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[5,3]

[1]Grammarly
[2]Faculty of Applied Sciences, Ukrainian Catholic University, Lviv, Ukraine
[3]Language Technology Group, Universität Hamburg, Hamburg, Germany
[4]Big Data Analytics Group, Martin-Luther Universität Halle-Wittenberg, Halle, Germany
[5]Skolkovo Institute of Science and Technology, Moscow, Russia

Abstract

We present TARGER, an open source neural argument mining framework for tagging arguments in free input texts and for keyword-based retrieval of arguments from an argument-tagged web-scale corpus. The currently available models are pre-trained on three recent argument mining datasets and enable the use of neural argument mining without any reproducibility effort on the user's side. The open source code ensures portability to other domains and use cases, such as an application to search engine ranking that we also describe shortly.

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 (NLP) 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 free input texts in real-time using models trained on recent argument mining corpora with a highly configurable and efficient neural sequence tagger. TARGER's web interface and API also allow for very fast keyword-based argument retrieval from a pre-tagged version of the Common Crawl-based DepCC (Panchenko et al., 2018).

The native PyTorch implementation underlying TARGER has no external depencies and is available as open source software:[2] it can easily be incorporated into any existing NLP pipeline.

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 of MARGOT are rather slow for single input sentences (>5 seconds) and the F1 scores of 17.5 for claim detection and 16.7 for evidence detection are slightly worse compared to our approach (see Section 4.1).

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 Architecture of TARGER

The independent components of the modular and flexible TARGER framework are shown in Figure 1. In an offline training step, a neural

[1]ltdemos.informatik.uni-hamburg.de/targer

[2]github.com/achernodub/targer

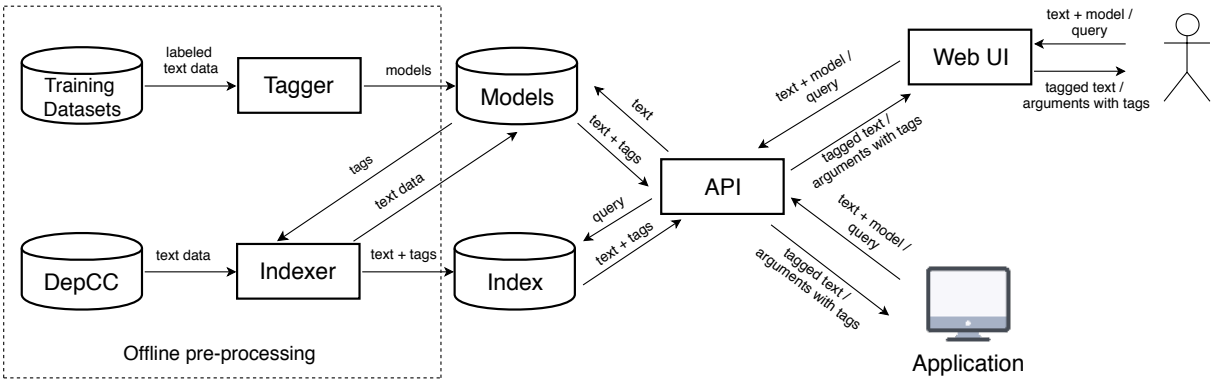

Figure 1: Modular architecture of TARGER. The central API is accessed through the Web UI or directly from any application; it routes requests either to the tagging models or to the retrieval component. TARGER's components communicate via HTTP requests and can be deployed on different servers—in Docker containers or natively.

BiLSTM-CNN-CRF sequence tagger is trained on different datasets yielding a variety of argument mining models (details in Section 3.1). As part of the preprocessing, the trained models are run on the 14 billion sentences of the DepCC corpus to tag and store argument unit information as additional fields in an Elasticsearch BM25F-index of the DepCC (details in Section 3.2).

The online usage is handled via a Flask-based web app whose API accepts AJAX requests from the Web UI component or via API calls (details in Sections 3.3 and 3.4). The web interface is based on the named entity visualiser displaCy ENT.[3] The API routes free text inputs to the respective selected model to be tagged with argument information or it routes keyword-based queries to the index to retrieve sentences in which the query terms match argument units.

3.1 Neural Sequence Tagger

We implement a BiLSTM-CNN-CRF neural tagger (Ma and Hovy, 2016) for identifying argumentative units and for classifying them as claims or premises. The BiLSTM-CNN-CRF method is a popular sequence tagging approach and achieves (near) state-of-the-art performance for tasks like named entity recognition and part-of-speech tagging (Ma and Hovy, 2016; Lample et al., 2016); it has also been used for argument mining before (Eger et al., 2017). The general method 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.

[3] github.com/explosion/displacy-ent

	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**

Table 1: Number of tokens per category in the training datasets. Note that the IBM data contains many duplicate claims; it was originally published as a dataset to identify relevant premises for 150 claims.

Besides the existing BiLSTM-CNN-CRF implementation of Reimers and Gurevych (2017), we also use an own Python 3.6 / PyTorch 1.0 implementation that does not contain any third-party dependencies, has native vectorized code for efficient training and evaluation, and supports several input data formats as well as evaluation functions.

The different argument tagging models currently usable through TARGER's API are trained on the persuasive essays (Essays) (Eger et al., 2017), the web discourse (WebD) (Habernal and Gurevych, 2017), and the IBM Debater (IBM) (Levy et al., 2018) datasets (characteristics in Table 1). The models use GloVe (Pennington et al., 2014), fastText (Mikolov et al., 2018), or dependency-based embeddings (Levy and Goldberg, 2014) (overview in Table 2).

For training, the following variations were used for hyperparameter tuning: optimizer [SGD, Adam], learning rate [0.001, 0.05, 0.01],

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

Table 2: Models currently deployed in TARGER.

dropout [0.1, 0.5], number of hidden units in recurrent layer [100, 150, 200, 250]. On all datasets, GloVe embeddings, Adam with learning rate of 0.001 and dropout rate of 0.5 performed best (hidden units: 200 on the persuasive essays, 250 on web discourse and IBM data).

3.2 Retrieval Functionality

Our background collection for the retrieval of argumentative sentences is formed by the DepCC corpus (Panchenko et al., 2018), a linguistically pre-processed subset of the Common Crawl containing 14.3 billion unique English sentences from 365 million web documents.

The trained WebD-GloVe model was run on all the sentences in the DepCC corpus since it performed best on the web data in a pilot experiment. The respective argumentative unit spans and labels were added as additional fields to an Elasticsearch BM25F-index of the DepCC.

3.3 TARGER API

To keep the TARGER framework modular and scalable while still allowing access to the models from external clients, online interaction is handled via a restful API. 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 UI

The web interface of TARGER offers two functionalities: *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 argumentative 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 it should be matched in claims, premises, etc. Every retrieved result is rendered as a text fragment colorized with argument and entity information just as on the analysis tab. To enable provenance, the URL of the source document is also provided.

Essays		**Web Discourse**	
Approach	F1	Approach	F1
STag_{BLCC}	64.74	SVM^{hmm}	22.90
TARGER (GloVe)	64.54	TARGER (GloVe)	24.20

Table 3: Comparison of TARGER's performance on the essays (Eger et al., 2017) and web discourse data (Habernal and Gurevych, 2017) to the best approaches from the original publications.

4 Evaluation

To demonstrate that our neural tagger is able to reproduce the originally published argument mining performances, we compare the best performing of our pre-trained models (parameter settings at the end of Section 3.1) to the best performances from the original dataset publications. We also report on a pilot study using TARGER as a subroutine in runs for the TREC 2018 Common Core track.

4.1 Experimental Results

Table 3 shows a comparison of TARGER's best performing models (parameter settings at the end of Section 3.1) on the Persuasive Essays and the Web Discourse datasets to the best performance in the original publications. We apply the experimental settings of the original papers: a fixed 70/20/10 train/dev/test split on the Essays data, and a 10-fold cross-validation for Web Discourse (in our case allocating 7 folds for training and 2 for development in each iteration).

On the Persuasive Essays dataset (paragraph level), the best TARGER model achieves a span-based micro-F1 of 64.54 for extracted argument components matching the best performance of 64.74±1.97 reported by Eger et al. (2017) for their STag_{BLCC} approach (BiLSTM-CRF-CNN approach (BLCC) similar to ours).

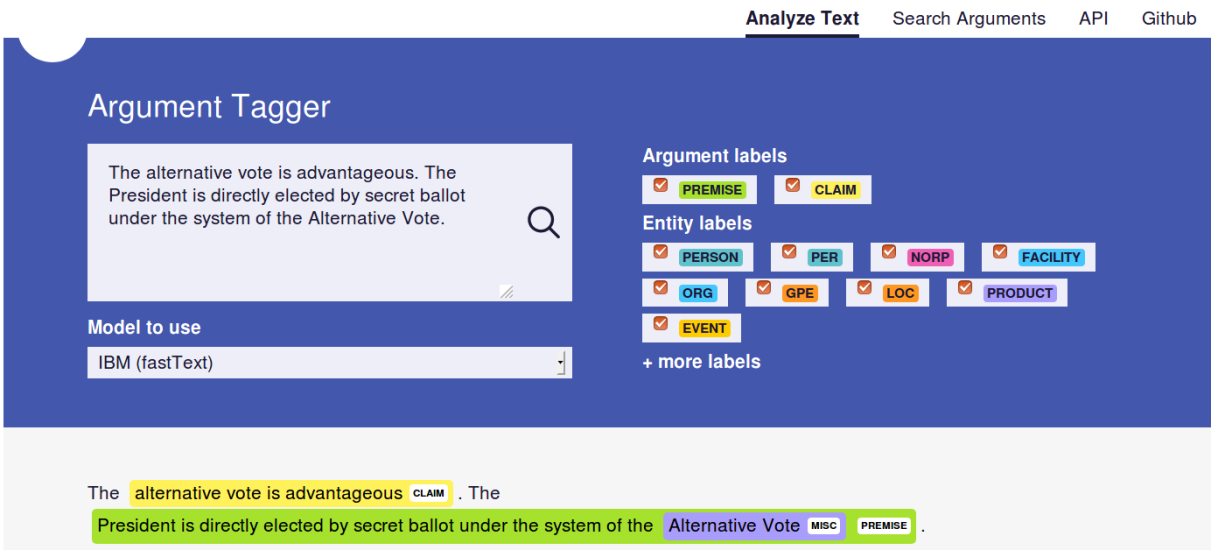

Figure 2: **Analyze Text**: input field, drop-down model selection, colorized labels, and tagged result.

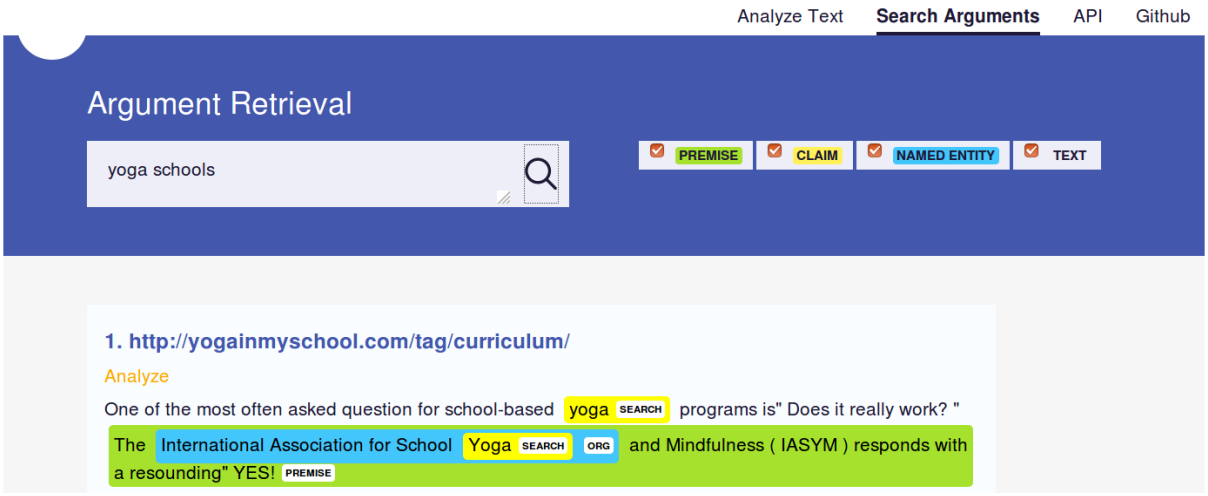

Figure 3: **Search Arguments**: query box, field selectors, and result with link to the original document.

On the Web Discourse dataset, TARGER's best model's token-based macro-F1 of 24.20 slightly improves upon the originally reported best macro-F1 of 22.90 (Habernal and Gurevych, 2017) achieved by a structural support vector machine model SVM^{hmm} for sequence labeling (Joachims et al., 2009). The SVM^{hmm} model uses lexical, structural, and other handcrafted feature types. In contrast, TARGER just uses word embeddings since especially for cross-domain scenarios, handcrafted features show a strong tendency to overfit on the topics of the training texts (Habernal and Gurevych, 2017). Thus, we chose "word embeddings only" as a more robust feature type for our domain-agnostic general-purpose argument mining system (free input text and web data).

We cannot compare TARGER's performance on the IBM dataset to originally published performances since the tasks are different. Instead of TARGER's identification of claims and premises, Levy et al. (2018) focus on the identification of relevant premises for a given claim (called "topic" in the original publication). Still, a large number of potential general domain premises for the overall 150 topics (i.e., claims) are contained in the dataset, such that we transformed the original entries to a token-level claim and premise annotation. This way, only some 2500 distinct tokens were labeled as not argumentative (e.g., punctuation) while the vast majority are tokens in claims and premises (but the only 150 different claims are heavily duplicated). Not surprisingly—given the class imbalance and duplication—, the resulting trained TARGER models "optimistically" iden-

Title / Query	BM25F	Axiomatic Re-Ranking
declining middle class in u.s.	0.91	0.98 (+0.07)
euro opposition	0.81	1.00 (+0.19)
airport security	0.52	0.72 (+0.20)
law enforcement, dogs	0.43	0.63 (+0.20)

Table 4: The TREC 2018 Common Core track topics with argument axiom re-ranked nDCG@10 improvements > 0.05 over a BM25F baseline.

tify some argumentative units in almost every input text. We still provide the models as a starting point with the intention to de-duplicate the data and to add more non-argumentative text passages for a more balanced / realistic training scenario.

4.2 TARGER @ TREC Common Core Track

As a proof of concept, we used TARGER's model pre-trained on essays with dependency-based embeddings in a TREC 2018 Common Core track submission (Bondarenko et al., 2018). The TARGER API served as a subroutine in a pipeline axiomatically re-ranking (Hagen et al., 2016) BM25F retrieval results with respect to their argumentativeness (presence/absence of arguments). For the Washington Post corpus used in the track, the dependency-based essays model best tagged argumentative units in a small pilot study.

Out of 25 topics manually labeled as argumentative from the 50 Common Core track topics, the TARGER-based argumentativeness re-ranking improved the retrieval quality by > 0.05 nDCG@10 for 4 topics (see Table 4). Argumentativeness-based re-ranking might thus be a viable way to integrate neural argument mining into the retrieval process—for instance, using TARGER.

5 Conclusion

We have presented TARGER: an open source 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, we make the recent argument mining technologies more accessible and usable to researchers and developers as well as the general public. The different argument mining models can easily be used to perform manual text analyses or can seamlessly be integrated into automatic NLP pipelines. New taggers can be deployed to TARGER at any time, so that users can have the state of the art in argument mining at their fingertips. For future work, we plan to integrate contextualized embeddings with ELMo- and BERT-based models (Peters et al., 2018; Devlin et al., 2018).

Finally, by looking at our experimental results as well as tagging examples for free input texts or the DepCC web data, we noticed that despite the recent advances in argument mining, there is still considerable headroom to improve in-domain, but especially out-of-domain argument tagging performance. Free input texts of different styles or genres taken from the web are tagged very inconsistently by the different models. More research on domain adaptation and transfer learning (Ruder, 2019) in the scenario of argument mining needs to address this issue.

Acknowledgments

This work was supported by the DAAD through the short-term research grant 57314022 and by the DFG through the project "ACQuA: Answering Comparative Questions with Arguments" (grants BI 1544/7-1 and HA 5851/2-1) as part of the priority program "RATIO: Robust Argumentation Machines" (SPP 1999). Finally, we are grateful to Ines Montani for developing displaCy ENT, an open source library our web interface is based on.

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)*, 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)*.

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 (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*, pages 26–39.

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.

Thorsten Joachims, Thomas Finley, and Chun-Nam J. Yu. 2009. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

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)*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 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 (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*, 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 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 (Volume 1: Long Papers)*, pages 1064–1074.

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*, pages 52–55.

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

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone P. 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*, pages 1816–1823.

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)*, pages 1532–1543.

Matthew 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, 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*, 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*, 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: 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*, pages 49–59.