

ArguminSci: A Tool for Analyzing Argumentation and Rhetorical Aspects in Scientific Writing

Anne Lauscher,^{1,2} Goran Glavaš,¹ and Kai Eckert²

¹Data and Web Science Research Group
University of Mannheim, Germany

²Web-based Information Systems and Services
Stuttgart Media University, Germany

{anne, goran, simone}@informatik.uni-mannheim.de
{lauscher, eckert}@hdm-stuttgart.de

Abstract

Argumentation is arguably one of the central features of scientific language. We present *ArguminSci*, an easy-to-use tool that analyzes argumentation and other rhetorical aspects of scientific writing, which we collectively dub *scitorics*. The main aspect we focus on is the fine-grained argumentative analysis of scientific text through identification of argument components. The functionality of *ArguminSci* is accessible via three interfaces: as a command line tool, via a RESTful application programming interface, and as a web application.

1 Introduction

Scientific publications are primary means for convincing scientific communities of the merit of one’s scientific work and importance of research findings (Gilbert, 1976). To this end, authors typically present their work by embracing and exploiting established practices and specific tools related to the scientific discourse, such as citations (Gilbert, 1977), that facilitate building persuading argumentation lines. Consequently, scientific texts are abundant with different interrelated rhetorical and argumentative layers. In this work, we refer to this set of mutually-related rhetorical aspects of scientific writing as *scitorics*.

Numerous research groups have already proposed computational models for analyzing scientific language with respect to one or multiple of these aspects. For example, Teufel and Moens (1999) presented experiments on the automatic assignment of *argumentative zones*, i.e., sentential discourse roles, to sentences in scientific articles. Similarly, there has been work on automatic classification of citations with respect to their polarity and purpose (Jha et al., 2017; Lauscher et al., 2017b). It has also been shown that through analyses of *scitorics* higher-level computational tasks

can be supported, such as the attribution of scientific statements to authors (Teufel and Moens, 2000), identification of research trends (McKeown et al., 2016), or automatic summarization of scientific articles (Abu-Jbara and Radev, 2011; Lauscher et al., 2017a).

In this work, we present *ArguminSci*¹ a tool that aims to support the holistic analyses of scientific publications in terms of *scitorics*, including the identification of argumentative components. We make *ArguminSci* publicly available for download.² In its core, it relies on separate neural models based on recurrent neural networks with the long short-term memory cells (LSTM) (Hochreiter and Schmidhuber, 1997) pre-trained for each of the five tasks in the area of scientific publication mining that *ArguminSci* addresses, namely (1) argumentative component identification, (2) discourse role classification, (3) subjective aspect classification, (4) summary relevance classification, and (5) citation context identification. *ArguminSci* is available as a command line tool, through a RESTful HTTP-based application programming interface, and as a web-based graphical user interface (i.e., as a web application).

2 Related Work

We divide the overview of related tools and systems into two categories: (1) systems targeting the analysis of *scitorics* and (2) tools for argument mining (in other domains).

Tools for the Analysis of Scitorics. Ronzano and Saggion (2015) presented the *Dr. Inventor Framework*, which provides end-to-end analysis of scientific text starting with the extraction of text

¹Pronounced like a Polish name and dedicated to Marvin Minsky (https://de.wikipedia.org/wiki/Marvin_Minsky): [aɪgjum'inski].

²<https://github.com/anlausch/ArguminSci>

from PDF documents. The system embeds several modules for mining scientific text, e.g., for the discourse role characterization of sentences. [Sagion et al. \(2017\)](#) presented *MultiScien*, a tool that analyzes scientific text collections in English and Spanish and offers a visualization of discourse categories and summaries. Also, several systems analyzing *argumentative zones* ([Teufel et al., 1999](#)) have been made publicly available (e.g., [Guo et al., 2012](#); [Simsek et al., 2013](#)). However, to the best of our knowledge, *ArguminSci* is the first publicly available system that provides fine-grained argumentative analysis of scientific publications, and allows for a joint analysis of *scitorics* – argumentation and several other rhetorical aspects of scientific language.

Argument Mining Tools. Apart from new research models and approaches, several systems and software tools have been proposed for argument mining in other domains, mainly using machine-learning models at their core. [Wachsmuth et al. \(2017\)](#) developed *args.me*, an argument search engine that aims to support users in finding arguments and forming opinions on controversial topics.³ Another similar system is *ArgumentText* ([Stab et al., 2018](#)). In contrast to *args.me*, the search engine of *ArgumentText* provides access to sentential arguments extracted from large amounts of arbitrary text. The system most similar to *ArguminSci* is *MARGOT* ([Lippi and Torroni, 2016](#)),⁴ which extracts argumentative components from arbitrary text provided by the user. However, *MARGOT* is not tuned for a particular domain and does not perform well on scientific text (i.e., it cannot account for peculiarities of argumentative and rhetorical structures of scientific text). While *MARGOT* focuses only on argumentative components, *ArguminSci* allows for parallel analysis of four other rhetorical aspects of scientific writing.

3 System Overview

We first describe the five annotation tasks that *ArguminSci* covers and the models we train for addressing these tasks. Next, we provide a technical overview of the system capabilities and interfaces through which it is possible to access *ArguminSci*.

³<http://www.argumentsearch.com/>

⁴<http://margot.disi.unibo.it>

3.1 Annotation Tasks and Dataset

Annotation Tasks. Our system supports the following aspects of rhetorical analysis (i.e., automatic annotation) of scientific writing: (1) argument component identification, (2) discourse role classification, (3) subjective aspect classification, (4) citation context identification, and (5) summary relevance classification. Out of these tasks – in accordance with the structure of the annotations in our training corpus – argument component identification and citation context identification are token-level sequence labeling tasks, whereas the remaining three tasks are cast as sentence-level classification tasks.

- *Argument Component Identification (ACI)*: The task is to identify argumentative components in a sentence. That is, given a sentence $\mathbf{x} = (x_1, \dots, x_n)$ with individual words x_i assign a sequence of labels $\mathbf{y}_{aci} = (y_1, \dots, y_n)$ out of the set of token tags Y_{aci} . The label set is a combination of the standard B-I-O tagging scheme and three types of argumentative components, namely *background claim*, *own claim*, and *data*.
- *Discourse Role Classification (DRC)*: Given a sentence \mathbf{x} the task is to classify the role of the sentence in terms of the discourse structure of the publication. The classes are given by the set $Y_{drc} = \{Background, Unspecified, Challenge, FutureWork, Approach, Outcome\}$.
- *Subjective Aspect Classification (SAC)*: Given a sentence \mathbf{x} the task is to assign a single class out of eight possible categories in $Y_{sac} = \{None, Limitation, Advantage, Disadvantage-Advantage, Disadvantage, Common Practice, Novelty, Advantage-Disadvantage\}$.
- *Summary Relevance Classification (SRC)*: Out of the set of possible relevance classes Y_{src} , choose one given a sentence \mathbf{x} , with $Y_{src} = \{Very\ relevant, Relevant, May\ appear, Should\ not\ appear, Totally\ irrelevant\}$.
- *Citation Context Identification (CCI)*: The task is to identify textual spans corresponding to citation contexts. More specifically, given a sentence $\mathbf{x} = (x_1, \dots, x_n)$ the task is to decide on a label for each of the tokens x_i . The possible labels are *Begin_Citation_Context*, *Inside_Citation_Context*, and *Outside*.

Annotation Layer	Labels
Argument Component	<i>Background claim, Own claim, Data</i> (coupled with B-I-O scheme)
Discourse Role	<i>Background, Challenge, Approach, Outcome, Future work</i>
Citation Context	<i>Begin citation context, Inside citation context, Outside</i>
Subjective Aspect	<i>Advantage, Disadvantage, Adv.-disadv., Disadv.-adv., Novelty, Common practice, Limitation</i>
Summarization Relevance	<i>Totally irrelevant, Should not appear, May appear, Relevant, Very relevant</i>

Table 1: Labels of *ArguminSci*’s annotation layers.

Dataset. For training our models, we used an extension of the Dr. Inventor Corpus (Fisas et al., 2015, 2016), which we annotated with fine-grained argumentation structures (Lauscher et al., 2018). The corpus consists of 40 scientific publications in the field of computer graphics and, besides our annotations of argumentative components, offers four layers of annotation, three of which are on the sentence level (DRC, SAC, SRC). Our argument annotation scheme includes three types of argumentative components:

- *Background claim:* A statement of argumentative nature, which is about or closely related to the work of others or common practices in a research field or about background facts related to the topic of the publication.
- *Own claim:* A statement of argumentative nature, which related to the authors own work and contribution.
- *Data:* A fact that serves as evidence pro or against a claim.

More details on the argument-extended corpus we use to train our models can be found in the accompanying resource paper (Lauscher et al., 2018). For more details on the original annotation layers of the Dr. Inventor Corpus, we refer the reader to (Fisas et al., 2015, 2016). In Table 1, we provide the overview of all labels for all five *scitorics* tasks that *ArguminSci* is capable of recognizing.

3.2 Annotation Models.

At the core of *ArguminSci* is a collection of bi-directional recurrent networks with long short-

term memory cells (Bi-LSTMs) (Hochreiter and Schmidhuber, 1997), one pre-trained for each of the five annotation tasks on our argumentatively extended Dr. Inventor corpus (Fisas et al., 2015, 2016; Lauscher et al., 2018).

Model Descriptions. As *ArguminSci* addresses (1) two token-level sequence tagging tasks and (2) three sentence-level classification tasks, the system implements two types of models:

- *Token-level Sequence Labeling:* Given a sentence $\mathbf{x} = (x_1, \dots, x_n)$ with words x_i , we first lookup the vector representations e_i (i.e., pre-trained word embeddings) of the words x_i . Next, we run a Bi-LSTM and obtain the sentence-contextualized representation h_i for each token:

$$h_i = [\overrightarrow{LSTM}(e_1, \dots, e_i); \overleftarrow{LSTM}(e_n, \dots, e_i)].$$

Finally, we feed the vector h_i into a single-layer feed-forward network and apply a softmax function on its output to predict the label probability distribution for each token:

$$y_i = \text{softmax}(Wh_i + b),$$

with $W \in \mathbb{R}^{2K \times |Y|}$ being the weight matrix, $b \in \mathbb{R}^{|Y|}$ the bias vector, and K being the state size of the LSTMs.

- *Sentence-level Classification:* The sentence-level classification builds upon the output of the Bi-LSTM described above: Following Yang et al. (2016), we first obtain a sentence representation by aggregating the individual hidden representations of the words h_i using an intra-sentence attention mechanism defined as

$$s_i = \sum_j \alpha_j h_j.$$

The individual weights α_i are computed as follows:

$$\alpha_i = \text{softmax}(U u_{att}),$$

with u_{att} as the trainable attention head vector and the matrix U containing the Bi-LSTM-contextualized token representations, transformed through a single-layer feed-forward network with non-linear activation (i.e., we first non-linearly transform vectors h_i and stack the transformations to form the matrix U):

```

usage: arguminsci.py [-h] [--argumentation] [--discourse] [--aspect]
                  [--citation] [--summary]
                  inputfile outputfolder

Analyze Argumentation and Rhetorical Aspects in Scientific Writing.

positional arguments:
  inputfile             The name of the textual file containing the input text.
  outputfolder         The name of the output folder where the output should
                      be stored.

optional arguments:
  -h, --help           show this help message and exit
  --argumentation      Extract argument components.
  --discourse          Analyze discourse roles.
  --aspect             Analyze subjective aspects.
  --citation           Extract citation contexts.
  --summary            Assign summary relevance.

```

Figure 1: Help text of *ArguminSci*'s command line interface.

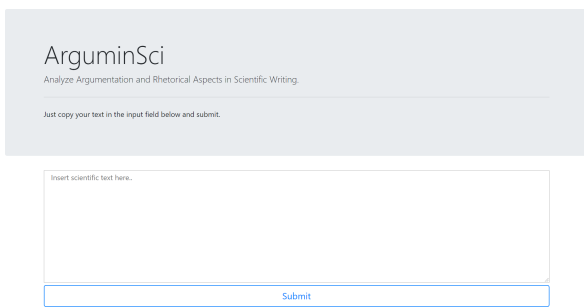


Figure 2: The *ArguminSci* web application offers a simple interfaces for easy analysis of *scitorics*.

$$U_i = \tanh(W_{att}h_i + b_{att}).$$

Analogous to the above-mentioned token-level sequence tagging model, in the last step we apply a feed-forward net with a softmax layer to get the class predictions from the obtained attention-based sentence representation s_i .

We implemented all models in Python, using the Tensorflow framework.⁵

Model Performance. We evaluated the performance of our models on a held-out test set, which comprises 12 randomly selected publications in our corpus (roughly 30% of the corpus, totaling in 2874 sentences). In Table 2 we report the results in terms of F_1 score, macro-averaged over the task labels.

3.3 Interfaces

We offer three different modes of access to *ArguminSci*: (1) using a command line tool, (2) via an RESTful application programming interface, and (3) using a web application.

⁵<https://www.tensorflow.org/>

Task	F1 (%)
Token-level	
Argument Component Identification	43.8
Citation Context Identification	47.0
Sentence-level	
Discourse Role Classification	42.7
Subjective Aspect Classification	18.8
Summary Relevance Classification	33.5

Table 2: Tagging and classification performances.

Command Line Tool. The first interface *ArguminSci* offers is a command line tool, invocable with Python. The script should be provided with two mandatory arguments defining the path to the input file containing the text to be annotated and the path to the output folder where the processing results (i.e., annotated text) will be stored. Furthermore, there are five optional flags which define the type of analysis to perform, each corresponding to one of the *scitorics* tasks. For example, if the user wants to run ACI and DRC on the input text, she should set the flags `--argumentation` and `--discourse`, respectively. Figure 1 shows the help content for the command line tool.

RESTful Application Programming Interface. The application programming interface (API) provides one main HTTP POST end point, which expects a string parameter `text` to be submitted. A second parameter `api_mode` acts as a flag for setting the output format of the predictions (i.e., annotated text) to JSON. A `cURL` request to our RESTful interface has the following format:

```

curl --request POST
--url http://<host>/predict
--data 'text=<text>&api_mode=True'

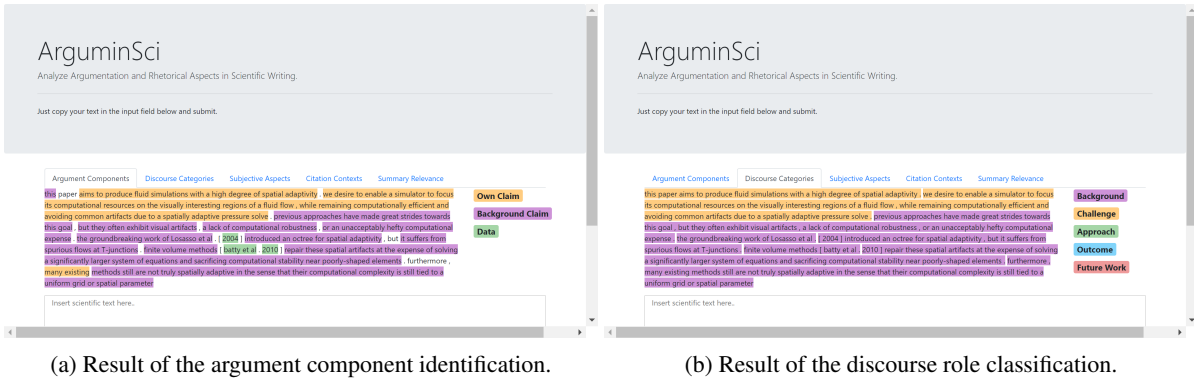
```

For example, given the text "*Our model performs best.*", the API will return a JSON object with the following nested structure:

```

{
  "argumentation":
  [
    [
      [
        "our",
        "BEGIN_OWN_CLAIM"
      ],
      [
        "model",
        "INSIDE_OWN_CLAIM"
      ]
    ]
  ]
}

```



(a) Result of the argument component identification.

(b) Result of the discourse role classification.

Figure 3: The *ArguminSci* web application visualizes the result of the analysis by highlighting the text in different colors. The user can navigate among the different *scitorics* by using the tab control.

```

    ],
    [
      "performs",
      "BEGIN_OWN_CLAIM"
    ],
    [
      "best",
      "INSIDE_OWN_CLAIM"
    ],
    [
      ".",
      "OUTSIDE"
    ]
  ], ...
],
"citation": [...],
"discourse": [...],
"aspect": [...],
"summary": [...]
}

```

In order to enable developers and researchers to use *ArguminSci* as an HTTP service, we make the RESTful API publicly accessible⁶. For the implementation of the API we used the Flask framework in Python.⁷

Web application. Finally, the third option for accessing *ArguminSci* is the web application, based on the template rendering engine Jinja2⁸ and the front-end library Bootstrap.⁹ We adopt a lean and simple design with a single interaction screen. Here, the user can enter the text she desires to annotate with *ArguminSci*'s scitorics annotation models (see Figure 2). Figures 3a and 3b depict the results of the processing. The result is displayed in a tab control in the middle of the screen – different annotation layers can be accessed via

⁶We keep the service endpoint address updated at <https://github.com/anlausch/ArguminSci>

⁷<http://flask.pocoo.org/>

⁸<http://jinja.pocoo.org/docs/2.10/>

⁹<https://getbootstrap.com/>

the tab navigation. The spans of the input text are highlighted with colors indicating different labels, as predicted by the *ArguminSci*'s neural models.

4 Conclusion

Scientific publications, as tools of persuasion (Gilbert, 1977), are highly argumentative and carefully composed texts in which explicit arguments are intertwined with other rhetorical aspects of scientific writing. In this paper, we presented *ArguminSci*, a tool that offers a holistic analysis of scientific publications through a set of rhetorical and argumentative aspects of scientific writing we collectively dub *scitorics*. The *ArguminSci* tool encompasses pre-trained recurrent neural models for two different token-level sequence tagging (identification of argumentative components and citation contexts) and three sentence classification tasks (discourse roles, subjective aspect, and summary relevance).

ArguminSci's functionality can be accessed in three different ways: as a command line tool, via a RESTful application programming interface, and as a web application. In future work, we intend to expose the training phase for the models as well. We also plan to allow for different annotation schemes and to extend the tool with models for other *scitorics* tasks, such as citation purpose and citation polarity classification.

Acknowledgments

This research was partly funded by the German Research Foundation (DFG) with the grant number EC 477/5-1 (LOC-DB). We thank the anonymous reviewers for their constructive and insightful comments.

References

- Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Volume 1*, pages 500–509, Portland, OR, USA. Association for Computational Linguistics.
- Beatriz Fisas, Francesco Ronzano, and Horacio Saggion. 2016. A multi-layered annotated corpus of scientific papers. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 3081–3088, Portorož, Slovenia. European Language Resources Association.
- Beatriz Fisas, Horacio Saggion, and Francesco Ronzano. 2015. On the discursive structure of computer graphics research papers. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 42–51, Denver, CO, USA. Association for Computational Linguistics.
- G Nigel Gilbert. 1976. The transformation of research findings into scientific knowledge. *Social Studies of Science*, 6(3-4):281–306.
- G Nigel Gilbert. 1977. Referencing as persuasion. *Social Studies of Science*, 7(1):113–122.
- Yufan Guo, Ilona Silins, Roi Reichart, and Anna Korhonen. 2012. Crab reader: A tool for analysis and visualization of argumentative zones in scientific literature. In *Proceedings of the 24th International Conference on Computational Linguistics: Demonstration Papers*, pages 183–190, Mumbai, India. The COLING 2012 Organizing Committee.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Rahul Jha, Amjad-Abu Jbara, Vahed Qazvinian, and Dragomir R. Radev. 2017. NLP-driven citation analysis for scientometrics. *Natural Language Engineering*, 23(1):93–130.
- Anne Lauscher, Goran Glavaš, and Kai Eckert. 2017a. University of mannheim@ clscisumm-17: Citation-based summarization of scientific articles using semantic textual similarity. In *CEUR workshop proceedings*, volume 2002, pages 33–42. RWTH.
- Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2018. An argument-annotated corpus of scientific publications. In *Proceedings of the 5th Workshop on Mining Argumentation*, Brussels, Belgium. Association for Computational Linguistics. To appear.
- Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Kai Eckert. 2017b. Investigating convolutional networks and domain-specific embeddings for semantic classification of citations. In *Proceedings of the 6th International Workshop on Mining Scientific Publications*, pages 24–28, Toronto, ON, Canada. Association for Computing Machinery.
- Marco Lippi and Paolo Torrioni. 2016. MARGOT: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303.
- Kathy McKeown, Hal Daume, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R. Fleischmann, Luis Gravano, Rahul Jha, Ben King, Kevin McInerney, Taesun Moon, Arvind Neelakantan, Diarmuid O’Seaghdha, Dragomir Radev, Clay Templeton, and Simone Teufel. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology*, 67(11):2684–2696.
- Francesco Ronzano and Horacio Saggion. 2015. Dr. inventor framework: Extracting structured information from scientific publications. In *Discovery Science*, pages 209–220, Cham. Springer International Publishing.
- Horacio Saggion, Francesco Ronzano, Pablo Accuosto, and Daniel Ferrés. 2017. Multiscien: a bi-lingual natural language processing system for mining and enrichment of scientific collections. In *Proceedings of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries*, volume 1888, pages 26–40, Tokyo, Japan. CEUR-WS.
- Duygu Simsek, Simon Buckingham Shum, Agnes Sandor, Anna De Liddo, and Rebecca Ferguson. 2013. Xip dashboard: visual analytics from automated rhetorical parsing of scientific metadiscourse. In *Proceedings of the 1st International Workshop on Discourse-Centric Learning Analytics*, Leuven, Belgium. Association for Computing Machinery.
- 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. Association for Computational Linguistics.
- Simone Teufel, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the 9th Conference on European Chapter of the Association for Computational Linguistics*, pages 110–117, Bergen, Norway. Association for Computational Linguistics.
- Simone Teufel and Marc Moens. 1999. Discourse-level argumentation in scientific articles: Human and automatic annotation. In *Towards Standards and Tools for Discourse Tagging, Workshop*, Maryland, MA, USA. Association for Computational Linguistics.

- Simone Teufel and Marc Moens. 2000. What's yours and what's mine: Determining intellectual attribution in scientific text. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 9–17, Hong Kong. Association for Computational Linguistics.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, CA, USA. Association for Computational Linguistics.