

FITAnnotator: A Flexible and Intelligent Text Annotation System

Yanzeng Li, Bowen Yu, Quangang Li, Tingwen Liu*

Institute of Information Engineering, Chinese Academy of Sciences

School of Cyber Security, University of Chinese Academy of Sciences

{liyanzeng, yubowen, liquangang, liutingwen}@iie.ac.cn

Abstract

In this paper, we introduce FITAnnotator, a generic web-based tool for efficient text annotation. Benefiting from the fully modular architecture design, FITAnnotator provides a systematic solution for the annotation of a variety of natural language processing tasks, including classification, sequence tagging and semantic role annotation, regardless of the language. Three kinds of interfaces are developed to annotate instances, evaluate annotation quality and manage the annotation task for annotators, reviewers and managers, respectively. FITAnnotator also gives intelligent annotations by introducing task-specific assistant to support and guide the annotators based on active learning and incremental learning strategies. This assistant is able to effectively update from the annotator feedbacks and easily handle the incremental labeling scenarios.¹

1 Introduction

Manually-labeled gold standard annotations are the first prerequisite for the training and evaluation of modern Natural Language Processing (NLP) methods. With the development of deep learning, neural networks have achieved state-of-the-art performance in a variety of NLP fields. These impressive achievements rely on large-scale training data for supervised training. However, building annotation requires a significant amount of human effort and incurs high costs, and can place heavy demands on human annotators for maintaining annotation quality and consistency.

To improve annotation productivity and reduce the financial cost of annotation, many text annotation softwares are developed by constraining user actions and providing an effective interface. In the early days, platforms for linguistic annotations such as O’Donnell (2008), BART (Stenetorp et al.,

2012), WebAnno-13 (Yimam et al., 2013) mainly focused on providing a visual interface for user labeling process, making annotation accessible to non-expert users. Recently, integrating active learning into annotation systems for providing suggestions to user has become mainstream (TextPro (Pianta et al., 2008), WebAnno-14 (Yimam et al., 2014), Active DOP (van Cranenburgh, 2018), INCEpTION (Klie et al., 2018), etc), but most of these works focus on English text and rarely consider the multi-lingual setting, which is necessary due to the growing demand for annotation in other languages. In addition to the interface and efficiency, incremental annotation is also necessary in real-world scenarios since the pre-defined annotation standards and rules cannot handle rapidly emerging novel classes in the real world, while being less addressed in existing annotation tools.

To address the challenges above, we propose FITAnnotator, a generic web-based tool for text annotation, which fulfills the following requirements:

- Extremely flexible and configurable: our system architecture is fully modular, even the user interface is a replaceable module. Which means it is model-agnostic and supports annotation on a variety of linguistic tasks, including tagging, classification, parsing, etc.
- Active learning: learning from small amounts of data, and selecting by itself what data it would like the user to label from an unlabeled dataset. Annotators label these selected instances and add them to the training set. A new model is automatically trained on the updated training set. This process repeats and results in dramatic reductions in the amount of labeling required to train the NLP model.
- Expansible data provider: the previous annotation tools are compatible with the static corpus for annotation, which is not convenient

*Corresponding author

¹A video demonstration of FITAnnotator is available at <https://vimeo.com/499446008>

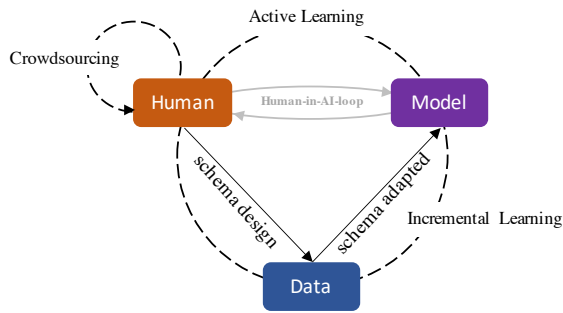


Figure 1: The interaction of the three major elements of the intelligent annotation system

for annotating from sketch and expansion. FITAnnotator sets up an independent data loader and data provider, which can continuously import data to the corpus in bulk. The flexible data provider also brings new problems, such as dynamic labeling schema, which should be solved by incremental learning.

- **Incremental learning:** creating a prototype for each category and enabling the prototypes of the novel categories far from the prototypes of the original categories while maintaining features to cluster near the corresponding category prototypes, which makes the tool suitable for annotating with new classes added incrementally.
- **Collaboration & crowdsourcing:** the system is designed for the multi-user scenario, where multiple annotators can work collaboratively at the same time. When multiple users cooperate in annotation, the dismountable crowdsourcing algorithm interface can be used to allocate overlapping data in apiece task packages, for evaluating the annotation quality of each user. Also, the system provides a manual review interface, which can perform sampling inspection and evaluation on various users' annotation.

Figure 1 reflects our design philosophy and comprehension of the interaction between the three major elements in our annotation system.

2 Related Works

In recent years, the NLP community has developed several annotation tools (Neves and Ševa, 2019). Yedda (Yang et al., 2018b) provides an easy-to-use and lightweight GUI software for collaborative text

annotation, and provides certain administrator analysis for evaluating multi-annotators. FLAT² introduces generalised paradigm and well-defined annotation format defined in folia (van Gompel, 2012), and provides web-based annotation interface. Doccano (Nakayama et al., 2018) is an open-source, web-based text annotation tool that provides collaboration, intelligent recommendation functions, and includes a user-friendly annotation interface. INCEpTION (Klie et al., 2018) is a comprehensive text annotation system, which is also web-based and open-source, integrates active learning algorithms and provides various interfaces for different annotation tasks, and it is developing for more tasks (de Castilho et al., 2018), more convenient (Bouloso et al., 2018) and low-resource scenarios (Klie et al., 2020).

In addition, commercial annotation tools such as prodigy³, tagtog⁴, LightTag⁵ also provide powerful active learning support, team-collaboration functions, efficient user interfaces, and provide more related commercial solutions, which have gained appreciable business achievement.

All of these intelligent text annotation tools have several common features: supporting active learning and a rich variety of tasks. And commercial annotation tools pay more attention to user experience and collaboration.

3 Architecture

The architecture of FITAnnotator is influenced by the ideas of functional programming and, in particular, by the desire to combine functional with object-oriented programming. The adherence to the programming principles such as immutability and modularity, FITAnnotator is developed by hybrid programming language Python. An overview of our system is shown in Figure 2, which has four main modules:

1. `core` module controls all data flow and provides the gateway for other modules. Tasks and projects are stored in the database of this module, and there are some fields to specify the URI of each related module. The system is based on these URIs to transfer and process data between modules. This module also

²<http://github.com/proycon/flat>

³<https://prodi.gy/>

⁴<http://www.tagtog.net>

⁵<https://www.lighttag.io/>

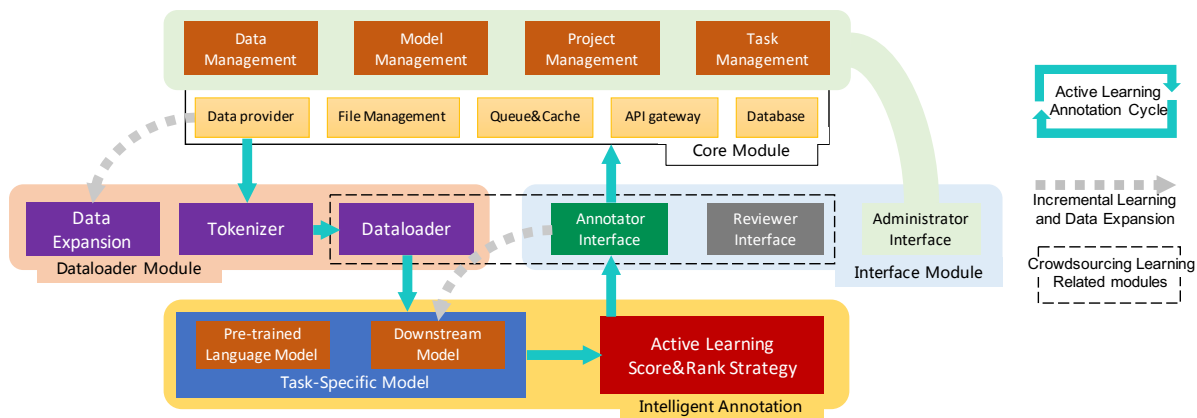


Figure 2: The overall architecture of the system

provides an administrator control panel for managing the system and database.

2. `data-loader` module contains fundamental tokenizer and data-loader of specific machine learning model. By deploy multifarious data-loader module with different tokenizers, we can adapt this system to different languages and tasks. In addition, we also provide data expansion function in this module. Expanded data would be cleaned in this module and passed to `core` module.
3. `intelligent annotation` module acts as the assistant which provides a pre-built machine learning model according to the type of tasks. This model could be simple as FastText (Joulin et al., 2017) or complex as BERT (Devlin et al., 2019). With such a model, we can obtain automatic labeling results for unlabeled data, and calculate their ranking scores according to the active learning strategy. By reordering the unlabeled data before pushing them to annotators, the annotation speed could be accelerated. Besides, incremental learning is also implemented in this module. We describe the details of this module in Section 4.
4. `interface` module contains three separate web interfaces: annotator, reviewer and administrator. The annotator interface presents the ranked unlabeled instances based on the recommendation score provided by the active learning module. Upon annotating a new sentence, the annotator is presented with the most probable labels recommended by the active learning model (see Figure 4). When the anno-

tators make a decision for confirming model recommendation or altering the labels, the operations will be fed back to the backend system and update the parameters of the active learning model. In the reviewer interface, the users monitor the progress of the annotation and see statistics such as the number of annotated instances, and the remaining unlabeled data. The reviewers can also review these already annotated instances and introduce corrections if necessary. In the administrator interface (shown in Figure 3), the project manager defines the annotation standards and sets all parameters for the annotation process, including the configures of active learning models, the management of annotators and reviewers, the assignment of tasks and so on.

The system is written with a modular design intended to be easily modifiable. Modules and interfaces (except `core` module and `administrator interface`) can be replaced easily for specific requirements. The flexibility it easy to adapt to multiple tasks and languages. FITAnnotator has three built-in annotation templates now: text classification, sequence tagging and semantic structure annotation, which cover most common NLP tasks, including sentence classification, sentence pair matching, named entity recognition and semantic role annotation. Users can also migrate to other tasks through simple modification of the framework.

4 Intelligent Annotation

Creating high-quality annotated corpora is a laborious process and requires experts who are highly familiar with the annotation schemes and stan-

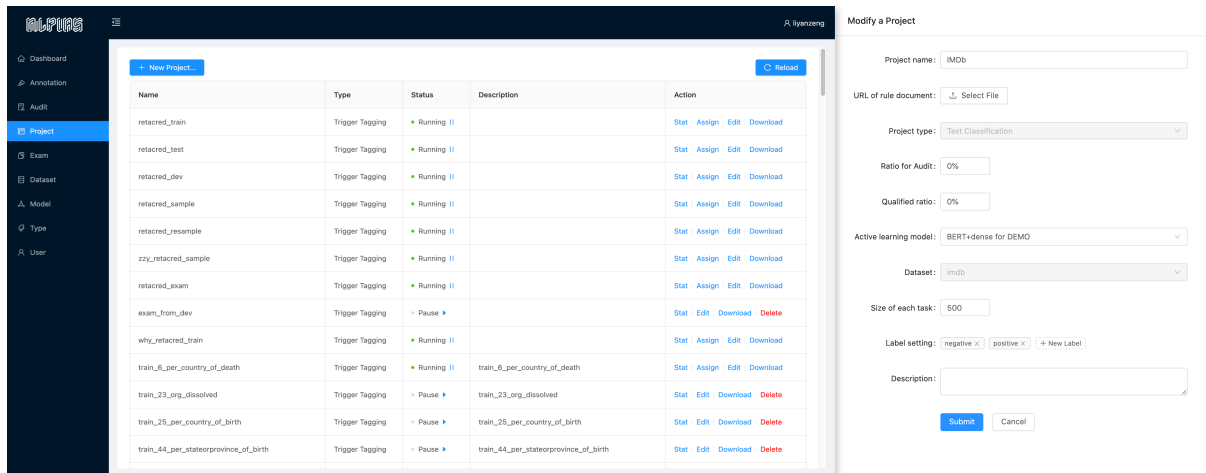


Figure 3: Screenshot of administrator interface

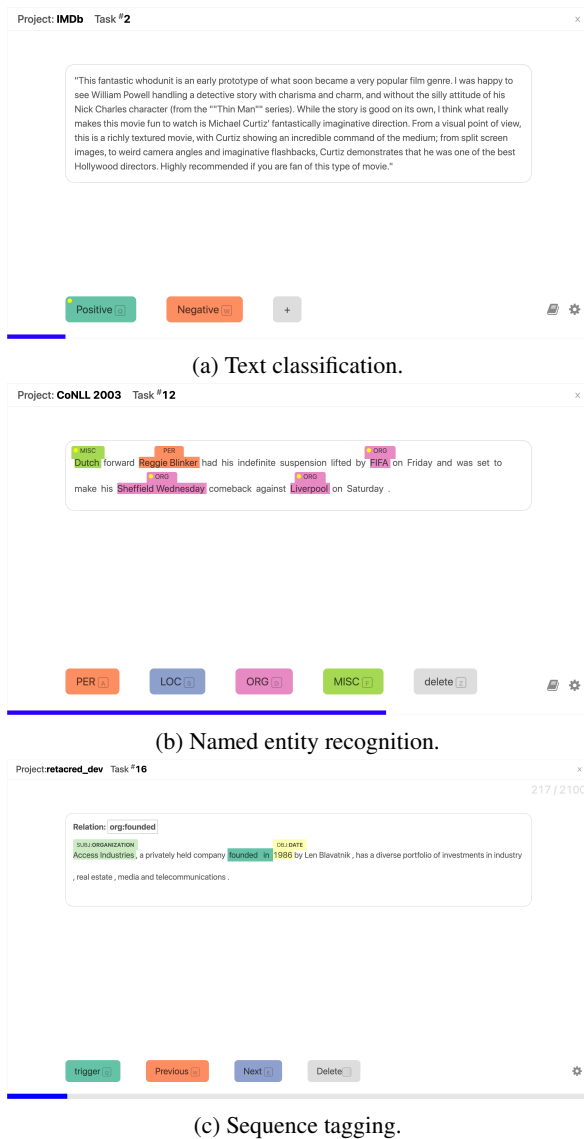


Figure 4: Screenshots of annotator interface for different tasks.

dards. To accelerate the annotation process, we introduce the intelligent assistant that incorporates task-specific neural networks which actively assist and guide annotators. The cores of intelligent annotation are two adaptive learning mechanisms: active learning and incremental learning.

4.1 Active Learning

A framework where a model learns from small amounts of data, and optimizes the selection of the most informative or diverse sample to annotate in order to maximize training utility value, is referred to as active learning (Gal et al., 2017; Schröder and Niekler, 2020). In particular, we employ a fused active learning method as a default strategy for evaluating, re-ranking and re-sampling data, which considers uncertainty and diversity at the same time (Zhou et al., 2017; Lutnick et al., 2019). Using such a strategy, the most difficult and diverse instances will be annotated first, which are more valuable for model learning with respect to the rest of the corpus. After the instances have been selected by active learning, the system displays them in the annotator interface with the highlighted suggestion labels. The annotator can then accept or modify the suggestion. The choices are stored and passed to the active learning module as new training data to update the parameters.

For analyzing the effectiveness of active learning strategies in FITAnnotator, we conduct a simple but representative comparative experiment based on the IMDb movie reviews sentiment classification task (Maas et al., 2011). In this experiment, we respectively explore the effectiveness of the uncertainty sampling and the diversity sampling in active

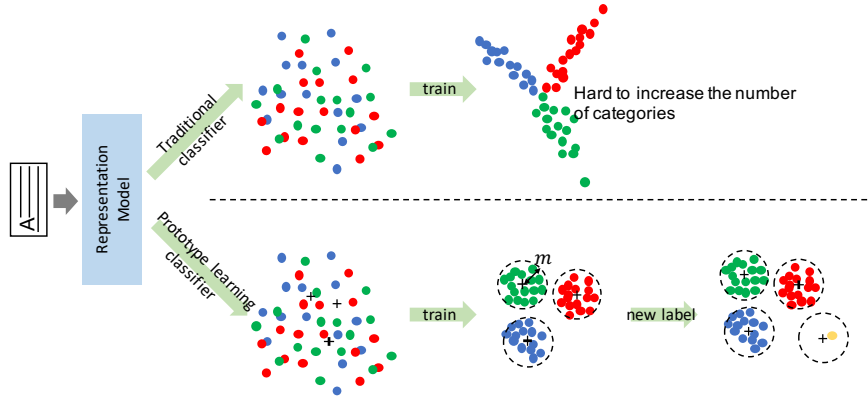


Figure 5: 2-dim sketch of prototype-based incremental learning

learning (Fu et al., 2013), and employ a random sampling strategy as the baseline method. Two kinds of popular text classification models (FastText (Joulin et al., 2017) and BERT (Devlin et al., 2019)) are respectively implemented as the backbone of active learning. We use *accuracy+* as the indicator to measure the performance (Lu et al., 2019):

$$accuracy+ = \frac{TP^H + TN^H + TP^M + TN^M}{N}$$

where N is the size of dataset, H and M represent the human-annotated labels and the model-predicted labels respectively. The evaluation is continuously carried out with the annotation process of the IMDB training set. Every 100 new annotation samples are generated, the performance of the backbone is evaluated on the standard test set. The results are shown in Figure 6. Apparently, the BERT-based active learning method outperforms the FastText-based method. In terms of training convergence speed, the sampling strategy based on the uncertainty criterion is similar to the diversity criterion, but both of them are obviously faster than the random sampling baseline. After plenty of samples are labeled, the accuracy of those sampling methods tends to be approximate. This observation demonstrates that our system is able to accelerate the training process of the models by introducing active learning algorithms, so as to provide users with label recommendations more quickly and accurately.

4.2 Incremental Learning

Existing annotation tools focus on labeling instances based on a fixed annotation scheme. However, the pre-defined standards may not cover all the cases met in the annotation process, especially

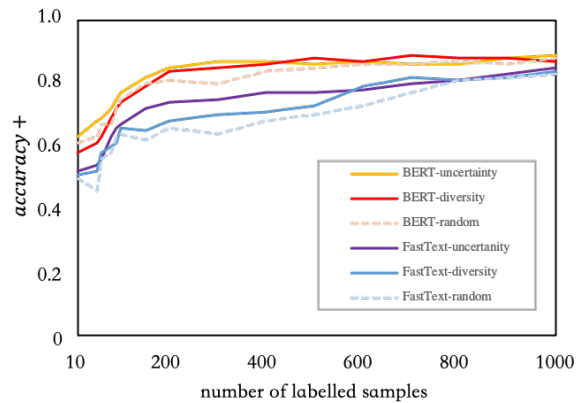


Figure 6: Results of different active learning strategies and models over imdb dataset. Curves start from 10 at along x-axis.

for the classification task with constantly updated source data. Take the case of aspect category classification (ACC). In E-commerce platforms, online reviews are valuable resources for providers to get feedback for their services. ACC aims to identify all the aspects discussed in a given review. Yet in the real world, new reviews and products are rapidly emerging, and it is impossible to annotate reviews with a pre-defined set of aspect categories once to cover all aspects (Toh and Su, 2015; Wu et al., 2018).

Considering the enormous cost of re-labeling the entire corpus, in an ideal annotation system, the new classes should be integrated into the existing labeled instances, sharing the previously learned parameters of active learning. To this end, we introduce an incremental learning mechanism into our annotation system. As Figure 5 shown, by creating a prototype for each category, the classification problem is converted into a problem of matching the samples to the prototypes (Yang et al.,

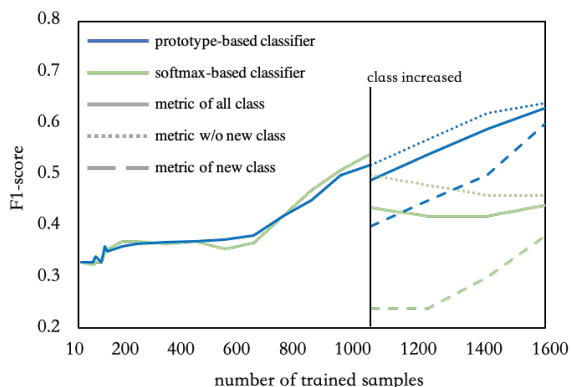


Figure 7: Results of different classifier (softmax-based and prototype-based) in class-incremental scenario.

2018a). During the training process, the loss function is designed to minimize the distance between the sample and the prototype (m in Figure 5 is the minimal margin between prototypes) and maximize the distance between prototypes. Thus the space of representation is sparse and clear outside of prototype clusters, a new prototype of the category can be added easily (Rebuffi et al., 2017).

To verify the effectiveness of FITAnnotator combined with incremental learning, we conduct experiments on the AG News dataset⁶, which is collected from the news corpus with four classes. In order to simulate the real-world scenario, we first use samples belonging to three of the four categories for annotation. After labeling 1000 samples, we import the data of the fourth category, and use the class-incremental function provided by FITAnnotator to change the annotation schema. For evaluation, we construct a word-level LSTM + CNN representation model with glove word embedding (Pennington et al., 2014) as the encoder, and compare our prototype-based method with the classic softmax-based classifier. The micro-F1 score is chosen as the evaluation metric.

Figure 7 illustrates the experimental results. In the ordinary text classification task, the performance of the softmax-based classifier and the prototype-based classifier is relatively approximate. After introducing the fourth class (new class), the performance of the softmax-based classifier occurs a catastrophic recession. On the contrary, the prototype-based method shows impressive results in the class-incremental scenario, and the negative effect of the newly introduced class is negligible.

⁶http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

5 Conclusion

In this paper, we present FITAnnotator, a web-based system for interactive NLP annotation. In order to reduce the workload of annotators, we integrate an active learning strategy in our system recommendation part, and introduce an incremental learning strategy to facilitate the rapid annotation of incessantly emerging novel categories. It supports a range of annotation types, and analyzing, assessing, and managing the annotations. In future work, FITAnnotator will integrate more advanced incremental learning and active learning algorithms, and be enhanced to develop more task templates.

Acknowledgement

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02040400.

References

- Beto Boullosa, Richard Eckart de Castilho, Naveen Kumar, Jan-Christoph Klie, and Iryna Gurevych. 2018. [Integrating knowledge-supported search into the inception annotation platform](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, volume Demo Papers, pages 127–132.
- Richard Eckart de Castilho, Jan-Christoph Klie, Naveen Kumar, Beto Boullosa, and Iryna Gurevych. 2018. [Linking text and knowledge using the inception annotation platform](#). In *Proceedings of the 14th eScience IEEE International Conference*, pages 327–328.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.
- Yifan Fu, Xingquan Zhu, and Bin Li. 2013. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431.

- Jan-Christoph Klie, Michael Bugert, Beto Boulosa, Richard Eckart de Castilho, and Iryna Gurevych. 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.
- Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. [From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993, Online. Association for Computational Linguistics.
- Jinghui Lu, Maeve Henchion, and Brian Mac Namee. 2019. Investigating the effectiveness of representations based on word-embeddings in active learning for labelling text datasets. *arXiv*, pages arXiv–1910.
- Brendon Lutnick, Brandon Ginley, Darshana Govind, Sean D. McGarry, Peter S. LaViolette, Rabi Yacoub, Sanjay Jain, John E. Tomaszewski, Kuang-Yu Jen, and Pinaki Sarder. 2019. An integrated iterative annotation technique for easing neural network training in medical image analysis. *Nature Machine Intelligence*, 1(2):112–119.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. [doccano: Text annotation tool for human](#). Software available from <https://github.com/doccano/doccano>.
- Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*.
- Mick O’Donnell. 2008. Demonstration of the uam corpustool for text and image annotation. In *Proceedings of the ACL-08: HLT Demo Session*, pages 13–16.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The textpro tool suite. In *LREC*.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. [icarl: Incremental classifier and representation learning](#). In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.
- Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*.
- 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.
- Zhiqiang Toh and Jian Su. 2015. Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 496–501.
- Andreas van Cranenburgh. 2018. Active dop: A constituency treebank annotation tool with online learning. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, page 38.
- Maarten van Gompel. 2012. Folia: Format for linguistic annotation. *CLIN22, Tilburg*.
- Chuhan Wu, Fangzhao Wu, Sixing Wu, Zhigang Yuan, and Yongfeng Huang. 2018. A hybrid unsupervised method for aspect term and opinion target extraction. *Knowledge-Based Systems*, 148:66–73.
- Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2018a. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3482.
- Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018b. [Yedda: A lightweight collaborative text span annotation tool](#).
- 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.
- Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6.
- Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. 2017. Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017, pages 4761–4772.