POTATO: The Portable Text Annotation Tool

Jiaxin Pei [1] **Aparna Ananthasubramaniam**[2] **Xingyao Wang**[3] **Naitian Zhou** [4]
Apostolos Dedeloudis[5] **Jackson Sargent**[2] **David Jurgens**[2]

[1]**Stanford University** [2]**University of Michigan** [3]**University of Illinois Urbana-Champaign**
[4]**University of California Berkeley** [5]**The American College of Greece**

[1]pedropei@stanford.edu [2]{akananth, jacsarge, jurgens}@umich.edu
[3]xingyao6@illinois.edu [4]naitian@berkeley.edu [5]apostolosded@gmail.com

Abstract

We present POTATO, the **Po**rtable **t**ext **a**nnotation **to**ol, a free, fully open-sourced annotation system that 1) supports labeling many types of text and multimodal data; 2) offers easy-to-configure features to maximize the productivity of both deployers and annotators (convenient templates for common ML/NLP tasks, active learning, keypress shortcuts, keyword highlights, tooltips); and 3) supports a high degree of customization (editable UI, inserting pre-screening questions, attention and qualification tests). Experiments over two annotation tasks suggest that POTATO improves labeling speed through its specially-designed productivity features, especially for long documents and complex tasks. POTATO is available at https://github.com/davidjurgens/potato and will continue to be updated.

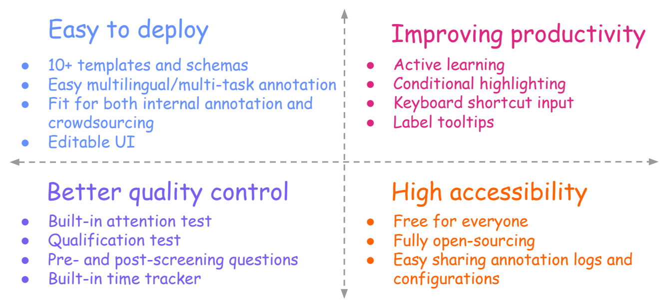

Figure 1: The four core design goals of POTATO: easy to deploy, greater productivity, better quality control, and high accessibility. Each design goal comes with a series of features that can make data annotation easier and more reliable.

Introduction

Much of NLP requires annotated data. As NLP has tried to tackle increasingly more complex linguistic phenomena or diverse labeling and classification tasks, the annotation process has increased in complexity—yet the need for and benefit of large labeled datasets remains (Halevy, Norvig, and Pereira 2009; Sun et al. 2017). Modern annotation tools like Label Studio (Tkachenko et al. 2021), LightTag (Perry 2021), Doccano (Nakayama et al. 2018), and Prodigy (Explosion 2017) have partially filled this gap, providing a variety of solutions to different types of annotations. However, these tools each bring their own challenges: requiring external access, limiting visual configurability for complex tasks, or even costing hundreds of dollars—prohibitive for small groups. We introduce POTATO, The **Po**rtable **t**ext **a**nnotation **to**ol, which allows practitioners to quickly design and deploy complex annotations tasks.

POTATO has been designed, developed, and tested over a two-year period with the following design goals in mind (Figure 1): 1) **High accessibility**. POTATO is open-sourced under the MIT license and free to anyone. POTATO is built with minimal dependencies to allow researchers and developers to easily build and integrate their own features. 2) **Easy to deploy**. POTATO comes with templates covering a wide range of annotation tasks like best-worst-scaling, text classification, and multi-modal conversation. Anyone can start a new annotation project with simple configurations. POTATO is also rapidly and easily deployable in local and web-based configurations and has seamless integration with common crowdsourcing platforms like Amazon Mechanical Turk and Prolific. POTATO flexibly supports diverse annotation needs. With our specially designed schema rendering and custom rendering mechanism, POTATO allows nearly all kinds of text annotation tasks and is visually customizable to support complex task designs and layout. 3) **Better quality control**. Attaining reliable annotations is one of the core goals of data annotation tasks. POTATO is designed with a series of features that can help to improve annotation quality, including built-in attention tests, prestudy qualification tests, and an annotation time tracker. POTATO also allows deployers to easily set up pre- and post-screening questions (e.g. demographics or psychological surveys), which can help researchers to better understand potential biases in data labeling. 4) **Productivity enhancing**. POTATO comes with a series of productivity features for both deployers and annotators like active learning, conditional highlighting, and keyboard shortcuts. While existing systems like Doccano (Nakayama et al. 2018) and Lighttag (Perry 2021) offer different subsets of these features, POTATO aims to support a holistic annotation experience by meeting all of these needs. Experiments on two annotation tasks demonstrate that POTATO leads to more efficient data labeling for complex tasks.

Architecture and Design

POTATO is written in Python and focuses on portability and simplicity in annotation and deployment. The user interface is created through an extensible HTML template and configuration file, which allows practitioners to quickly develop and deploy common setups like Likert-scale annotation while also supporting extensive display customization. The POTATO server populates the interface with data provided by the operator and supports displaying any HTML-supported modality, e.g., text or images. An overview of the architecture is shown in Figure 2.

Data Management POTATO loads data in common file formats, such as delimited files or newline-delimited JSON. This allows it to ingest data in the JSON format supplied by the Twitter or Reddit APIs, as well as other types of data used by the deployer. All formats are converted into internal data structures that link the deployer-selected instance ID to annotations. At a minimum, deployers must specify which field represents the unique instance ID and, for most tasks, the text to be annotated. The data may contain other columns which will be included in the final output and can optionally be used in customized visualizations.

Annotation Schema Rendering POTATO allows deployers to select one or more forms of annotation for their data using predefined schema types, such as multiple choice or best-worst scaling. Deployers fill out which options should be shown and then each scheme is rendered into HTML upon the completion of loading the data. Annotation instructions can be provided as an external URL that annotators may view or using HTML text shown in POTATO that annotators may collapse vertically to free up screen space. POTATO provides default HTML templates that automatically lay out each scheme's annotation questions. However, deployers may additionally customize the HTML templates and select their own layout using JINJA expressions (e.g., `{{text}}`; Jinja) to specify where parts of the annotation task and data are populated within the user-defined template.

User Management Annotators create accounts and then log in to view their tasks using a secure user management system. When used with crowdsourcing platforms, POTATO also allows workers to directly jump to the annotation task using their crowdsourcing user ID. For each new annotator, POTATO automatically assigns instances as configured by the deployer and all the annotations are recorded on the backend. When logging out and back in, annotators resume at the most recent unannotated item. POTATO also allows deployers to, with minimum configurations, set up pre- and post-screening questions (e.g., having annotators provide demographics or complete psychological questionnaires), pre-study tests, and attention tests to identify unreliable annotators.

Design highlights POTATO is designed to flexibly support diverse annotation tasks and improve the productivity of annotators. Here we briefly highlight several features of POTATO. First, with simple configurations, deployers can quickly add *keyboard shortcuts* to specific options or *tooltips* to help annotators. Second, in settings where an annotator is reading a dense or long passage, or where there are many potentially subtle cues, annotators are likely to struggle due to having to slowly and carefully read each passage or accidentally omitting relevant annotations due to the complexity of the task. POTATO introduces a new feature, *conditional highlighting*, to help in these settings, where the deployer specifies certain keywords to trigger highlights in the text, drawing the annotator's focus to those words or phrases. For example, if annotating for Twitter-based stance towards politicians, a deployer might use keywords and phrases for common politicians or political parties to ensure these are not missed. If conditional highlighting is enabled, POTATO will also randomly label some words with highlights, based on a deployer-specified rate, to ensure annotators do not rely too heavily on highlights.

Comparison with Existing Systems

POTATO has been developed to fill a key niche left by existing systems for providing visual customization, easy annotator-support features, and rapid development. The ultimate goal is to provide simple and comprehensive solutions to common annotation tasks as well as allow personalized design for complex tasks. Table 1 shows the comparisons between POTATO and other common text annotation tools over a series of important dimensions including flexibility, productivity, quality, and accessibility. We highlight major differentiators next. Please note that we only compare annotation systems that are currently available for anyone to use, free of cost.

Flexibility POTATO is designed to maximize flexibility for a variety of annotation settings. For common annotation tasks like text classification, POTATO comes with a wide range of templates and allows a quick start for deployers. However, unlike many existing annotation tools, which provide fixed user interfaces with selected types of annotation tasks (e.g., Doccano offers neither templates nor an editable UI (Nakayama et al. 2018)), POTATO allows deployers to customize their own annotation interface to support diverse needs. For example, Wang and Jurgens (2021) used animated GIFs as the labels in the annotation and Mendelsohn, Budak, and Jurgens (2021) used a 27-class scheme under three categories, both of which required visual customization to make the task feasible. POTATO also allows deployers to easily set up unlimited numbers of similar annotation tasks, which can be especially helpful for multilingual annotations. For example, for all the other data annotation systems, the deployers need to set up separate tasks and guidelines for each language. With POTATO, deployers only need to create a sheet containing translated guidelines and POTATO's built-in script can help to generate annotation sites for each language.

Productivity POTATO is designed to maximize the productivity of both annotators and deployers. While most of the existing annotation tools generally focus on labeling data, POTATO supports a series of features that can help with the entire data annotation pipeline.

POTATO allows easily-customizable keyboard shortcuts to allow efficient annotation. For visually or cognitively challenging settings, POTATO allows conditional highlights, which helps to reduce task complexity and focus annotators' attention. Finally, active learning can reduce the annotation time needed to curate an informative dataset. Often, annotation tools that offer a highly customizable annotation interface do not also implement productivity features: the

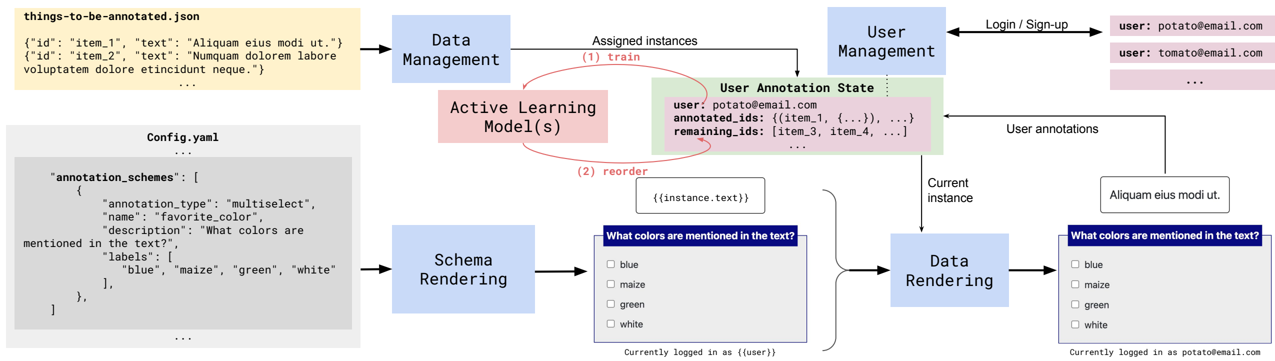

Figure 2: The overall architecture of POTATO features a modular design that decouples the task specification from the rendering, allowing rapid deployment of new task designs and separate customization of the visualization.

		Label Studio	Doccano	FLAT	LightTag	Prodigy	Tagtog	FITAnnotator	BRAT	WebAnno/INCEpTION	POTATO
Flexibility	Multiple Schema	✓			✓		✓		✓	✓	✓
	Multimodal	✓	✓		✓	✓					✓
	Span-Based Annotation	✓	✓		✓	✓	✓		✓	✓	✓
	Editable UI	✓		✓		✓		✓			✓
Productivity	Active Learning	✓*				✓		✓		✓	✓
	Conditional Highlighting										✓
	Keyboard Shortcuts	✓	✓		✓	✓			✓		✓
Quality control	Prestudy Qualification Test										✓
	Attention Test										✓
	Behavioral Tracking										✓
	Pre- and Post-screeing Questions										✓
Accessibility	Open-Source	✓	✓	✓					✓	✓	✓
	Easy Sharing and Replicating										✓
	Price	Free	Free	Free	Free for academia	$390	$59/person/month	Not available	Free	Free	Free

Table 1: Comparisons between POTATO and other text annotation systems. * means the feature is not available for the free plan.

open-sourced version of LabelStudio (Tkachenko et al. 2021) only supports keypress shortcuts, while Flat (Gompel et al. 2017) supports none of these features. For deployers, POTATO allows seamless integration with common crowdsourcing platforms like Amazon Mechanical Turk and Prolific.

Quality control Collective high-quality and reliable annotations is the ultimate goal of data labeling tasks and is usually the key to the success of the final ML/NLP systems. POTATO comes with a series of quality control feature which helps deployers to reliably collect annotations. While some other annotation systems like Label Studio and WebAnno also support agreement calculation, none of the existing systems come with features that help deployers to improve the annotation quality and analyze factors affecting it. POTATO allows deployers to easily set up prestudy qualification tests (annotators have to pass a small test to participate in the full annotation) and attention tests (attention test questions are randomly inserted in the annotation queue as configured by the deployer) to identify unreliable annotators before, within, and after annotation. POTATO also allows deployers to freely insert survey questions before and after the annotation phase. Deployers can easily define different pages of pre- and post-screening questions with minimum effort and POTATO also provides a series of templates for common survey questions like user consent and demographic information. Recent studies suggest that the background of annotators has substantial effects on the quality and bias of data labeling and further affects the fairness of ML/NLP models (e.g., Sap et al. 2022). With POTATO, researchers can easily collect background information of annotators and analyze the effect of annotator backgrounds on data labeling.

Accessibility POTATO is free to use and actively maintained. While commercial annotation tools like Prodigy (Explosion 2017) can come with more functionality, these tools are expensive; for example, Prodigy costs $390 USD for individual users, and Tagtog (Cejuela et al. 2014) costs at least $59 USD per person per month. These costs are potentially prohibitive for students and researchers without access. POTATO is fully open-sourced and is deployed with minimum dependencies. Moreover, in addition to giving the flexibility to freely define UIs and annotation settings, POTATO allows researchers to easily share their annotation settings with a simple YAML file, aiding in replication and future extension of prior work.

Conclusion and Future Plans

POTATO distinguishes itself with a comprehensive suite of productivity-enhancing features that allow annotators to efficiently and accurately label data and researchers to quickly configure complex tasks on a wide range of data types. POTATO was created both for the computational scholar and the overburdened student or crowdworker, looking to annotate more data in their limited time. POTATO has been in continuous development for over two years and will continue to be developed to support new task designs, easier management, and faster annotation.

References

Cejuela, J. M.; McQuilton, P.; Ponting, L.; Marygold, S. J.; Stefancsik, R.; Millburn, G. H.; Rost, B.; Consortium, F.; et al. 2014. TagTog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database*, 2014.

Explosion. 2017. Prodigy. https://prodi.gy.

Gompel, M. v.; Sloot, K.; Reynaert, M.; and van den Bosch, A. 2017. FoLiA in Practice. The Infrastructure of a Linguistic Annotation Format.

Halevy, A.; Norvig, P.; and Pereira, F. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2): 8–12.

Jinja. ???? JINJA Template Designer Documentation. https://jinja.palletsprojects.com/en/3.0.x/templates/.

Mendelsohn, J.; Budak, C.; and Jurgens, D. 2021. Modeling Framing in Immigration Discourse on Social Media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2219–2263.

Nakayama, H.; Kubo, T.; Kamura, J.; Taniguchi, Y.; and Liang, X. 2018. doccano: Text Annotation Tool for Human. Software available from https://github.com/doccano/doccano.

Perry, T. 2021. LightTag: Text Annotation Platform. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 20–27.

Sap, M.; Swayamdipta, S.; Vianna, L.; Zhou, X.; Choi, Y.; and Smith, N. A. 2022. Annotators with Attitudes: How Annotator Beliefs And Identities Bias Toxic Language Detection.

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, 843–852.

Tkachenko, M.; Malyuk, M.; Shevchenko, N.; Holmanyuk, A.; and Liubimov, N. 2021. Label Studio: Data labeling software, 2020-2021. *Open source software available from https://github. com/heartexlabs/label-studio*.

Wang, X.; and Jurgens, D. 2021. An animated picture says at least a thousand words: Selecting Gif-based Replies in Multimodal Dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3228–3257.