: Drug-drug interaction extraction using majority voting. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 628–635, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
- Philippe Thomas, Mariana Neves, Tim Rocktäschel, and Ulf Leser. 2013b. Wbi-ddi: drug-drug interaction extraction using majority voting. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 2, pages 628–635.
- Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in webanno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96. Association for Computational Linguistics.