TARGER: Neural Argument Mining at Your Fingertips

Artem Chernodub[1,2], **Oleksiy Oliynyk**[3], **Philipp Heidenreich**[3], **Alexander Bondarenko**[4], **Matthias Hagen**[4], **Chris Biemann**[3], **and Alexander Panchenko**[3]

[1]Grammarly
[2]Faculty of Applied Sciences, Ukrainian Catholic University, Ukraine
[3]Language Technology Group, Department of Informatics, University of Hamburg, Germany
[4]Big Data Analytics Group, Martin-Luther University of Halle-Wittenberg, Germany

Abstract

We present a neural *argument tagger* coming with a web interface and API for argument mining. The tool can tag arguments in input free texts and retrieve arguments from a pre-tagged web-scale corpus. The web interface and the accompanying API provide models pre-trained on various datasets enabling the use of neural argument mining without any reproducibility effort at the user side. The code is open source to ensure portability to other domains and use cases.

1 Introduction

Argumentation is a multi-disciplinary field that extends from philosophy and psychology to linguistics as well as to artificial intelligence. Recent developments in argument mining apply natural language processing methods to argumentation (Palau and Moens, 2011; Lippi and Torroni, 2016a) and are mostly focused on training classifiers on annotated text fragments to identify argumentative text units, such as claims and premises (Biran and Rambow, 2011; Habernal et al., 2014; Rinott et al., 2015). More specifically, current approaches mainly focus on three tasks: (1) detection of sentences containing argumentative units, (2) detection of the argumentative units' boundaries inside sentences, and (3) identifying relations between argumentative units.

Despite vital research in argument mining, there is a lack of freely available tools that enable users, especially non-experts, to make use of the field's recent advances. In this paper, we close this gap by introducing TARGER: a system with a user-friendly web interface[1] that can extract argumentative units in user-specified free texts in real-time based on recent neural models trained on common argument mining corpora with a highly configurable and efficient neural sequence tagger. The native PyTorch implementation is also available as open source.[2] It has no external dependencies and thus is very easy to incorporate into any existing NLP pipeline. For cases where a user's free text input does not contain any arguments, TARGER's API and web interface also allow for very fast retrieval of arguments from an argument-tagged version of the web-scale Common Crawl-based DepCC (Panchenko et al., 2018).

2 Related Work

There are three publicly available systems offering some functionality similar to TARGER. ArgumenText (Stab et al., 2018) is an argument search engine that retrieves argumentative sentences from the Common Crawl and labels them as *pro* or *con* given a keyword-based user query. Similarly, args.me (Wachsmuth et al., 2017) retrieves *pro* and *con* arguments from 300,000 arguments crawled from debating portals. Finally, MARGOT (Lippi and Torroni, 2016b) provides argument tagging for free-text inputs. However, answer times are rather slow in the 5+ seconds range for single input sentences.

TARGER offers a real-time retrieval functionality similar to ArgumenText and fast real-time free-text argument tagging with the option of switching between different pre-trained state-of-the-art models (MARGOT offers only a single one).

3 Design of TARGER

The overall architecture of TARGER is shown in Figure 1. It contains components for data preprocessing and a front-end web interface serving pre-trained models and indexed data. The framework

[1] http://ltdemos.informatik.uni-hamburg.de/argsearch

[2] https://github.com/achernodub/targer

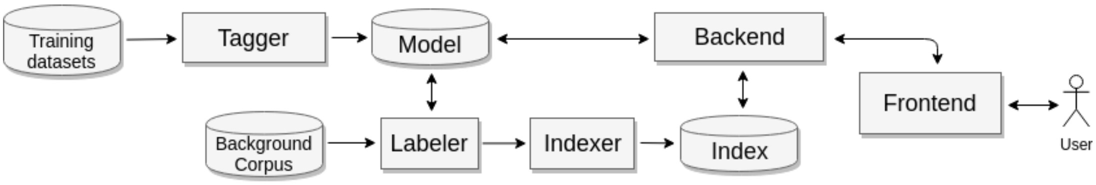


Figure 1: Architecture of TARGER, an open source system for tagging and retrieving arguments.

Table 1: Characteristics of the training datasets.

	Essays	WebD	IBM
Claims	22,443	3,670	8,073,589
Premises	67,157	20,906	35,349,501
Major Claims	10,966	-	-
Backing	-	10,775	-
Refutations	-	867	-
Rebuttals	-	2,247	-
None	47,619	46,352	3,710,839
Combined	**148,185**	**84,817**	**47,133,929**

is modular and flexible. Each component is independent of the others and uses HTTP requests for internal interaction. Due to the modularity, the different parts can be deployed on different servers, either in Docker containers or natively.

In the preprocessing phase, we trained a neural sequence tagger (component "Tagger") on different datasets (cf. Section 3.1) yielding a variety of argument mining models. For free-text tagging, the user can choose between them. In the final preprocessing step, the trained models were run on the web-scale background corpus and the argument units were stored in additional fields of a respective Elasticsearch index (components "Labeler" and "Indexer").

The frontend of TARGER is implemented as a Flask-based web app. Any user input is sent as an AJAX request to the app, where it is processed, interactions with other components are performed, and results are sent back to the user interface. The backend accepts unformatted free text to be tagged with argumentative units by one of the pre-trained models, while the index can be queried from the frontend with keyword queries to retrieve text passages where the query terms match argumentative units in the background corpus. In the following sections, we present various components and technical details of our framework.

3.1 Neural Sequence Tagger

We implemented a BiLSTM-CNN-CRF neural tagger (Ma and Hovy, 2016) for identifying argument components and classifying them as claims or premises. The BiLSTM-CNN-CRF method is popular for sequence tagging tasks such as NER and POS-tagging (Ma and Hovy, 2016; Lample et al., 2016) as well as for argument mining (Eger et al., 2017). It relies on pre-computed word embeddings, a single bidirectional-LSTM/GRU recurrent layer, convolutional character-level embeddings to capture out-of-vocabulary words, and a first-order Conditional Random Field (Lafferty et al., 2001) to capture dependencies between adjacent tags. The end-to-end learning architecture achieves state-of-the-art or near state-of-the-art performance for sequence tagging tasks like NERor POS-tagging.

The argument tagging models were trained on the persuassive essays (Essays) (Eger et al., 2017), web discourse (WebD) (Habernal and Gurevych, 2017), and IBM Debater (IBM) (Levy et al., 2018) datasets; characteristics given in Table 1.

Our Python 3.6 / PyTorch 1.0 implementation does not contain any third-party dependencies, has native vectorized code for high-performance training and evaluation, and supports several input data formats as well as evaluation score functions. The models currently deployed to the TARGER API and web interface use GloVe (Pennington et al., 2014), fastText (Mikolov et al., 2018), or dependency-based embeddings (Levy and Goldberg, 2014) (overview in Table 2). During the training, we performed hyperparameter tuning for every model with the following variations: optimizer [SGD, Adam], learning rate [0.001, 0.05, 0.01], dropout [0.1, 0.5], number of hidden units in recurrent layer [100, 150, 200, 250].

3.2 Retrieval Functionality

We used an Apache Spark cluster to run the trained models on the DepCC background collection (component "Labeler" in Figure 1) and extended our Elasticsearch index of the DepCC by additional fields containing the argument unit spans and labels of the different models.

Table 2: Models currently deployed in TARGER.

Data	Embeddings	Tagger
Essays	fastText	(Reimers and Gurevych, 2017)
Essays	Dependency	(Reimers and Gurevych, 2017)
Essays	GloVe	Ours
WebD	fastText	(Reimers and Gurevych, 2017)
WebD	Dependency	(Reimers and Gurevych, 2017)
WebD	GloVe	Ours
IBM	fastText	(Reimers and Gurevych, 2017)
IBM	GloVe	Ours

We use a linguistically pre-processed Common Crawl corpus (Panchenko et al., 2018) as a background document collection for argument retrieval. It contains 365 million documents and 14.3 billion sentences in English from the Web.

3.3 TARGER API

To keep the TARGER framework modular and scalable while still allowing access to the models from external clients, we provide a restful API (component "Backend" in Figure 1). Each trained model is associated with a separate API endpoint accepting raw text as input. The output is provided as a list of word-level tokens with IOB-formatted labels for argument units (premises and claims) and the tagger's confidence scores for each label.

3.4 TARGER Web Interface

The web interface of the tool offers two functionalities available as tabs in a web browser: *Analyze Text* and *Search Arguments*. On the analysis tab (cf. Figure 2), the user can choose one of the deployed models to identify arguments in a user-provided free text. The result is shown with colored labels for different types of argumentative units (premises and claims) as well as detected named entities (nested tags for entities in argument units are supported). Once a result is shown, it is possible to customize the display by enabling/disabling different labels without performing additional tagging runs.

On the retrieval tab (cf. Figure 3), the user can enter a keyword query and choose whether the keywords should be matched in claims, premises, etc. Every retrieved result then comes as a text fragment colorized with argument and entity information just as on the analysis tab. To indicate provenance, the URL of the original document is also provided.

4 Evaluation

We evaluate the pre-trained models for argument tagging on the common argument mining benchmark datasets of Persuasive Essays (Eger et al., 2017), Web Discourse (Habernal and Gurevych, 2017), and IBM (Levy et al., 2018) using the 70/10/20 train/dev/test split of these datasets.

4.1 Experimental Results

We used the evaluation setup suggested in the respective original publications. On the Persuasive Essays dataset (Paragraph level), we achieved a span-level micro-F1 of 64.54 for extracted argument components matching the best performance of 64.74±1.97 reported in the (Eger et al., 2017).

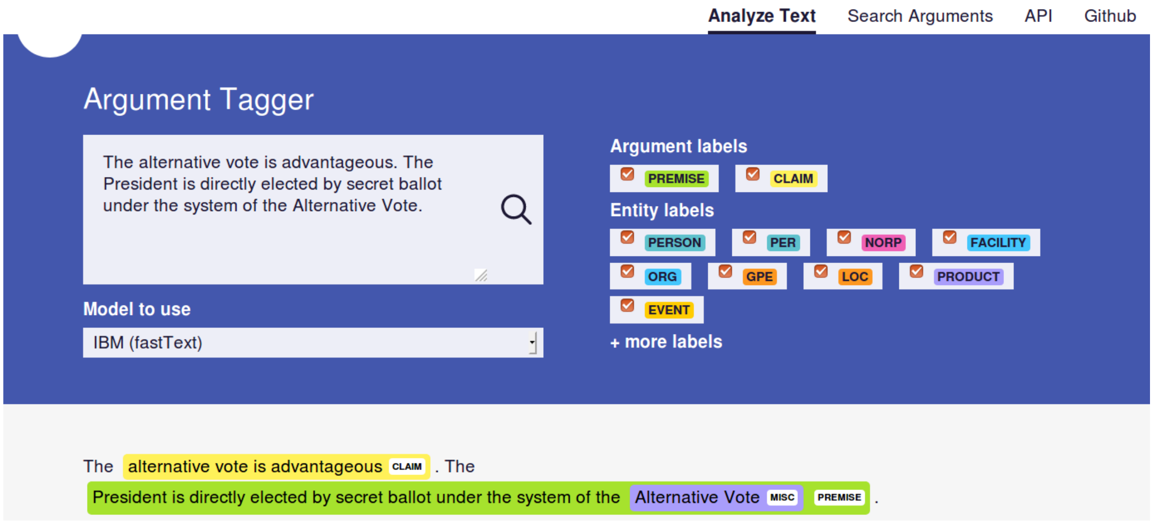
For adapting the IBM Debater dataset, we transformed original entries to the token-level claims and premises. We got a micro-F1 score of 99.87 which we explain by the specific type of the data and its structure. This dataset does not contain much text labeled as non-argumentational that complicates transfer to other domains.

On the Web Discourse dataset, we reproduced 10-fold cross-validation from (Habernal and Gurevych, 2017). Our neural tagger only used word embeddings as features opposed to lexical, structural, morphological and other handcrafted features used in the original publication. As it was shown for cross-domain experimental scenarios (Habernal and Gurevych, 2017), handcrafted features show a strong tendency to overfit on the topics of the training texts, so "word embeddings only" is a more robust feature choice in domain-agnostic general-purpose systems as required for web or free-text data. In this setting, we slightly improved the originally reported token-level macro-F1 of 22.9 (Habernal and Gurevych, 2017) by achieving a macro-F1 of 24.2.

4.2 TARGER @ TREC Common Core Track

As a proof-of-concept, we used the TARGER API in a TREC 2018 Common Core track submission (Bondarenko et al., 2018). TARGER served as a subroutine in a pipeline of *axiomatic re-ranking* (Hagen et al., 2016) of BM25F retrieval results with respect to their argumentativeness (presence/absence of arguments). For articles from the underlying Washington Post collection[3], TARGER identified the argumentative units. We achieved re-ranking results in the Core track demonstrate

[3] https://trec.nist.gov/data/wapost/

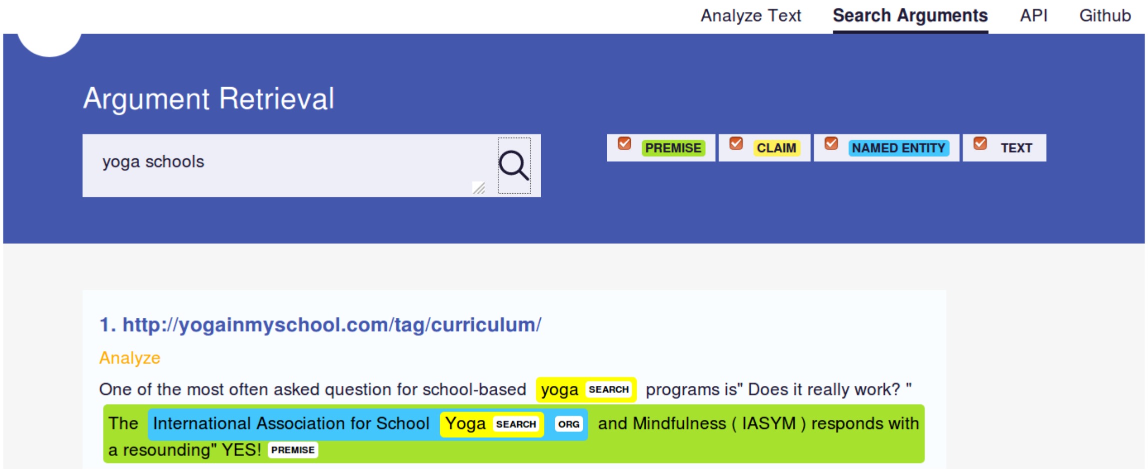


The alternative vote is advantageous. The President is directly elected by secret ballot under the system of the Alternative Vote.

Model to use: IBM (fastText)

The alternative vote is advantageous CLAIM . The President is directly elected by secret ballot under the system of the Alternative Vote MISC PREMISE .

Figure 2: **Analyze Text**: input field, drop-down model selection, colorized labels, and tagged result.



yoga schools

1. http://yogainmyschool.com/tag/curriculum/

Analyze

One of the most often asked question for school-based yoga SEARCH programs is" Does it really work? " The International Association for School Yoga SEARCH ORG and Mindfulness (IASYM) responds with a resounding" YES! PREMISE

Figure 3: **Search Arguments**: query box, field selectors, and result with link to original document.

substantial improvements of 10+% increase in retrieval quality for four out of 25 TREC topics that we manually labeled as potentially "argumentative" (i.e., where relevant results probably should contain arguments).

In similar and different scenarios TARGER could serve as a valuable complementary subroutine such that integrating new neural argument mining models will be an ongoing effort at our side to constantly support TARGER's users.

5 Conclusion

We have presented a system for tagging arguments in free text and for retrieving arguments from a web-scale corpus. With the available RESTful API and the web interface, users can simply utilize and integrate different state-of-the-art argument mining approaches in their pipelines or perform manual analysis of texts. Thus, we make the recent argument mining technologies more accessible and usable to the general public and developers. Our framework is available under a permissive open source license. Besides, it allows to train and deploy new taggers so users can take advantage of the argument mining state-of-the-art. Our future work is to integrate models based on ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018).

Finally, by looking at results of tagging of various state-of-the-art taggers using our system, it becomes clear than despite recent advanced in argument mining there is still a long way to go from the domain adaptation side. While in-domain classification results are relevant, random text snippets of different styles and genres taken from the Web, are annotated surprisingly inconsistently. Our finding urges more research on domain adaptation and transfer learning (Ruder, 2019) for argument mining to address this issue.

References

Or Biran and Owen Rambow. 2011. Identifying Justifications in Written Dialogs. In *Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011), Palo Alto, CA, USA, September 18-21, 2011*, pages 162–168.

Alexander Bondarenko, Michael Völske, Alexander Panchenko, Chris Biemann, Benno Stein, and Matthias Hagen. 2018. Webis at TREC 2018: Common Core Track. In *27th International Text Retrieval Conference (TREC 2018)*, NIST Special Publication. National Institute of Standards and Technology (NIST).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 11–22.

Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation Mining on the Web from Information Seeking Perspective. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014.*

Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics*, 43(1):125–179.

Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. 2016. Axiomatic Result Re-Ranking. In *25th ACM International Conference on Information and Knowledge Management (CIKM 2016)*, pages 721–730. ACM.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308.

Ran Levy, Ben Bogin, Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. Towards an Argumentative Content Search Engine Using Weak Supervision. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2066–2081.

Marco Lippi and Paolo Torroni. 2016a. Argumentation Mining: State of the Art and Emerging Trends. *ACM Trans. Internet Techn.*, 16(2):10:1–10:25.

Marco Lippi and Paolo Torroni. 2016b. MARGOT: A Web Server for Argumentation Mining. *Expert Syst. Appl.*, 65:292–303.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-End Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers.*

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Raquel Mochales Palau and Marie-Francine Moens. 2011. Argumentation Mining. *Artif. Intell. Law*, 19(1):1–22.

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. 2018. Building a Web-Scale Dependency-Parsed Corpus from CommonCrawl. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 338–348.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 440–450.

Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching for Arguments in Heterogeneous Sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HTL 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Demonstrations*, pages 21–25.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an Argument Search Engine for the Web. In *Proceedings of the 4th Workshop on Argument Mining, ArgMining@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 49–59.