Datenbank Spektrum (2020) 20:155–160
https://doi.org/10.1007/s13222-020-00346-8

SCHWERPUNKTBEITRAG

Answering Comparative Questions with Arguments

Alexander Bondarenko[1] · Alexander Panchenko[2] · Meriem Beloucif[3] · Chris Biemann[3] · Matthias Hagen[1]

Received: 14 February 2020 / Accepted: 30 May 2020 / Published online: 16 June 2020

Abstract
Question answering platforms such as Yahoo! Answers or Quora always contained questions that ask other humans for help when comparing two or more options. Since nowadays more and more people also "talk" to their devices, such comparative questions are also part of the query stream that major search engines receive. Interestingly, major search engines answer some comparative questions pretty well while for others, they just show the "standard" ten blue links. But a good response to a comparative question might be very different from these ten blue links—for example, a direct answer could show an aggregation of the pros and cons of the different options. This observation motivated our DFG-funded project "ACQuA: Answering Comparative Questions with Arguments" for which we describe the achieved results so far, and ongoing activities like the first shared task on argument retrieval.

Keywords Comparative Questions · Argumentation · Information Retrieval · Natural Language Processing

1 Introduction

The goal of the ACQuA project (funded within the DFG-SPP 1999 RATIO) is to develop algorithms and technology that help to understand and answer comparative information needs expressed as natural language questions by retrieving and combining facts, opinions, and arguments from knowledge graphs and web-scale text resources. To this end, the "Big Data Analytics" group from the MLU Halle[1] and the "Language Technology" group from the Universität Hamburg[2] collaborate with Alexander Panchenko's group from the Skolkovo Institute of Science and Technology[3] as an associated partner (before moving to Moscow, Alexander was a PostDoc in the ACQuA project).

The project is motivated by the fact that everyone faces a variety of choices on a daily basis (e.g., what programming language to use or whether to buy an electric car) and often can easily formulate a respective question containing the potential options and important aspects. However, current major web search engines do not answer many such comparative questions in another form than by repeating answers from question answering platforms to similar questions or showing ten blue links somewhat related to the question.

Instead, exploiting the web as a knowledge source, an answer to a comparative question should ideally directly combine the available facts, opinions, and arguments in a (short) natural language answer explaining under what circumstances which alternative should be chosen and why. This is the envisioned behavior of our comparative argumentation machine (CAM) for which we work on the following modules in the ACQuA project: (1) a user-friendly interface to submit a comparative question in natural language, (2) a question understanding component that identifies the compared objects and important comparison aspects, (3) a system that retrieves appropriate facts from

[1] https://halle.webis.de.

[2] https://www.inf.uni-hamburg.de/en/inst/ab/lt.

[3] https://sites.skoltech.ru/nlp.

✉ Alexander Bondarenko
alexander.bondarenko@informatik.uni-halle.de

Alexander Panchenko
A.Panchenko@skoltech.ru

Meriem Beloucif
beloucif@informatik.uni-hamburg.de

Chris Biemann
biemann@informatik.uni-hamburg.de

Matthias Hagen
matthias.hagen@informatik.uni-halle.de

[1] Martin-Luther-Universität Halle-Wittenberg, Halle, Germany

[2] Skolkovo Institute of Science and Technology,
Moscow, Russian Federation

[3] Universität Hamburg, Hamburg, Germany

a knowledge graph and relevant (possibly argumentative) documents from a web-scale text resource, and (4) a component that generates a (short) natural language answer from the different extracted facts and retrieved documents.

2 Project Results So Far

In this section, we give a brief overview of the results that we have achieved since we started working on the modules of the envisioned CAM system in late 2017.

2.1 Comparative Argumentative Machine (CAM)

We have developed a prototype of the CAM system [15] that can be accessed online.[4] The system takes as input two target objects and an optional list of comparison aspects (i.e., no natural language question, yet) and then retrieves sentences supporting either of the objects with respect to the given but also some further automatically identified comparison aspect(s) (e.g., "Python is better than PHP for web development."). The answer is then presented in form of the retrieved supporting sentences for the two objects and an overall "score" showing which object is favored in the retrieved sentences.

The CAM system has the following components.

(1) Sentence retrieval: the input query (objects and aspects) is run against an Elasticsearch index of the Common Crawl-based DepCC [13] (14.3 billion linguistically pre-processed English sentences).
(2) Sentence classification: a classifier [12] maps the retrieved sentences to one of four classes: the first object from the user input is better/equal/worse than the second one, or no comparison is found.
(3) Sentence ranking: the retrieved sentences are re-ordered by descending products of the classification confidence and the Elasticsearch retrieval scores.
(4) Aspect identification: up to ten additional aspects are automatically identified, even when no comparison aspects are provided by the user, by searching for (phrases with) comparative adjectives/adverbs and hand-crafted patterns like "because of higher ..." or "reason for this is ...".
(5) User interface: keyword boxes as input form and an answer presentation component (cf. Fig. 1).

We compared the CAM prototype to a "classical" keyword-based search system in a user study that asked participants to answer comparative questions. The results showed that the CAM users were 15% more accurate in finding correct answers about 20% faster (for more details, see our respective paper [15]).

In the current CAM prototype, the sentence classifier is pre-trained on sentences from only three domains: computer science, brands, and misc (books, sports, animals, etc.). Further diversifying the training domains is thus one idea to improve the prototype while another rather "obvious" important step is to allow for natural language questions as inputs and not to require the objects and aspects to be given in separate fields. Finally, an important direction for future improvements is the identification of answer sentences that are more argumentative and a "real" summarization of the answer as one coherent and concise text fragment. We have already started with some further steps into these directions that are presented in the next sections.

2.2 Argument Mining and Retrieval with TARGER

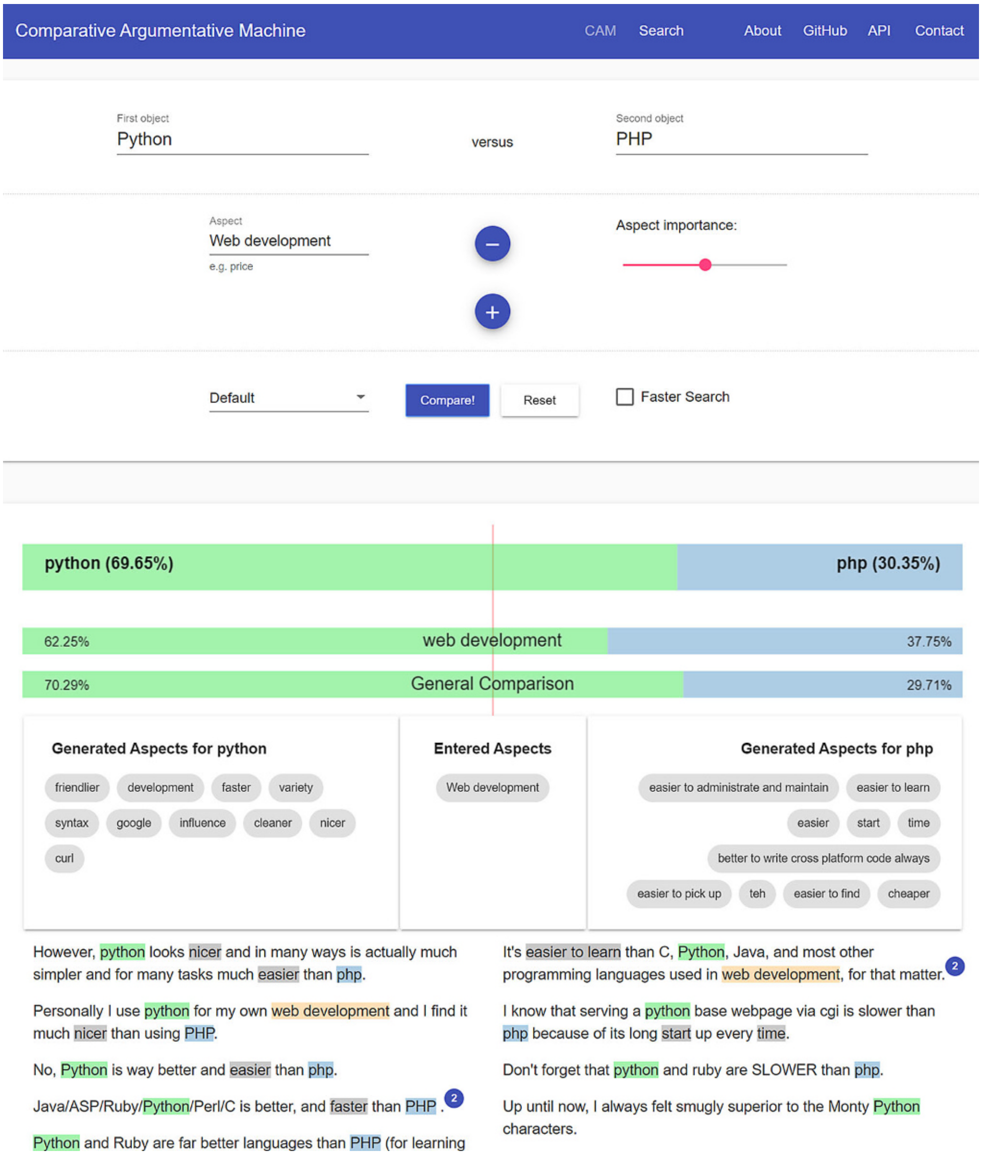
To identify more "argumentative" sentences (or even documents) for the CAM answer, we have developed TARGER [5]: a neural argument tagger, coming with a web interface[5] and a RESTful API. The tool can tag arguments in free text inputs (cf. Fig. 2) and can retrieve arguments from the DepCC corpus that is also used in the CAM prototype (cf. Fig. 3). TARGER is based on a BiLSTM-CNN-CRF neural tagger [10] pre-trained on the persuasive essays (Essays) [7], web discourse (WebD) [8], or IBM Debater (IBM) [9] datasets and is able to identify argument components in text and classify them as claims or premises. Using TARGER's web interface or API, researchers and practitioners can thus use state-of-the-art argument mining without any reproducibility effort (for more details on the implementation and effectiveness, see our respective paper [5]).

2.3 Re-Ranking with Argumentativeness Axioms

To examine the effect of argumentativeness for search, we have experimented with re-ranking results based on their argumentativeness and credibility that are captured via respective preference-inducing axioms (i.e., retrieval constraints for pairs of documents). The argumentativeness axioms use TARGER to tag arguments as premises and claims and then re-rank the top-50 BM25F results with respect to several facets of argumentativeness (e.g., which document contains more argumentative units close to the query terms). We tested the axiomatic re-ranking with a focus on argumentativeness in the TREC 2018 Common Core track [4] and also in the TREC 2019 Decision track [3], where we also added credibility axioms. The results show some encouraging improvements for some

[4] http://ltdemos.informatik.uni-hamburg.de/cam/.

[5] http://ltdemos.informatik.uni-hamburg.de/targer.



First object
Python

versus

Second object
PHP

Aspect
Web development
e.g. price

Aspect importance:

Default

□ Faster Search

python (69.65%) php (30.35%)

62.25% web development 37.75%

70.29% General Comparison 29.71%

Generated Aspects for python: friendlier, development, faster, variety, syntax, google, influence, cleaner, nicer, curl

Entered Aspects: Web development

Generated Aspects for php: easier to administrate and maintain, easier to learn, easier, start, time, better to write cross platform code always, easier to pick up, teh, easier to find, cheaper

However, python looks nicer and in many ways is actually much simpler and for many tasks much easier than php.

Personally I use python for my own web development and I find it much nicer than using PHP.

No, Python is way better and easier than php.

Java/ASP/Ruby/Python/Perl/C is better, and faster than PHP . 2

Python and Ruby are far better languages than PHP (for learning

It's easier to learn than C, Python, Java, and most other programming languages used in web development, for that matter. 2

I know that serving a python base webpage via cgi is slower than php because of its long start up every time.

Don't forget that python and ruby are SLOWER than php.

Up until now, I always felt smugly superior to the Monty Python characters.

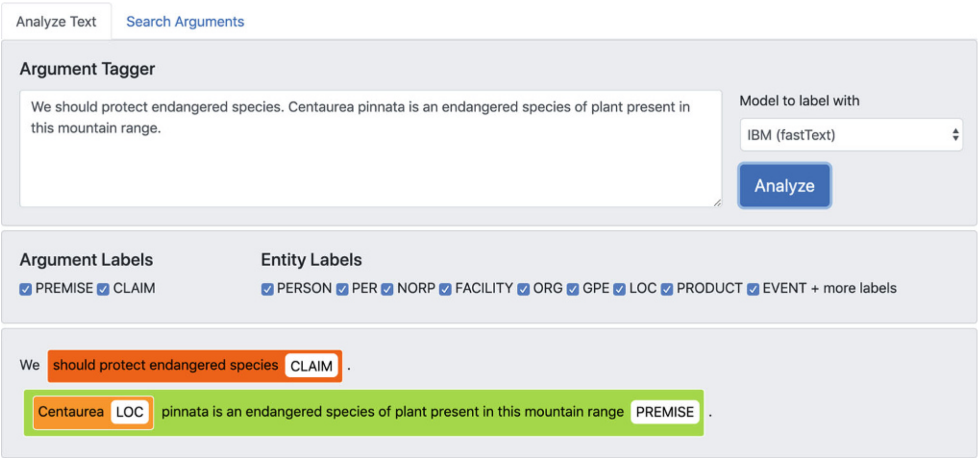
Fig. 1 CAM comparison python vs. php with respect to the aspect web development. Comparison targets and aspects are specified by the user (*upper part of the figure*), results are presented as a high-level overview as well as with detailed evidence from the index in form of snippets (visible when clicking on the output sentences), which are linked to the original web documents

of the TREC topics that we manually identified as potentially "argumentative" while the generalizability to more topics needs some further investigation (for more details on axioms and results, see our respective TREC reports [3, 4]).

2.4 Identifying Comparative Questions

As a first step towards allowing questions as inputs to the CAM prototype, we have studied real comparative questions submitted as queries to the Russian search engine Yandex or posted on the Russian community question answering platform Otvety. We have manually annotated a sample of 50,000 Yandex questions and 12,500 Otvety questions as comparative or not. The comparative questions were further tagged with ten fine-grained labels (e.g., whether the question asks for a fact or arguments) to form a taxonomy of the different comparison intents.

To identify comparative questions, we trained a classifier that can recall 60% of the comparative questions with a perfect precision; we also trained separate classifiers for the fine-grained subclasses. A qualitative analysis after running the classifiers on a one year-long Yandex log of about 1.5 billion questions showed that about 2.8% of the questions are comparative (about one per second with seasonal effects like mushroom comparisons in fall). The majority of the comparison intents cannot be answered by retrieving similar questions from a question answering platform and go way beyond just comparing products or asking for simple facts. A search engine that wants to answer comparative questions in their entirety—like our envisioned CAM system—can thus not just rely on a knowledge graph or on



Analyze Text | Search Arguments

Argument Tagger

We should protect endangered species. Centaurea pinnata is an endangered species of plant present in this mountain range.

Model to label with: IBM (fastText)

Analyze

Argument Labels

☑ PREMISE ☑ CLAIM

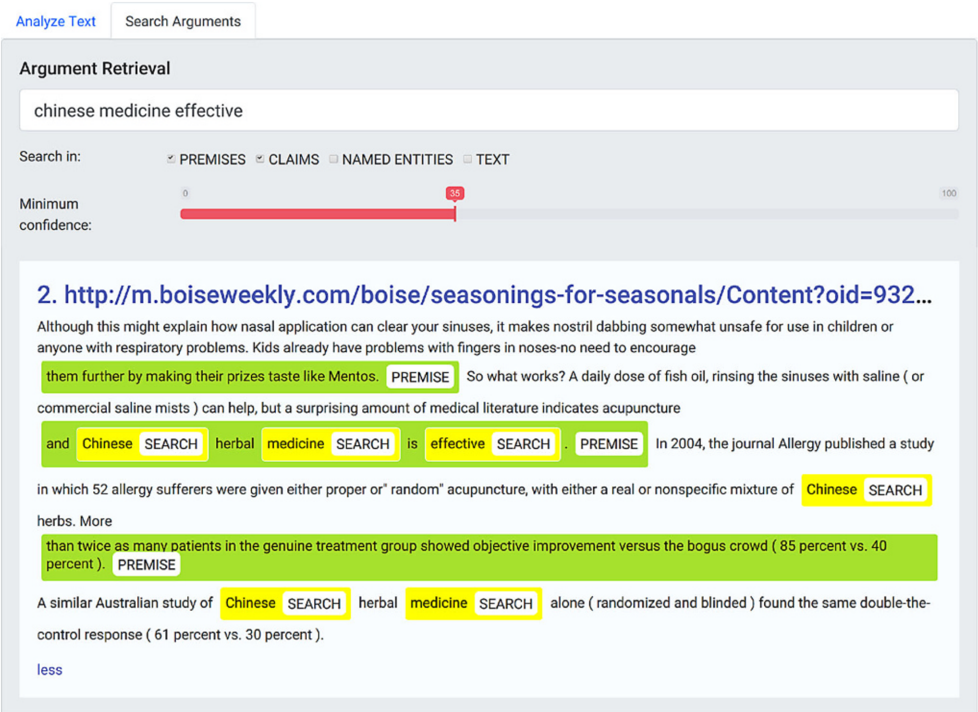
Entity Labels

☑ PERSON ☑ PER ☑ NORP ☑ FACILITY ☑ ORG ☑ GPE ☑ LOC ☑ PRODUCT ☑ EVENT + more labels

We should protect endangered species CLAIM .

Centaurea LOC pinnata is an endangered species of plant present in this mountain range PREMISE .

Fig. 2 Analyze Text with TARGER: input field, drop-down pre-trained model selection, colorized argument labels, a set of entity labels, claim and premises identification, and a tagged result



Analyze Text | Search Arguments

Argument Retrieval

chinese medicine effective

Search in: ☑ PREMISES ☑ CLAIMS □ NAMED ENTITIES □ TEXT

Minimum confidence: 0 35 100

2. http://m.boiseweekly.com/boise/seasonings-for-seasonals/Content?oid=932...

Although this might explain how nasal application can clear your sinuses, it makes nostril dabbing somewhat unsafe for use in children or anyone with respiratory problems. Kids already have problems with fingers in noses-no need to encourage them further by making their prizes taste like Mentos. PREMISE So what works? A daily dose of fish oil, rinsing the sinuses with saline (or commercial saline mists) can help, but a surprising amount of medical literature indicates acupuncture and Chinese SEARCH herbal medicine SEARCH is effective SEARCH . PREMISE In 2004, the journal Allergy published a study in which 52 allergy sufferers were given either proper or" random" acupuncture, with either a real or nonspecific mixture of Chinese SEARCH herbs. More than twice as many patients in the genuine treatment group showed objective improvement versus the bogus crowd (85 percent vs. 40 percent). PREMISE A similar Australian study of Chinese SEARCH herbal medicine SEARCH alone (randomized and blinded) found the same double-the-control response (61 percent vs. 30 percent).

less

Fig. 3 Search Arguments with TARGER: query box, tag selector, and a result with the link to the original document

online question answering platforms (for more details, see our respective paper [1]).

2.5 Touché: Shared Task on Argument Retrieval

To foster and consolidate the research community dealing with argument search and retrieval, we are organizing the Touché lab at CLEF 2020:[6] the first shared task on argument retrieval [2]. The Touché lab has two subtasks: (1) the retrieval of arguments from a focused debate collection to support argumentative conversations, and (2) the retrieval of argumentative documents from a generic web crawl to answer comparative questions with argumentative results.

In the first subtask, we address the scenario of users who directly search for arguments on controversial or socially important topics (e.g., to support their stance or to form a stance) while in the second subtask we address the scenario of personal decisions from everyday life in form of comparative information needs (e.g., “Is X better than Y for Z?” similar to our CAM prototype). For the first subtask, we provide a dataset of more than 380,000 short argumentative text passages crawled from online debate portals, and the task of the lab participants is to retrieve relevant arguments for 50 given topics that cover a wide range of controversial issues. For the second subtask, the dataset

[6] https://touche.webis.de.

is the ClueWeb12, and the task of the lab participants is to retrieve documents that help to answer 50 comparative questions given as the topics.

3 Work in Progress

One of the main limitations of the current CAM prototype is the absence of a natural language interface. A user question like “Should I use Python or Matlab for web development?” still needs to be manually split into the input fields by the user and CAM’s reply is not one coherent passage of text but a collection of individual sentences and an overall score. Ideally, natural language questions could be submitted and the answer with supporting arguments would resemble that of a human expert in the domain like “In your case, I would suggest the open-source Python, since Matlab is rather meant for scientific computing, and many different frameworks for web development are available for Python ... for example Django.” Such a natural language interface for input and output then also would open the perspective of integrating our technology in today’s omnipresent voice-based agents, dialog systems, chatbots, or messengers.

In the final ACQuA project phase, we will be working on the following four tasks to further improve our technology for answering comparative questions.

(1) Extending our analysis of the comparative questions on the Russian web to English questions: We annotate questions from the MS MARCO and Google Natural Questions datasets (Bing and Google queries) and develop approaches to automatically identify the compared objects and aspects using neural models like BERT [6], XLNet [17], and BiLSTM [10]. Based on a reliable identification of the compared objects and aspects in comparative questions, we will then be able to switch to a CAM user interface that can take actual questions as inputs.

(2) Improving the axiomatic re-ranking pipeline: We are currently working on more fine-tuned argumentative axioms that address a wider spectrum of argument facets. The goal is to identify “better” (in terms of argument quality [16] or credibility) pro/con evidences for the compared objects that will then be part of the CAM prototype’s answer.

(3) Improving the CAM prototype’s answer presentation: We are working on hand-crafted templates and automatic summarization of the sentences currently presented in tabular form in the CAM prototype’s answer interface (cf. Fig. 1). For the automatic summaries, we are experimenting with TextRank [11] and text generation via pre-trained language models like GPT-2 [14]. Together with a natural language question input, the more concise natural language output might then enable a human-computer interaction with the CAM prototype via voice interfaces.

(4) Improving CAM’s answers by complementing retrieval from the Common Crawl with structured knowledge bases: We are currently analyzing Wikidata and DBpedia as additional sources of (structured) information besides the retrieval of sentences/documents from the Common Crawl. From the two knowledge bases, we are currently constructing a CAM-specific knowledge graph containing the entities people might want to compare—in a first iteration collected from our analyzed question datasets, but also from Wikipedia “List of” articles (List of car brands, etc.) and the respective properties of the “List of” entities. With the additional CAM-specific knowledge graph, we want to integrate a high-precision structured knowledge source into the current web crawl-based pipeline (high coverage). From a preliminary user study, we could already conclude that using structured information from knowledge bases offers a large potential to improve the CAM answers.

4 Conclusion

The main objective of our DFG-funded project “ACQuA: Answering Comparative Questions with Arguments” is to build a robust argumentation machine that can answer open-domain comparative questions with pros and cons for different options to support informed decision making at the user side.

Our project’s results so far include a working prototype of such a system, a neural argument tagger that allows everyone to use state-of-the-art argument mining via a web interface or an API, a deep analysis of real-world comparative questions, and the organization of the first shared task on argument retrieval.

In the final phase of the project, we will be working on a natural language input and output interface of our current prototype, on an optimization of the axiomatic re-ranking pipeline, and on an integration of a structured knowledge base into the current prototype.

Acknowledgements This work was supported 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) and by the DAAD through the short-term research grant 57314022.

Funding Open Access funding is provided by Projekt DEAL.

References

1. Bondarenko A, Braslavski P, Völske M, Aly R, Fröbe M, Panchenko A, Biemann C, Stein B, Hagen M (2020) Comparative web search questions. Proc. of WSDM, pp 52–60
2. Bondarenko A, Hagen M, Potthast M, Wachsmuth H, Beloucif M, Biemann C, Panchenko A, SteinB (2020) Touché: first shared task on argument retrieval. Proc. of ECIR, pp 517–523
3. Bondarenko A, KasturiaV, FröbeM, VölskeM, SteinB, HagenM (2019) Webis at TREC 2019: decision track. Proc. of TREC.
4. Bondarenko A, Völske M, Panchenko A, Biemann C, Stein B, Hagen M (2018) Webis at TREC 2018: common core track. Proc. of TREC.
5. Chernodub A, Oliynyk O, Heidenreich P, Bondarenko A, Hagen M, Biemann C, Panchenko A (2019) TARGER: neural argument mining at your fingertips. Proc. of ACL, pp 195–200
6. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. Proc. of NAACL-HLT, pp 4171–4186
7. Eger S, Daxenberger J, Gurevych I (2017) Neural end-to-end learning for computational argumentation mining. Proc. of ACL, pp 11–22
8. Habernal I, Gurevych I (2017) Argumentation mining in user-generated web discourse. Comput Linguist 43(1):125–179
9. Levy R, Bogin B, Gretz S, Aharonov R, Slonim N (2018) Towards an argumentative content search engine using weak supervision. Proc. of COLING, pp 2066–2081
10. Ma X, Hovy EH (2016) End-to-end sequence labeling via Bi-directional LSTM-CNNs-CRF. Proc. of ACL, pp 1064–1074
11. Mihalcea R, Tarau P (2004) TextRank: bringing order into text. Proc. of EMNLP, pp 404–411
12. Panchenko A, Bondarenko A, Franzek M, Hagen M, Biemann C (2019at) Categorizing comparative sentences. Proc. of ArgMining, ACL, pp 136–145
13. Panchenko A, Ruppert E, Faralli S, Ponzetto SP, Biemann C (2018) Building a web-scale dependency-parsed corpus from common-crawl. Proc. of LREC, pp 1816–1823
14. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9
15. Schildwächter M, Bondarenko A, Zenker J, Hagen M, Biemann C, Panchenko A (2019) Answering comparative questions: better than ten-blue-links? Proc. of CHIIR, pp 361–365
16. Wachsmuth H, Naderi N, Hou Y, Bilu Y, Prabhakaran V, Thijm TA, Hirst G, Stein B (2017) Computational argumentation quality assessment in natural language. Proc. of EACL, pp 176–187
17. Yang Z, Dai Z, Yang Y, Carbonell JG, Salakhutdinov R, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. Proc. of NeurIPS, pp 5754–5764