SenTag: A Web-Based Tool for Semantic Annotation of Textual Documents

Andrea Loreggia,[1*] **Simone Mosco,** [2] **Alberto Zerbinati** [2]

[1] European University Institute
[2] University of Padova
andrea.loreggia@gmail.com, simone.mosco@studenti.unipd.it, alberto.zerbinati@studenti.unipd.it

Abstract

In this work, we present SenTag, a lightweight web-based tool focused on semantic annotation of textual documents. The platform allows multiple users to work on a corpus of documents. The tool enables to tag a corpus of documents through an intuitive and easy-to-use user interface that adopts the Extensible Markup Language (XML) as output format. The main goal of the application is two-fold: facilitating the tagging process and reducing or avoiding errors in the output documents. It allows also to identify arguments and other entities that are used to build an arguments graph. It is also possible to assess the level of agreement of annotators working on a corpus of text.

Introduction

In the scope of Natural Language Processing (NLP), many applications necessitate of a great amount of annotated documents in order to train machine learning (ML) systems (Quintarelli et al. 2019). Group of experts are employed to manually annotate documents in order to identify important details that can be used to train artificial intelligence models (Poudyal et al. 2020). In this preliminary step, a predefined structures language is adopted, such as Extensible Markup Language (XML) or JSON. Manually annotated datasets play an important role in the definition of gold standards. For instance, such datasets might be used for training machine learning tools to extract argument-related information from case texts (Ashley 2017), to predict the outcome of a sentence (e.g., (Medvedeva et al. 2020)), or to train machine learning systems able to summarize the original text (e.g., (Xu, Savelka, and Ashley 2020)). Unfortunately, the manual process is prone to errors. For instance, annotators may introduce non-existing or misspelled tags. Thus, annotators have to spend time for checking the consistency and the validity of their work. In this work, we present SenTag, a web-based application that provides an intuitive interface for document annotation. The application allows multiple users to work on the same corpus of documents. Users can belong to 3 different groups (i.e., admins, editors, and annotators) based on the role that each user should play. Users that belong to the admin group are in charge of creating other users called editors and annotators. Editors upload XML schemas, documents, and specify which schema an annotator will use to annotate a document. At the end of the tagging phase, an annotator has to validate an annotated document against the XML schema. In case of errors (e.g., attributes with missing values), the annotator will be alerted. Otherwise, the resulting document complies with the schema and thus the tagging process produced a well-formed document. Moreover, the application allows the use of specific tags to identify and easily build a graph of arguments: annotators can use these tags to identify different arguments in a document and define relationships among them. To the best of our knowledge, no existing platform implements all these features altogether.

Related Work. The use of machine learning techniques shows their potential in getting better performance in this area (Slonim et al. 2021). These systems still strongly depend on manually tagged datasets and on the ability of human annotators. Unfortunately, obtaining good manual annotations is not an easy task. That is why some approaches try to remove the domain expert from the loop (Loreggia et al. 2016). To facilitate this task, some systems were developed over the last few years. Due to lack of space, we report a non-exhaustive list of works and we point the reader to a complete review on the topic (Ashley 2017)): Gloss is an annotation system developed by researchers from the University of Pittsburgh, it leverages on some individual components to assist the user during the whole annotation process, including corpus assembly, type system definition, document annotation, as well as quality control. NER Annotator[1] is another web-based tool which provide a graphical user interface that helps users to annotate documents and generates traning data as a JSON which can be readily used. Unstructured Information Management applications (UIMA) is a suite of software systems developed to analyze large volumes of unstructured information (Ferrucci and Lally 2003). The suite provides a developer's toolbox software that also includes an annotation interface. GATE Teamware is an open-source, web-based, collaborative text annotation framework. It enables users to carry out complex

[*]A. Loreggia has been supported by the H2020 ERC Project "CompuLaw" (G.A. 833647).

[1]https://github.com/tecoholic/ner-annotator - Last accessed 27th July 2021

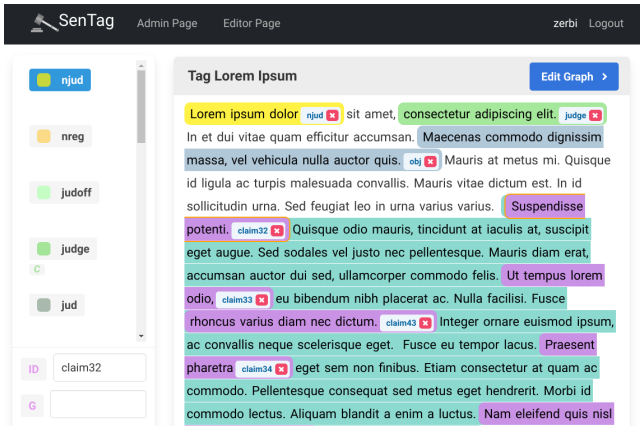

Figure 1: The annotation interface

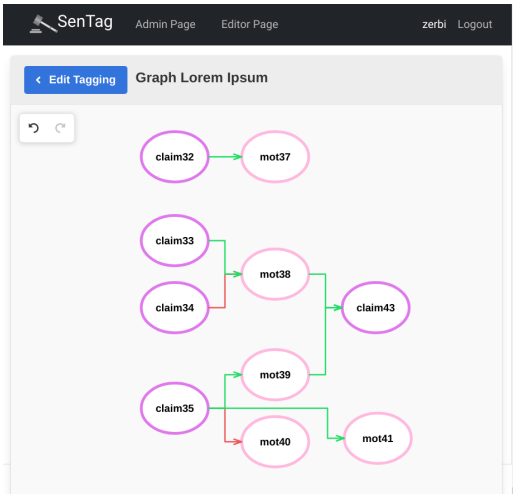

Figure 2: The graph interface

corpus annotation projects, involving distributed annotator teams (Bontcheva et al. 2013). WebAnno is a generic web-based annotation tool for distributed teams (De Castilho et al. 2016).

SenTag

SenTag is developed in Python 3.9 as a Django 3.2 app employing VUE 3 for the graphical interface. The application is based on NER Annotator, from which it derives and expands the tagging part [2] The main strengths of the application are: a) security and user management, b) multiple annotators and agreement score, c) intuitive interface for text annotation, d) graph of arguments. All these features allow to build XML documents avoiding annotators for a deep understanding of the language and thus reducing possible errors.

Security and User Management. The application allows to group users into three different categories: a) **admin**: users with the highest level of rights. Admins can create other users and assign them to a specific group. They also have the rights to do all the tasks allowed to editors; b) **editor**: this type of users is in charge of: (i) uploading texts and XML schema; (ii) assigning texts to the associated schema; (iii) assigning annotators to texts they should annotate; (iv) check for annotators agreement. Editors also have the rights to perform all the tasks allowed to annotators; c) **annotator**: this type of users is in charge of tagging documents that are assigned to them using the set of tags specified in the schema associated to each document. Annotators can also validate their work against the XML schema after completing the annotation phase and build the graph of arguments based on the arguments identified in the document.

Multiple Annotators and Agreement Score. Multiple annotators can be assigned to a document. In this case, the interface will report statics about the quality and the validity of the annotation performed on the document. For each document, the Krippendorff's alpha (Krippendorff 2011) is reported describing the level of agreement for the group of annotators working on the document. Moreover, for each annotator, the interface reports whether he/she has completed the annotation for a document. It also reports whether each annotated document passed the validity check against the XML schema.

Text Annotation. Annotators are in charge of the annotation of documents. After being authenticated, an annotator is proposed with a list of documents assigned to him/her. Documents are grouped based on whether the annotation phase is completed and thus validated. The annotation is done using the interface depicted in Figure 1. The screen is divided into two different areas. On the left side, all the available tags (and their attributes) are reported, this enables the annotator to choose which tag to use. When a tag is selected, the list of its attributes appears. This allows the annotator to change (if she wants) the value for a specific attribute or simply to consult it for future purposes. After clicking on a tag, the annotator can select part of the text which will be highlighted with the color associated to the tag. This would serve as a reminder for the annotator, the name of the tag associated to a part of the text is always visualized.

Graph of Arguments. The platform allows to build the graph of arguments from the tagged document. The XML schema might contains tags with the specific attribute GRAPH. When part of the text is tagged with one of these tags, automatically a node appears in the graph. A dedicated area allows annotators to draw edges between nodes in order to describe relationships among arguments. Any time a node is connected to another one, the list of ancestors and descendants of the nodes are adjusted with the ids of the selected nodes. This allows to enrich the final XML document by adjusting the correspondent attributes in the output file. Figure 2 depicts the graphical interface for graph editing.

Conclusion

We presented SenTag a new lightweight web-based tool focused on semantic annotation of textual documents. As future work, we plan to embed into the system artificial intelligence techniques (such as ensemble methods (Cornelio et al. 2021)) with the goal of automating some parts of tagging.

[2] Available at https://github.com/AlbertoZerbinati/sentag

References

Ashley, K. D. 2017. *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press.

Bontcheva, K.; Cunningham, H.; Roberts, I.; Roberts, A.; Tablan, V.; Aswani, N.; and Gorrell, G. 2013. GATE Teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4): 1007–1029.

Cornelio, C.; Donini, M.; Loreggia, A.; Pini, M. S.; and Rossi, F. 2021. Voting with random classifiers (VORACE): theoretical and experimental analysis. *Autonomous Agents and Multi-Agent Systems*, 35(2): 1–31.

De Castilho, R. E.; Mujdricza-Maydt, E.; 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)*, 76–84.

Ferrucci, D.; and Lally, A. 2003. Accelerating corporate research in the development, application and deployment of human language technologies. In *SEALTS '03: Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems*, 67–74. Morristown, NJ, USA: Association for Computational Linguistics.

Krippendorff, K. 2011. Agreement and information in the reliability of coding. *Communication Methods and Measures*, 5(2): 93–112.

Loreggia, A.; Malitsky, Y.; Samulowitz, H.; and Saraswat, V. 2016. Deep learning for algorithm portfolios. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Medvedeva, M.; Xu, X.; Wieling, M.; and Vols, M. 2020. JURI SAYS: An Automatic Judgement Prediction System for the European Court of Human Rights. In Serena, V.; Harasta, J.; and Kremen, P., eds., *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, 277–280. IOS Press.

Poudyal, P.; Savelka, J.; Ieven, A.; Moens, M. F.; Gonçalves, T.; and Quaresma, P. 2020. ECHR: Legal Corpus for Argument Mining. In *Proceedings of the 7th Workshop on Argument Mining*, 67–75.

Quintarelli, S.; Corea, F.; Fossa, F.; Loreggia, A.; and Sapienza, S. 2019. An ethical perspective on Artificial Intelligence: Principles, rights and recommendations. *BioLaw Journal*, 2019(3): 159–177.

Slonim, N.; Bilu, Y.; Alzate, C.; Bar-Haim, R.; Bogin, B.; Bonin, F.; Choshen, L.; Cohen-Karlik, E.; Dankin, L.; Edelstein, L.; et al. 2021. An autonomous debating system. *Nature*, 591(7850): 379–384.

Xu, H.; Savelka, J.; and Ashley, K. D. 2020. Using Argument Mining for Legal Text Summarization. In Serena, V.; Harasta, J.; and Kremen, P., eds., *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, 184–193. IOS Press.