Finding datasets in publications: The University of Paderborn approach

Rricha Jalota[0000−0003−1517−6394], Nikit Srivastava, Daniel Vollmers, René Speck, Michael Röder, Ricardo Usbeck[0000−0002−0191−7211], and Axel-Cyrille Ngonga Ngomo[0000−0001−7112−3516]

DICE Group, CS Department, Paderborn University, Germany
firstname.lastname@uni-paderborn.de

Abstract. The steadily increasing number of publications available to researchers makes it difficult to keep track of the state of the art. In particular, tracking the datasets used, topics addressed, experiments performed and results achieved by peers becomes increasingly tedious. Current academic search engines render a limited number of entries pertaining to this information. However, having this knowledge would be beneficial for researchers to become acquainted with all results and baselines relevant to the problems they aim to address. With our participation in the NYU Coleridge Initiative's Rich Context Competition, we aimed to provide approaches to automate the discovery of datasets, research fields and methods used in publications in the domain of Social Sciences. We trained an Entity Extraction model based on Conditional Random Fields and combined it with the results from a Simple Dataset Mention Search to detect datasets in an article. For the identification of Fields and Methods, we used word embeddings. In this chapter, we describe how our approaches performed, their limitations, some of the encountered challenges and our future agenda.

1 Literature Review

Previous works on information retrieval from scientific articles are mainly seen in the field of Bio-medical Sciences and Computer Science, with systems [11] built using the MEDLINE[1] abstracts, full-text articles from PubMed Central[2] or ACL Anthology dataset[3]. The documents belonging to the above-mentioned datasets follow a similar format, and thus, several metadata and bibliographical extraction frameworks like CERMINE [10] have been built on them. However, since articles belonging to the domain of Social Sciences do not follow a standard format, extracting key sections and metadata using already existing frameworks like GROBID [7], ScienceParse[4] or ParsCit [3] did not seem as viable options, majorly because these systems were still under development and lacked certain

[1] https://www.nlm.nih.gov/bsd/medline.html
[2] https://www.ncbi.nlm.nih.gov/pmc/
[3] https://www.aclweb.org/anthology/
[4] https://github.com/allenai/science-parse

desired features. Hence, building upon the approach of Westergaard et. al [11], we built our own sections-extraction framework for dataset detection and research fields and methods identification.

Apart from content and metadata extraction, key-phrase or topic extraction from scientific articles has been another emerging research problem in the domain of information retrieval from scientific articles. Jansen et al. [5] extracted core claims from scientific articles by first detecting keywords and key-phrases using rule-based, statistical, machine learning and domain-specific approaches and then applying document summarization techniques. For characterizing a research work in terms of its focus, application domain and techniques used, Gupta et al. [4] proposed applying semantic extraction patterns to the dependency trees of sentences in an article's abstract. On the other hand, to thematically represent scientific articles and for ranking the extracted key-phrases, Mahata et al. [8] devised an approach for processing text documents to train phrase embeddings.

The problem of dataset detection and methods and fields identification is not only different from the ones mentioned above, but also our approach for tackling it is radically disparate. The following sections describe our approach in detail.

2 Project Architecture

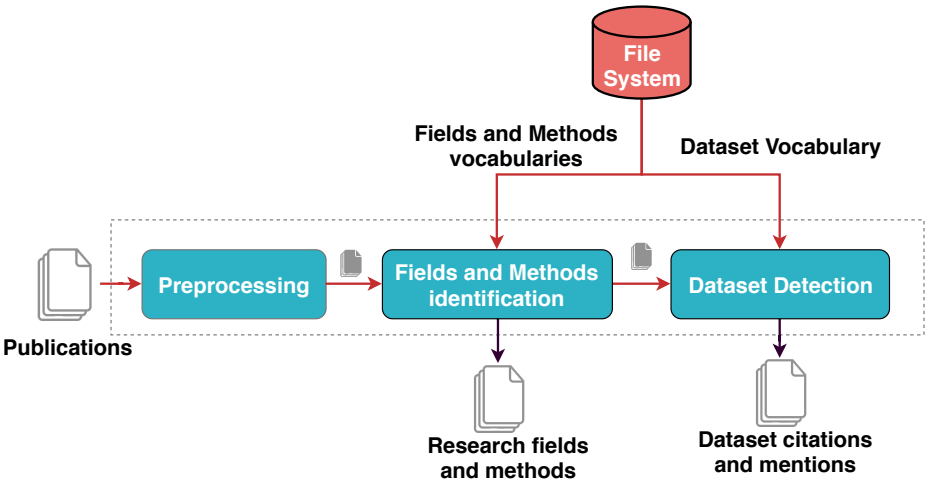

Fig. 9.1: Data Flow Pipeline: Red lines depict the flow of given and generated files between components whereas black lines represent the generation of final output files

Our pipeline (shown in Figure 9.1) consisted of three main components: 1) Preprocessing, 2) Fields and Methods Identification and 3) Dataset Extraction. The Preprocessing module read the text from publications and generated some additional files (see Section 3 for details). These files along with the given Fields

and Methods vocabularies were used to infer Research Fields and Methods from the publications. Then, the information regarding Research Fields was passed onto the Dataset Detection module and using the Dataset Vocabulary, Dataset Citations and Mentions were identified. The following sections provide a detailed overview of each of these components.

3 Preprocessing

As discussed in Chapter 5, the publications were provided in two formats: PDF and text. For Phase-1, we used the given text files, however during Phase-2, we came across many articles in the training files that had not been properly converted to text and contained mostly non-ASCII characters. To work with such articles, we relied on the open source tool `pdf2text` from `poppler suite`[5] to extract text from PDFs. The `pdf2text` command served as the first preprocessing step and was called as a subprocess from within a python script. It was used with `-nopgbrk` argument to generate the text files.

Once we had the text files, we followed the rule-based approach as proposed by Westergaard et al. [11] for pre-processing. The following series of operations based mostly on regular expressions were performed:

- Words split by hyphens were de-hyphenated
- Irrelevant data was removed (i.e., equations, tables, acknowledgment, references);
- Main sections (i.e., abstract, keywords, JEL-Classification, methodology/data, summary, conclusion) were identified and extracted;
- Noun phrases from these sections were extracted (using the python library, spaCy[6]).

We came up with the heuristics for identifying the main sections after going through the articles from different domains in the training data. We collected the surface forms for the headings of all major sections (abstract, keywords, introduction, data, approach, summary, discussion) and applied regular expressions to search for them and separate them from one another. The headings and their corresponding content were stored as key-value pairs in a file. For generating noun-phrases, this file was parsed and for all the values (content) in key-value (heading-content) pairs, a spaCy object, `doc`, was created sentence-wise. Using the built-in function for extracting noun chunks (`doc.noun_chunks`), we generated key-value pairs of heading and noun-phrases found in the content and stored them in another file. This file was later used for fields and methods identification.

To determine how well our approach performed in distinguishing sections, we evaluated it on the articles in the validation dataset. During evaluation, we figured out the limiting cases of our approach. A section could not be differentiated either when there was no explicit mention of any of its surface forms or if there

[5] https://manpages.debian.org/testing/poppler-utils
[6] https://github.com/explosion/spaCy

were multiple mentions of the surface forms in the articles. For instance, in the validation dataset (see Table 9.1), keywords were not extracted from 13 articles because of no explicit mention of the term 'keywords' or its variants. On manual inspection, we found keywords were actually not mentioned in these 13 articles. In the remaining articles where the keywords were present, our algorithm could not detect them from 1 article. For brevity, we have reported only four main sections in Table 9.1: title, abstract, keywords and methodology/data, since these are the ones getting preferential treatment in methods and fields identification. If a section was not found in the article (because of no explicit mention of any of the surface forms), then only the sections that could be detected were extracted. The remaining content was saved as `reduced_content` after cleaning and noun-phrases were extracted from it to prevent loss of any meaningful data.

Table 9.1: Evaluation of identification of sections in Validation Data (100 articles)

Sections	**No explicit mention**	**Mentioned but not distinguished**
Title	0	0
Keywords	13	1
Abstract	0	1
Methodology/Data	18	4

In addition to the main sections, we also extracted PDF metadata using `pdfinfo` service from the `poppler suite` library. The metadata very often contained the keywords and subject of an article, which was helpful in those cases where the keywords were not found by the regular expression.
In the end, the preprocessing module generated four text files for a publication: PDF-converted text, PDF-metadata, processed articles containing relevant data, and noun phrases from the relevant sections, respectively. These files were then passed on to the other two components of the pipeline, which have been discussed below.

4 Approach

4.1 Research Fields and Methods Identification

Vocabulary Generation and Model Preperation

1. **Research Methods Vocabulary**: In Phase-1 of the challenge, we used the given methods vocabulary. However, the feedback that we received from Phase-1 evaluation gave more emphasis to statistical methods used by the authors, references to the time scope, unit of observation, and regression equations rather than the means used to compile the data, i.e., surveys.

Since the given methods vocabulary was not a complete representation of statistical methods and also consisted terms depicting surveys, in Phase-2, we decided to create our own Research Methods Vocabulary using Wikipedia and DBpedia.[7] We manually curated a list of all the relevant statistical methods from Wikipedia[8] and fetched their descriptions from the corresponding DBpedia resources. For each label in the vocabulary, we extracted noun phrases from its description and added them to the vocabulary. Please refer Table 9.2 for examples.

Table 9.2: Examples from manually-curated methods vocabulary

Label	**Description**	**Noun Phrases from Description**
Political forecasting	Political forecasting aims at predicting the outcome of elections.	Political forecasting, the outcome, elections
Nested sampling algorithm	The nested sampling algorithm is a computational approach to the problem of comparing models in Bayesian statistics, developed in 2004 by physicist John Skilling.	algorithm, a computational approach, the problem, comparing models, Bayesian statistics, physicist John Skilling

2. **Research Fields Vocabulary**: For both the phases, we used the given research fields vocabulary and, just like the methods vocabulary, supplemented it with the noun phrases from the description of the research field labels. However, since our phase-1 model seemed to confuse fields with methods, for Phase-2, we additionally created a stopword-list of terms that didn't contain any domain-specific information, such as; Mixed Methods, Meta Analysis, Narrative Analysis and the like.
3. **Word2Vec Model generation**: In this pre-processing step, we used the above-mentioned vocabulary files containing noun phrases to generate a vector model for both research fields and methods. The vector model was generated by using the labels and noun phrases from the description of the available research fields and methods to form a sum vector. The sum vector was basically the sum of all the vectors of the words present in a particular noun phrase. The pre-trained Word2Vec model `GoogleNews-vectors-negative300.bin` [9] was used to extract the vectors of the individual words.
4. **Research Method training results creation**: For research methods, we generated an intermediate result file with the publications present in the training data. It was generated using a `naïve finder algorithm` which, for each publication, selected the research method with the highest cosine

[7] https://wiki.dbpedia.org/services-resources/ontology
[8] https://en.wikipedia.org/wiki/Category:Statistical_methods

similarity to any of its noun phrase's vectors. This file was later used to assign weights to research methods using Inverse Document Frequency.

Processing with Trained Models

- **Finding Research Fields and Methods:** To find the research fields and methods for a given list of publications, we performed the following steps: (At first, Step 1 was executed for all the publications, thereafter Step 2 and 3 were executed iteratively for each publication).
  1. **Naïve Research Method Finder run** - In this step, we executed the `naïve research method finding algorithm` (i.e. selected a research method based on the highest cosine similarity between vectors) against all the current publications and then merged the results with the existing result from the `research methods' preprocessing step`. The combined result was then used to generate IDF weight values for each `research method`, to compute the significance of recurring terms.
  2. **IDF-based Research Method Selection** - We re-ran the algorithm to find the closest research method to each noun phrase and then sorted the pairs based on their weighted cosine similarity. The weights were taken from the IDF values generated in the first step and the manual weights assigned (section-wise weighting). Here, the noun phrases that came from the methodology section and from the methods listed in JEL-classification (if present) were given a higher preference. The pair with the highest weighted cosine similarity was then chosen as the Research Method of the article.
  3. **Research Field Finder run** - In this step, we first found the closest research field from each noun phrase in the publication. Then we selected the Top N (= 10) pairs that had the highest weighted cosine similarity. Afterwards, the noun phrases that had a similarity score less than a given threshold (= 0.9) were filtered out. The end-result was then passed on to a post-processing algorithm.
     For weighted cosine similarity, the weights were assigned manually based on the section of publication from which the noun phrases came. In general, noun phrases from title and keywords (if present) were given a higher preference than other sections, since usually these two sections hold the crux of an article. Note, if sections could not be discerned from an article, then noun phrases from the section, reduced_content (see section 3), were used to find both fields and methods.
  4. **Research Field Selection** - The top-ranked term from the result of step 3, which was not present in the stopword-list of irrelevant terms, was marked as the research field of the article.

The experimental set-up and average training times (ATT) have been reported in Table 9.3:

Table 9.3: Experimental set-up for training the word2vec models for Research Field (RF) and Methods (RM) Identification

Computing Infrastructure	macOS, 2 GHz Intel Core i7 processor, 4 cores RAM 16 GB 1600 MHz
ATT - RF model	3m 21s
ATT - RM model	3m 19s
Link to Implemented Code	https://github.com/nikit91/Jword2vec/tree/rich-context

4.2 Dataset Extraction

For identifying the datasets in a publication, we followed two approaches and later combined results from both. Both the approaches have been described below.

1. **Simple Dataset Mention Search:** We chose the dataset citations from the given Dataset Vocabulary that occurred for one dataset only and used these unique mentions to search for the corresponding datasets using regular expressions in the text documents. Then, we computed a frequency distribution of the datasets. As can be seen from Figure 9.2, certain dataset citations occurred more often than others. This is because while searching for dataset citations, apart from the dataset title, the corresponding mention_list from Dataset Vocabulary was also considered, which contained many commonly occurring terms like 'time', 'series', 'time series', 'population' etc. Therefore, we filtered out those dataset citations that occurred more than a certain threshold value (=1.20) multiplied by the median of the frequency distribution and that had less than 3 distinct mentions in a publication. The remaining citations were written to an interim result file. Table 9.4 depicts the improvement in performance of Simple Dataset Mention Search with the inclusion of filtering. The filtering process improved the F1-measure by 42.86%. Note, as the validation data consisted of only 100 articles, changing the threshold value to 1.10 or 1.30 didn't result in any significant change, hence we have maintained a constant threshold value of 1.20 in our comparison table.

Table 9.4: Performance of Simple Dataset Mention Search against Validation Data.

Metrics	**without filtering**	**Threshold=1.20, mentions < 3**	**Threshold=1.20, mentions < 4**
Precision	0.09	0.71	0.09
Recall	0.28	0.12	0.28
F1-score	0.14	**0.20**	0.14

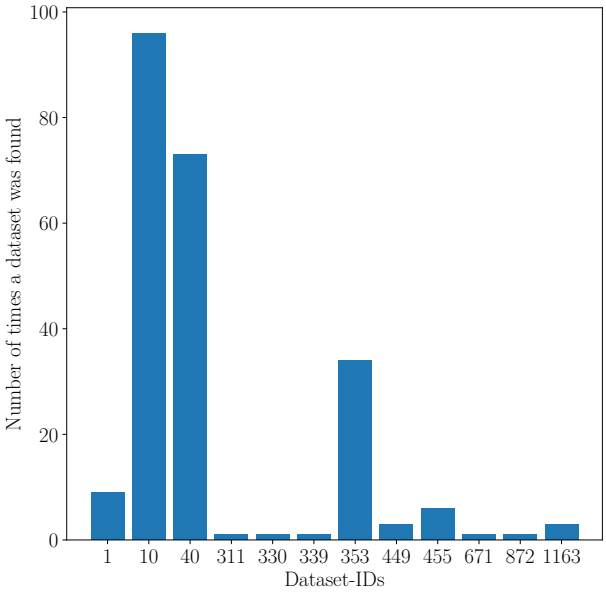

Fig. 9.2: Frequency Distribution of Dataset Citations

2. **Rasa-based Dataset Detection:** In our second approach, we trained an entity extraction model based on conditional random fields (CRF) using Rasa NLU [1]. For training the model we used the Spacy Tokenizer[9] for the preprocessing step. For Entity Recognition we used BILOU tagging and used 50 iterations to train the CRF. We used the Part of Speech tags, the case of the input tokens and the suffixes of the tokens as input features for the CRF model. We particularly tested two configurations for training the CRF-based Named Entity Recognition (NER) model. In Phase-1, the 2500 labeled publications from the training dataset were used for training the Rasa NLU[10] model. Later in Phase-2, when the Phase-1 holdout corpus was released, we combined its 5000 labeled publications with the previously given 2500 labeled publications and then retrained the model again with these 7500 labeled publications.
Running the CRF-Model: The trained model was run against the pre-processed data to detect dataset citations and mentions. Only the entities that had a confidence score greater than a certain threshold value (= 0.72) were considered as dataset mentions. A dataset mention was considered as a citation only if it was found in the given Dataset Vocabulary (via string matching either with a dataset title or any of the terms in a dataset 'mention_list') and if it belonged to the research field of the article. To check if

[9] https://spacy.io/api/tokenizer
[10] https://rasa.com/docs/nlu

a dataset belonged to the field of research, we found the cosine similarity of the terms in the 'subjects' field of the Dataset Vocabulary with the keywords and the identified Research Field of the article.

3. **Combining the two approaches:** The output generated by the Rasa-based approach was first checked for irrelevant citations before a union was performed to combine the results. This was done by checking if a given dataset_id occured more than a threshold value (= 1.20) multiplied by the median of the frequency distribution (same as the filtering process of the Simple Dataset Mention Search).

Note that, the threshold values mentioned above were set after some experiments of trial and testing. For dataset extraction, the goal was to keep the number of false positives low while not compromising the true positives. For research methods and fields, a manual evaluation (see the next section for details) was done to test if the results made sense with the articles.

5 Evaluation

We performed a quantitative evaluation for Dataset Extraction using the evaluation script provided by the competition organizers (refer Chapter 5 for more details). This evaluation (see Table 9.5) was carried out against the validation data, wherein we compared four different configurations. As can be inferred from the table, there was only a slight increase in performance for the Rasa-based model, when the training samples were increased. However, combining it with the Simple Dataset Mention Search, increased the performance by *19.42%*. Interestingly, there was no improvement in performance in the combined approach even when the training samples for the Rasa-based model were increased. This might be because of the removal of frequently-occuring terms from the Rasa-generated output, based on the frequency distribution of dataset mentions as computed in the Simple Dataset Mention Search.

Table 9.5: Quantitative Evaluation of Datasets against Validation Data. (The numbers inside brackets indicate training samples)

	Phase-1	**Phase-2**		
Metrics	**Rasa-based Approach** (2500)	**Rasa-based Approach** (7500)	**Combined Approach** (2500)	**Combined Approach** (7500)
Precision	0.382	0.388	**0.456**	**0.456**
Recall	0.26	0.26	**0.31**	**0.31**
F1	0.309	0.311	**0.369**	**0.369**

For Research Fields and Methods, we carried out a qualitative evaluation against 10 randomly selected articles from Phase-1 holdout corpus. Tables 9.6 and 9.7 depict a comparison between the predicted fields and methods in Phase-1 and Phase-2. In general, our models returned a more granular output in the second phase, solely because of the modifications we made in the vocabularies.

Table 9.6: Evaluation of Research Fields against Phase-1 holdout

pub_id	**Keywords**	**Phase-1**	**Phase-2**
10328	Cycling for transport, leisure and sport cyclists	Health evaluation	**Public health and health promotion**
7270	Older adult drug users, harm reduction	Health Education	**Correctional health care**
6053	Economic conditions - crime relationship, homicide	Homicide	**Gangs and crime**

Table 9.7: Evaluation of Research Methods against Phase-1 holdout

pub_id	**Keywords**	**Phase-1**	**Phase-2**
10328	Thematic content analysis	Thematic analysis	**Sidak correction**
7270	Interviews conducted face to face, finding systematic patterns or relationships among categories identified by reading the interview transcript	Qualitative interviewing	**Sampling design**
6053	Autoregressive integrated moving average (ARIMA) time-series model	Methodological pluralism	**Multivariate statistics**

6 Discussion

Throughout the course of this competition, we encountered several challenges and limitations in all the three stages of the pipeline. In the preprocessing step, the appropriate extraction of text from PDFs turned out to be rather challenging. This was especially due to the varied formats of the publications, which made the extraction of specific sections—that contained all data relevant to our work—demanding. As mentioned before, if there was no explicit mention of the key-terms like `Abstract, Keywords, Introduction, Methodology/Data, Summary, Conclusion` in the text, then the content was saved as 'reduced_content' after applying all other preprocessing steps and filtering out any irrelevant data.

Our experiments suggest that the labeled publications we received for dataset detection were not uniform in the dataset mentions provided, which made it difficult to train an entity extraction model even with an increased number of training samples. Hence, there was only a slight improvement in performance when the Rasa-model was trained with 7500 publications instead of 2500. This was also why we combined the Rasa-based approach with the Simple Dataset Mention Search, so that at least the datasets that were present in the vocabulary do not get missed.

Regarding the fields and methods, vocabularies played an immense role in their identification. The vocabularies that were provided by the SAGE publications contained some terms that were either polysemous or very high-level and therefore, were picked up by our model very often. Hence, for research methods, we created our own vocabulary containing all the relevant statistical methods, and for fields, we introduced a stopword-list of irrelevant terms and looked it up each time, before writing the result to the output file. The goal of stopword-list generation was to filter the terms that did not carry domain-specific information and sounded more like research methods than fields. Since the focus was on more granulated results, we tried to look for open ontologies for Social Science Fields and Methods and unfortunately, could not find any. It is worth mentioning that since our approach for Fields and Methods identification relied heavily upon vocabularies, it could not find any new methods or fields from the publications.

Based on the final evaluation feedback, since our Phase-2 models did not perform as good as we expected, following are a few things that we could have done differently.

1. For research methods, merging the given SAGE methods vocabulary with our manually curated vocabulary, could have resulted in methods that would have been both granular and statistical while still being relevant to the publications. Introducing a stopword-list just as we did for research field identification, could also have been another workaround.
2. For both fields and methods identification, we could have also tried pretrained embeddings from glove[11] and fastText[12].
3. As our entity-extraction approach for Dataset Detection suffered from a limitation of inconsistent labels (i.e. datasets mentioned in the form of abbreviation, full-name, collection procedure, and keywords) in training data, we could have extended the Simple Dataset Mention Search to a pattern-oriented search based on handcrafted rules derived from sentence structure and other heuristics.

7 Future Agenda

The data provided to us in the competition displayed a cornucopia of inconsistencies even after human processing. We hence propose that machine-aided

[11] https://nlp.stanford.edu/projects/glove
[12] https://fasttext.cc/docs/en/crawl-vectors.html

methods for computing correct and complete structured representation of publications are of central importance for scientific research such as an Open Research Knowledge Graph [6][2]. Previous works on never-ending learning have shown how humans and extraction algorithms can work together to achieve high-precision and high-recall knowledge extraction from unstructured sources. In our future work, we hence aim to populate a **scientific knowledge graph** based on never-ending learning. The methodology we plan to develop will be domain-independent and rely on active learning to classify, extract, link and publish scientific research artifacts extracted from open-access papers. Inconsistency will be remedied by ontology-based checks learned from other publications such as SHACL constraints which can be manually or automatically added.[13] The resulting graphs will

- rely on advanced distributed storage for RDF to scale to the large number of publications available;
- be self-feeding, i.e., crawl the web for potentially relevant content and make this content available for processing;
- be self-repairing, i.e., be able to update previous extraction results based on insights gathered from new content;
- be weakly supervised by humans (e.g. authors of publications), who would assist in correcting wrong hypotheses, thereby leveraging semi-supervised learning;
- provide standardized access via W3C Standards such as SPARQL.

Having such knowledge graphs would make it easier for the researchers (both young and veteran) to easily follow along with their domain of fast-paced research and eliminate the need to manually update the domain-specific ontologies for fields, methods and other metadata as new research innovations come up.

8 Appendix

The code and documentation for all our submissions can be found here: https://github.com/dice-group/rich-context-competition.

References

1. Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *CoRR*, abs/1712.05181, 2017.
2. Davide Buscaldi, Danilo Dessì, Enrico Motta, Francesco Osborne, and Diego Reforgiato Recupero. Mining scholarly data for fine-grained knowledge graph construction. In *Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019), Portoroz, Slovenia, June 2, 2019.*, pages 21–30, 2019.

[13] https://www.w3.org/TR/shacl/

3. Isaac G Councill, C Lee Giles, and Min-Yen Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, volume 8, pages 661–667, 2008.
4. Sonal Gupta and Christopher Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th international joint conference on natural language processing*, pages 1–9, 2011.
5. Tom Jansen and Tobias Kuhn. Extracting core claims from scientific articles. In *Benelux Conference on Artificial Intelligence*, pages 32–46. Springer, 2016.
6. Mohamad Yaser Jaradeh, Sören Auer, Manuel Prinz, Viktor Kovtun, Gábor Kismihók, and Markus Stocker. Open research knowledge graph: Towards machine actionability in scholarly communication. *CoRR*, abs/1901.10816, 2019.
7. Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer, 2009.
8. Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 634–639, 2018.
9. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
10. Dominika Tkaczyk, Pawel Szostek, Piotr Jan Dendek, Mateusz Fedoryszak, and Lukasz Bolikowski. Cermine–automatic extraction of metadata and references from scientific literature. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 217–221. IEEE, 2014.
11. David Westergaard, Hans Henrik Stærfeldt, Christian Tønsberg, Lars Juhl Jensen, and Søren Brunak. A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts. *PLoS Computational Biology*, 14(2), 2018.