Hannotate: Flexible Annotation for Text Analytics from Anywhere

Tan-Tai To[1], Hoang Dao[2], Huyen Nguyen[1], Thanh-Ha Do[1], and Tuan-Anh Hoang[⋆3]

[1] Hanoi University of Science
{totantai_ch2020,huyennt,dothanhha}@hus.edu.vn
[2] Independent Researcher, hoangdhph04904@gmail.com
[3] RMIT University Vietnam anh.hoang62@rmit.edu.vn

Abstract. Data annotation is a critical but the most expensive step in any text analytics project. There have been several frameworks built for enabling and easing this step. Most of these frameworks are however either not easy to be configured to specific users' needs, have no functionalities for annotating text pairs, or lack of efficient mechanism for data management and progress monitoring. Moreover, they have mostly no graphical user interfaces that are specifically designed for mobile devices. In this paper, we introduce Hannotate, a highly flexible, lightweight web-based framework that provides functionalities for a wide range of text annotation from both desktop and mobile devices. Our framework inherits the advantages of the typical existing ones while allowing users to easily customize the annotation work according to their demand and budget. The framework also supports users in managing data, monitoring the progress, and giving feedback to annotators.

1 Introduction

Labeled datasets are essential for any text analytics project. These datasets are however not always available at the beginning of the projects. Moreover, the available datasets, if there is any, are not always suitable for the projects, for various reasons, including mismatch in the domains, modalities, languages, and size of the datasets and the projects' requirements. Hence, annotating new datasets is a critical step for enabling the projects. This step is however time consuming and expensive, which requires efficient frameworks for distributing and crowd-sourcing the annotation work [9].

There have been a number of frameworks constructed for text annotation in different contexts and for different purposes [7]. Most of these frameworks however suffer from some of the following shortcomings:

- Limited options and not intuitive interfaces for customization. For example, several existing frameworks require complicated interactions for setting the possible labels to be assigned, and the number of annotations required for each text.
- No functionalities for annotating pairs of texts. This type of annotation is used for entailment inference [1], and semantic similarity measurement [2], which are crucial in natural language understanding applications.

[⋆] Corresponding author.

- Limited functionalities for giving feedback to annotators. This is critical for training the annotator in order to improve the quality of the annotation jobs.
- No interfaces tailored for mobile devices. This would drastically reduces the productivity of a large number of annotators who work on their handheld smart devices.

In this work, we would like to address the aforementioned shortcomings in existing frameworks by developing a lightweight, flexible one for a wide range of text annotation jobs. We aim to provide users with friendly interfaces for designing and customizing their jobs, monitoring the annotation progress, giving feedback to annotators, and managing the datasets. We also want to provide annotators with convenient interfaces for performing the annotation tasks efficiently from both desktop or mobile devices. In summary, the notable features of our framework are as follows.

- **Highly flexible**: it allows users to customize their jobs easily and intuitively
- **Highly accessible**: it is accessible from a wide range of devices with consistent user experience across the devices
- **Convenient**: it is a lightweight framework that integrates user-friendly interfaces for managing data and monitoring the progress.

In the rest of this paper, we briefly discuss the existing frameworks in Section 2. We then describe our framework in detail in Section 3. Lastly, we conclude the paper and sketch some directions for further improvement for the framework in Section 4.

2 Related Work

Among the frameworks that have been widely used in text mining communities, **brat**[4], is considerably the most popular one. This framework was is originally designed for linguistic annotation (e.g., part-of-speech tagging and named entity recognition). One can configure **brat** for non-linguistic annotation, e.g., aspect-specific sentiment analysis [8]. However, the configuration is performed through a text file, which is not intuitive. Moreover, the configuration, once set, is shared among all the projects, which do not allow different types of annotation to be performed concurrently. This shortcoming has been partially addressed in **WebAnno**[5] [3] and **INCEpTION**[6] [5] – the successors to **brat** with most recent releases. These two frameworks have major extensions for project-specific configuration and interfaces for data curation. The configuration is however still quite complicated with required sequences of interactions. Also, these two frameworks provide no mechanism for giving feedback to the annotators.

There are also existing frameworks that are more specifically designed for non-linguistic annotation, e.g., **doccano**[7], **WARP-Text** [6], **prodigy**[8], and **Label Studio**[9]. These frameworks are however limited to annotating of single texts, and therefore not suitable for tasks that require annotating of pairs of texts. Moreover, to the best of

[4] https://brat.nlplab.org

[5] https://webanno.github.io/webanno

[6] https://inception-project.github.io

[7] https://github.com/doccano/doccano

[8] https://prodi.gy

[9] https://github.com/heartexlabs/label-studio

our knowledge, there is no existing open-source frameworks that provide convenient interfaces for performing the annotation from mobile devices.

Certainly, there are commercial frameworks and platforms that provide customizable services for annotating texts, e.g., **Labelbox**[10], **CloudFactory**[11], and **Amazon Mechanical Turk**[12]. These services are however expensive while may suffer from low quality annotation and data confidentiality risks [4].

3 Hannotate

We now highlight the notable features of our framework. We start by describing the types of annotation that our framework is designed for. We then specify the users of the frameworks and the functionalities we would like to provide them. Finally, we describe the components for deploying the framework. Please refer to the extended version of this paper[13] for more detailed information on the framework's design, its enable technologies, and its implementation.

3.1 Annotation types

Currently, our framework is facilitated for the following types of annotation.

- **Single text annotation**, which includes topic labeling, and sentiment recognition.
- **Text pair annotation**, e.g., entailment inference and semantic similarity measurement.
- **Span annotation** such as key-phrase detection named entity recognition.
- **Span pair annotation** such as relational extraction and co-reference recognition.
- **Sentence rewriting**, e.g., translation and paraphasing.

3.2 User role and Functionalities

Our framework is designed to server the **job managers** – who manage dataset(s) to be annotated and the annotation process, the **annotators** – who would like to perform the annotation jobs, and the **admin** – who manages an operate the whole system. We provide the user-friendly graphical interfaces for the following functionalities for each user role. The **job managers** are supported to

- Create and customize annotation job. Here, a job will be created for each dataset to be annotated. Current options for customization includes specifying the label sets, the number of labels, and the number of annotation for each data instance.
- Approve or reject bids from annotators who register to perform the job, and assign tasks to the approved annotators. Here, a task is the annotating of a data instance, and tasks are assigned to annotators in batches. This help to better distribute the job among the annotators while managing their progress and annotation quality better.

[10] https://labelbox.com

[11] https://www.cloudfactory.com

[12] https://www.mturk.com

[13] Hannotate's full version

- Monitor the work progress of the whole project, and that of each individual annotator.
- Examine the annotated data instance, then approve or reject the annotation, and give feedback to the corresponding annotator.
- Export the annotated text and related information (including the annotators and their labels for each data instance, timestamp, etc.) to local for later use.

The **annotators** are supported to

- Search and bid for open jobs. Each annotator can bid for and perform multiple jobs concurrently.
- Perform the assigned annotation tasks and submit the result
- View feedback from the job's manager, revise and re-submit the tasks.

Lastly, the **admin** is supported to manage user accounts and to perform basic customization regarding the operating of the whole system.

3.3 Components

Our whole framework is packed into front-end and back-end components separately. Each component is indeed a stand-alone package that can be run as it. This allows each part of the framework to be deployed independently from different environments. We also provide a walk-through video presentation about our framework and an online demo. Please refer to our project's repository[14] for those components and their usage.

4 Conclusion

We have introduced Hannotate, a lightweight yet highly flexible and accessible framework for a wide range of text annotation. Our framework inherits the advantages of the existing ones while addressing their main issues by providing users with user-friendly interfaces for customizing, managing, and monitoring their annotation jobs. We also provide annotators with interfaces for annotating smoothly and consistently across devices.

Possible extensions to our framework include adding functionalities for more types of annotations and analyzing the annotation results. We would also like to add smart suggestion mechanisms for aiding annotators in performing the tasks, which have been shown to significantly improve the annotators' productivity [10].

5 Acknowledgement

This work is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2020.DA14

[14] https://github.com/smutahoang/hannotate

References

1. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: EMNLP 2015 (2015)
2. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: SemEval-2017 (2017)
3. De Castilho, R.E., Mujdricza-Maydt, E., Yimam, S.M., Hartmann, S., Gurevych, I., Frank, A., Biemann, C.: A web-based tool for the integrated annotation of semantic and syntactic structures. In: COLING 2016 Workshop (2016)
4. Fort, K., Adda, G., Cohen, K.B.: Amazon mechanical turk: Gold mine or coal mine? Computational Linguistics **37**(2), 413–420 (2011)
5. Klie, J.C., Bugert, M., Boullosa, B., de Castilho, R.E., Gurevych, I.: The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In: COLING 2018 (2018)
6. Kovatchev, V., Martí, M.A., Salamó, M.: Warp-text: a web-based tool for annotating relationships between pairs of texts. In: COLING 2018 (2018)
7. Neves, M., Ševa, J.: An extensive review of tools for manual annotation of documents. Briefings in bioinformatics **22**(1), 146–163 (2021)
8. Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis. SemEval 2014 (2014)
9. Sabou, M., Bontcheva, K., Derczynski, L., Scharl, A.: Corpus annotation through crowdsourcing: Towards best practice guidelines. In: LREC'14 (2014)
10. Yang, J., Zhang, Y., Li, L., Li, X.: Yedda: A lightweight collaborative text span annotation tool. In: ACL 2018 (2018)