Scientific Paper Advisor
a tool for analyzing scientific publications and their links

(Scientific Paper Advisor - narzędzie do analizy artykułów naukowych i ich powiązań)

Cezary Troska Antoni Dąbrowski

Praca inżynierska

Promotor: dr. Paweł Rychlikowski

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

06.02.2023

Abstract

Google Scholar is a useful tool for obtaining scientific articles on any subject. For our engineering thesis, we decided to create a program extending Scholar's functionalities. Our web browser extension Scientific Paper Advisor lets the user see the scientific articles within the context of references and uses of the publication, and makes searching for topics for new papers easier by extracting research suggestions left by authors in recently publicized studies. SPA is available in the Web Store, the source code was published on Github, and this thesis describes the premise, development process, and technical details of the project.

Google Scholar jest użytecznym instrumentem do odnajdowania artykułów naukowych na dowolny temat. W ramach pracy dyplomowej podjeliśmy się stworzenia narzędzia rozszerającego jego funkcje. Wtyczka Scientific Paper Advisor pozwala na zobaczenie prac badawczych w kontekście ich odniesień i cytowań, oraz łatwe znalezienie nowych tematów do badań dzięki sugestiom pozostawionym w najnowszych publikacjach przez ich autorów. SPA jest już dostępne w Web Store, kod źródłowy został upubliczniony w serwisie Github, a ten dokument opisuje założenia, proces powstawania oraz szczeóły techniczne projektu.

Contents

1 Introduction **7**

1.1 Concept and premise 7

1.2 User needs assessment 7

1.2.1 User profile 7

1.2.2 Use cases and user stories 8

2 Application overview **11**

2.1 Functionalities provided 11

2.2 Usage 11

2.2.1 Instllation 12

2.2.2 Custom backend – optional step 16

2.2.3 Example of usage 16

3 Implementation details **27**

3.1 Data gathering and web scraping 27

3.1.1 Google Scholar 27

3.1.2 Further research database 29

3.1.3 Article popularity database 29

3.2 Frontend 30

3.2.1 Browser extension 30

3.2.2 Layout 31

3.2.3 Backend communication 32

3.2.4 Results visualization 33

3.3 Backend 33
3.3.1 Django-based application 33
3.3.2 PostgreSQL database 34
3.3.3 Dockerization 35
3.3.4 Google Cloud deployment 36

4 Popularity prediction 37

4.1 Dataset 37
4.2 Objective 38
4.3 Feature engeneering 39
4.3.1 Stylistic features 39
4.3.2 Semantic features 40
4.3.3 Side information 40
4.4 Model – training and evaluation 42

5 Analysis of 'further research' segments 43

5.1 Category prediction 43
5.2 Clusterization procedure 44
5.2.1 Dimensionality reduction 45
5.2.2 t-SNE vs PCA 47

6 Conclusions 49

6.1 Further development 49

Chapter 1

Introduction

1.1 Concept and premise

The topic of the project came from our experiences as students. Google Scholar was invaluable in collecting information for presentations and seminars, so expanding on its functionality will likely benefit the student community. Easy access to articles of interest and ones derived from them is something that shall prove useful for the wider scientific community (as we described in the following sections).

The other part of the tool is providing suggestions for new scientific studies. It has a direct connotation with our case – deciding what topic to work on was not an easy choice. We hope research suggestions found with this extension will help others with similar dilemmas.

1.2 User needs assessment

The main goal of our work is to create an application that would be useful and would provide valuable functionality. In order to do that, we had to first identify who our target user is, and what needs our extension would satisfy.

1.2.1 User profile

Our application is dedicated to:

- students, researchers, and science enthusiasts,
- people who want to get their technical knowledge from the source,
- people who want to keep up to date with all the latest science news.

1.2.2 Use cases and user stories

In this section, we are going to provide the user stories i.e. a general description of software features written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

Use cases map the structure of a system as it is seen by its users. It specifies the behaviour of an application, by describing performed actions in order to deliver the solution to a specific user problem.

Case I - category analysis Let us imagine a student fascinated with computer engineering, especially computer architecture. He wants to know all about his machine. Where they came from? What was the process of its development? How operating systems were made? He is soon faced with the challenge of insufficient knowledge sources. The pop science online articles were too little for him. How can he effectively navigate in the vast ocean of scientific literature?

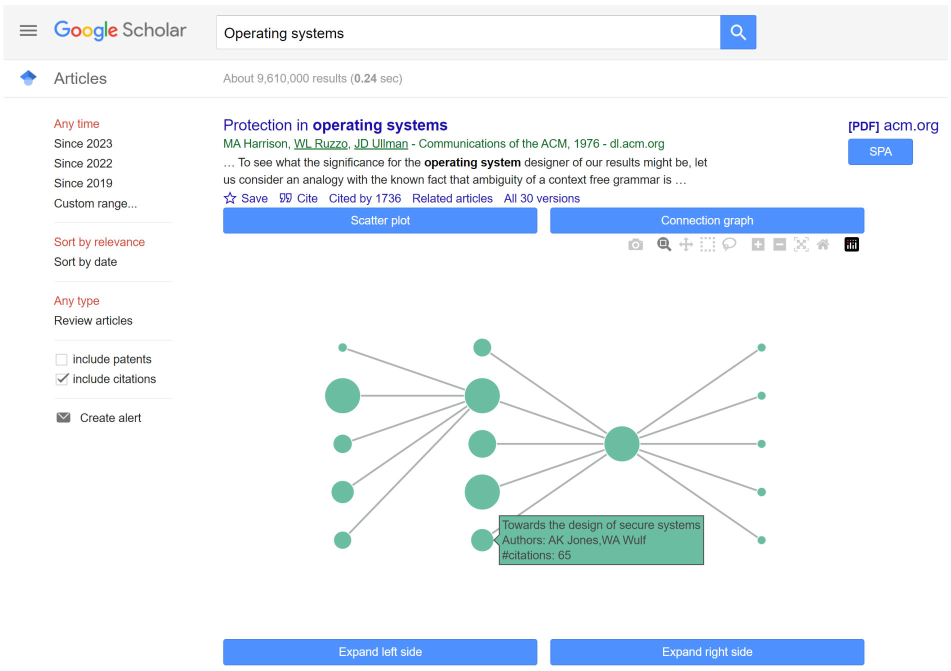
With our extension, he can simply identify the most distinguishable papers in a particular domain of interest. He can also track the history of the development of any computer part as well as side inventions. On top of that, the display is interactive, intuitive and easy to use.

When a student searches for "Operating systems" in the Google Scholar query the browser reveals a ranking of publications. However, none of them is a good place to start. Each seems too specific for a newcomer. The student decides to use our extension on one of the found articles. It takes a while, but the information given by the extension reveals insight into how this domain was explored in previous and later research. Visualization of user actions can be seen in Figure 1.1.

Case II - researcher view The user is a novice scientist who specialises in natural language processing. He has significant scientific knowledge but is not very familiar with proceeding research, attempts to solve current main problems and who are the big names in its science niche. Even though he can find simple answers to all of these problems, they still do not provide an understanding of sophisticated connotations of different names, publications, and journals. Also, articles found on Google were probably too short on content and deeper analysis, which is essential for him.

That is where our search engine extension finds its application. It not only provides a list of direct relations between publications and their authors but also displays in semantic-similarity space the current major problems in the specific domain. It is an operation that cannot be done easily by hand as it requires collecting a significant database of the latest articles and using advanced models to analyse them.

The use case might look as follows. The user tries to find the most prominent publications from the past few years and see how he can extend their work. He



Operating systems

About 9,610,000 results (**0.24** sec)

Protection in **operating systems**

MA Harrison, WL Ruzzo, JD Ullman - Communications of the ACM, 1976 - dl.acm.org

… To see what the significance for the **operating system** designer of our results might be, let us consider an analogy with the known fact that ambiguity of a context free grammar is …

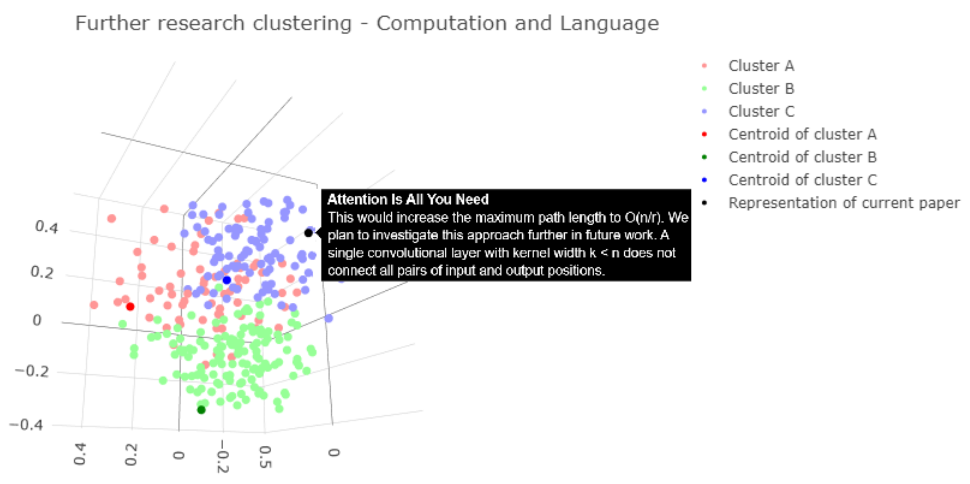
Towards the design of secure systems
Authors: AK Jones,WA Wulf
#citations: 65

Figure 1.1: Analysis of the first record – "Protection in operating systems" (central node) shows previous publications. Among them "Towards the design of secure systems", which is more general and can be used as an introduction to the topic of the main article.

quickly comes across the title "Attention is all you need" which is one of the greatest publications in its domain. However, it is now a few years old. In order to see more recent publications that pushed its work further the user asks the Scientific Paper Advisor for an extended analysis of this publication. After pushing the extension button, a scatterplot starts visualizing over 150 publications from the last two years that are treating about natural language processing. The plot displays not only those research papers but also their segments containing information about potential further directions of development. The process visualized in Figure 1.2

Case III - fun and exploration The user is a science enthusiast who just realised that all researches ever proceeded have its own unique structure – that all the publications are somehow connected to each other. That the world of scientific papers has a particular order and its rules are manifested in citations. But what to do with that knowledge? How to explore such a broad domain?

Scientific Paper Advisor displays small parts of the 'science graph' and gives an opportunity to traverse it freely with an eye-pleasing display. It shows papers' correlations much better than a bibliography. The user can move back and forth. Check what are the origins of a certain domain and what are the latest achievements.



Attention Is All You Need
This would increase the maximum path length to O(n/r). We plan to investigate this approach further in future work. A single convolutional layer with kernel width k < n does not connect all pairs of input and output positions.

Figure 1.2: Visualization of publication and its sections consisting descriptions of future work. All in a semantic-similarity space, the distance between points is inversely proportional to the similarity of the article subject.

Chapter 2

Application overview

This chapter provides a high-level description of the extension's functionalities and illustrates them with some usage examples. Details of those processes are presented in Chapters 3, 4, and 5.

2.1 Functionalities provided

- Presenting a chosen article in the context of its references and citations, as a graph.
- Providing basic information and links about the affiliated articles.
- Expanding relationship graphs to include secondary references or citations. Such expansion can be repeated an unlimited amount of times, restricted only by the existence of and access to the records requested.
- Presenting a summary of the most recent, important research topics in the same category[1] as the selected article. The summary includes the suggestion given by researchers, the abstract, and some other basic information.
- Links to all the articles returned by any function of the extension.
- An option to use custom backend[2] for collecting, processing, and storing article data.

2.2 Usage

Let us go through an example of how to use the SPA tool.

[1]The categories we used for this project follow arXiv's Category Taxonomy

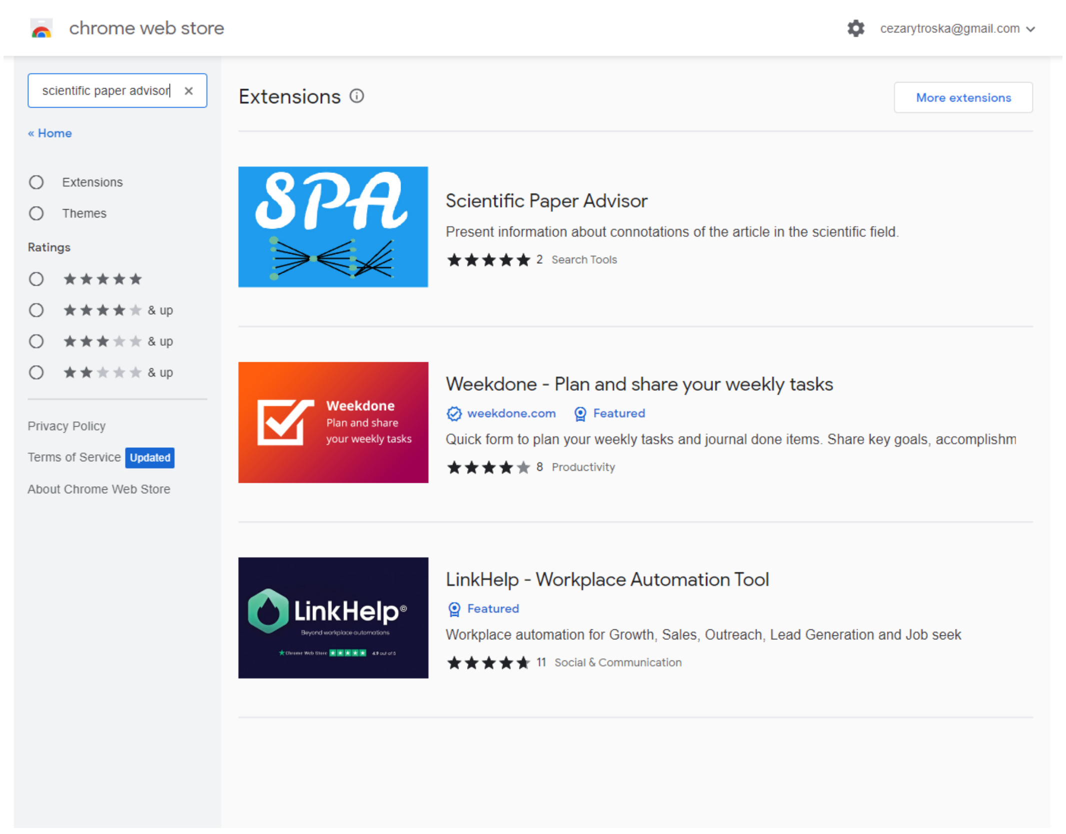
[2]All the resources needed for deployment of a custom backend server for SPA[3] extension are provided in the project's repository.

2.2.1 Instllation

The user can acquire the extension in two ways.

Installing from the Web Store

The easiest method is to get the extension from Chrome's Web Store. Our product can be found there under the title 'Scientific Paper Advisor'.



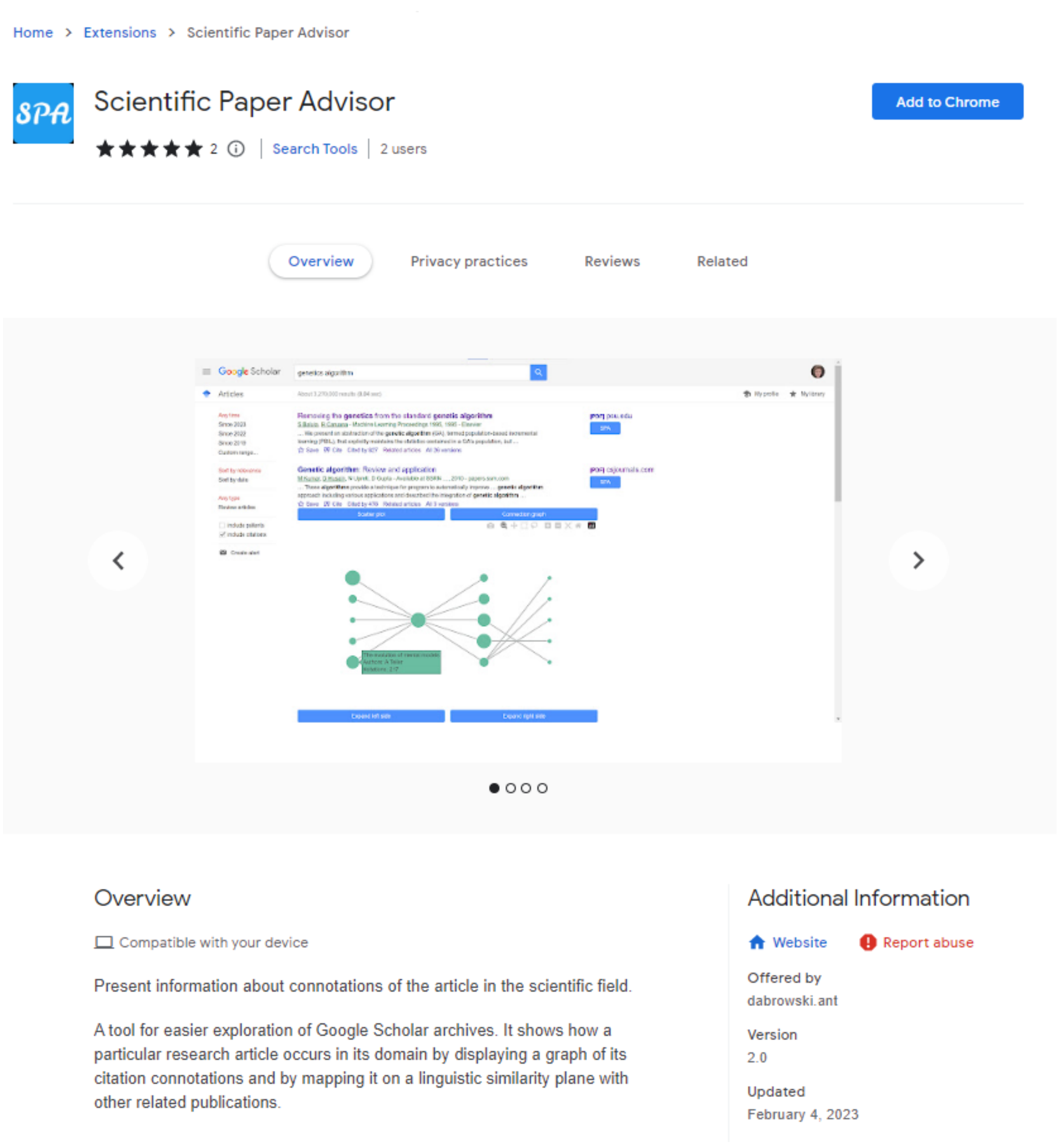
scientific paper advisor

Scientific Paper Advisor
Present information about connotations of the article in the scientific field.
★★★★★ 2 Search Tools

Weekdone - Plan and share your weekly tasks
weekdone.com Featured
Quick form to plan your weekly tasks and journal done items. Share key goals, accomplishm
★★★★☆ 8 Productivity

LinkHelp - Workplace Automation Tool
Featured
Workplace automation for Growth, Sales, Outreach, Lead Generation and Job seek
★★★★★ 11 Social & Communication

Figure 2.1: Search results for Scientific Paper Advisor on Chrome WebStore



Home > Extensions > Scientific Paper Advisor

Scientific Paper Advisor

★★★★★ 2 | Search Tools | 2 users

Add to Chrome

Overview | Privacy practices | Reviews | Related

Overview

Compatible with your device

Present information about connotations of the article in the scientific field.

A tool for easier exploration of Google Scholar archives. It shows how a particular research article occurs in its domain by displaying a graph of its citation connotations and by mapping it on a linguistic similarity plane with other related publications.

Additional Information

Website | Report abuse

Offered by
dabrowski.ant

Version
2.0

Updated
February 4, 2023

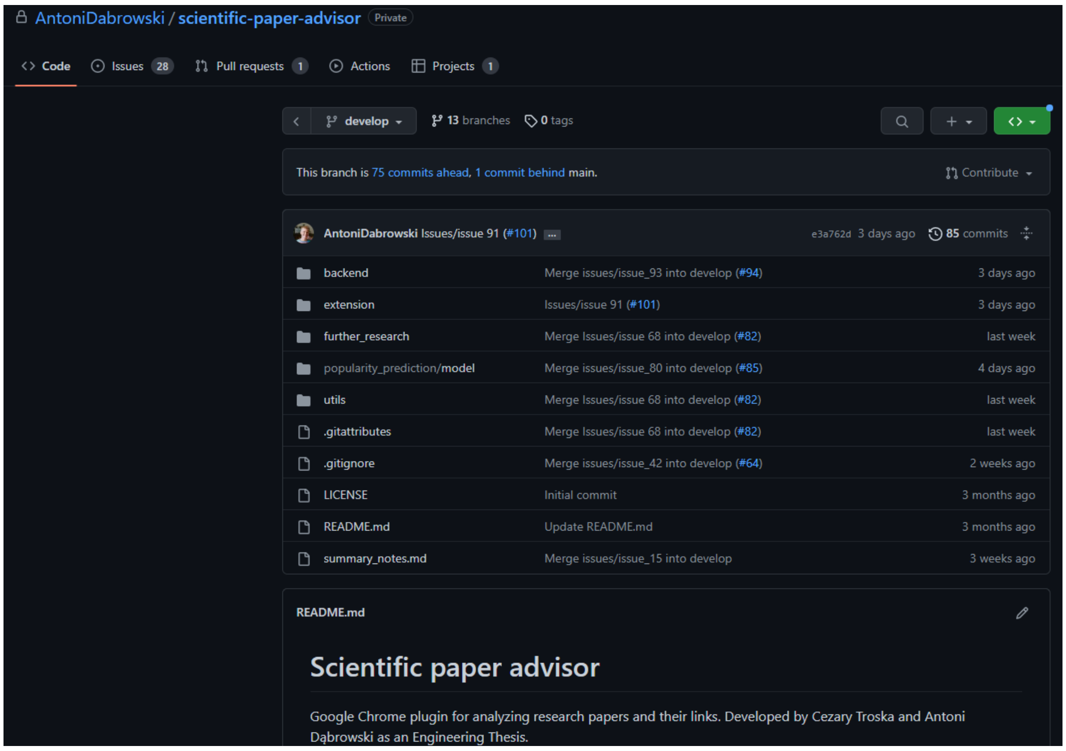
Figure 2.2: Scientific Paper Advisor's page on the WebStore

Then the user shall click on the 'Add to Chrome' button and the browser will take care of the installation process.

Installing from source

Our tool is an open-source project, so anyone can deploy the extension from the source, with potential customizations.

To do that clone the repository.



This branch is 75 commits ahead, 1 commit behind main.

AntoniDabrowski Issues/issue 91 (#101) e3a762d 3 days ago 85 commits

backend	Merge issues/issue_93 into develop (#94)	3 days ago
extension	Issues/issue 91 (#101)	3 days ago
further_research	Merge Issues/issue 68 into develop (#82)	last week
popularity_prediction/model	Merge issues/issue_80 into develop (#85)	4 days ago
utils	Merge Issues/issue 68 into develop (#82)	last week
.gitattributes	Merge Issues/issue 68 into develop (#82)	last week
.gitignore	Merge issues/issue_42 into develop (#64)	2 weeks ago
LICENSE	Initial commit	3 months ago
README.md	Update README.md	3 months ago
summary_notes.md	Merge issues/issue_15 into develop	3 weeks ago

README.md

Scientific paper advisor

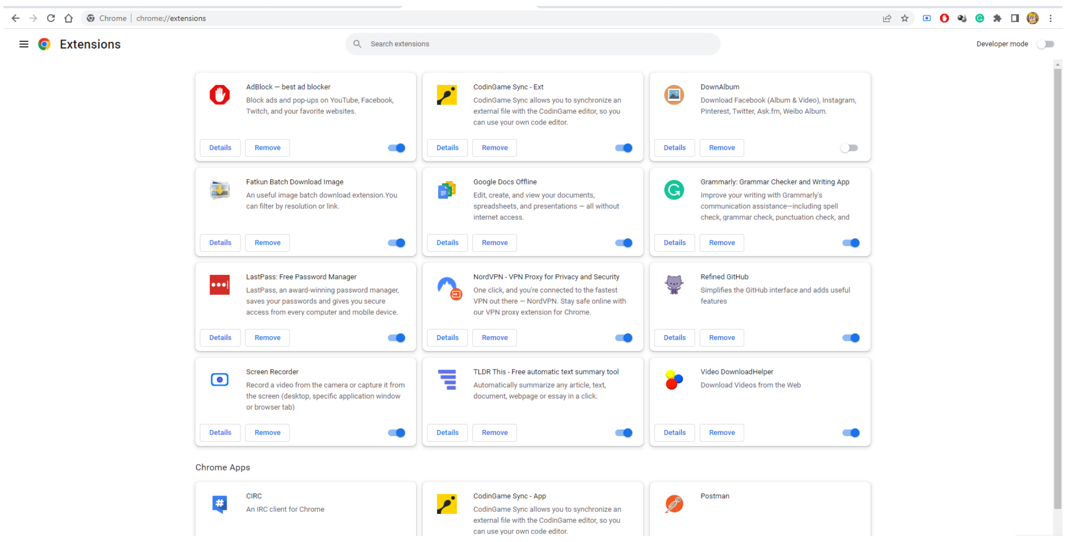
Google Chrome plugin for analyzing research papers and their links. Developed by Cezary Troska and Antoni Dąbrowski as an Engineering Thesis.

Figure 2.3: Scientific Paper Advisor GitHub repository

Open the repository, go to the */extension* directory, make any modifications you want[4] and build the project with the `npx webpack` command.

Then the built extension can be added to Chrome. Go to the 'Extensions' page in your browser, turn on *developer mode*, select *Import unpacked* option, and navigate to the directory with the extension's manifest.

[4]In this tutorial we assume these modifications neither alter the deployment process nor break the extension.



Chrome | chrome://extensions

Extensions

Search extensions

Developer mode

AdBlock — best ad blocker
Block ads and pop-ups on YouTube, Facebook, Twitch, and your favorite websites.

CodinGame Sync - Ext
CodinGame Sync allows you to synchronize an external file with the CodinGame editor, so you can use your own code editor.

DownAlbum
Download Facebook (Album & Video), Instagram, Pinterest, Twitter, Ask.fm, Weibo Album.

Fatkun Batch Download Image
An useful image batch download extension.You can filter by resolution or link.

Google Docs Offline
Edit, create, and view your documents, spreadsheets, and presentations — all without internet access.

Grammarly: Grammar Checker and Writing App
Improve your writing with Grammarly's communication assistance—including spell check, grammar check, punctuation check, and

LastPass: Free Password Manager
LastPass, an award-winning password manager, saves your passwords and gives you secure access from every computer and mobile device.

NordVPN - VPN Proxy for Privacy and Security
One click, and you're connected to the fastest VPN out there — NordVPN. Stay safe online with our VPN proxy extension for Chrome.

Refined GitHub
Simplifies the GitHub interface and adds useful features

Screen Recorder
Record a video from the camera or capture it from the screen (desktop, specific application window or browser tab)

TLDR This - Free automatic text summary tool
Automatically summarize any article, text, document, webpage or essay in a click.

Video DownloadHelper
Download Videos from the Web

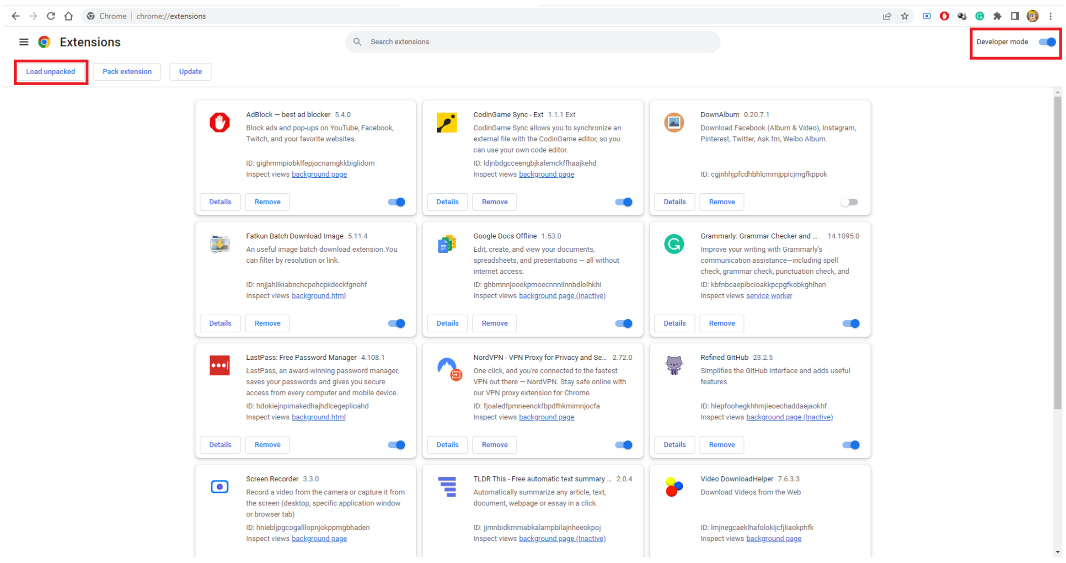
Chrome Apps

CIRC
An IRC client for Chrome

CodinGame Sync - App
CodinGame Sync allows you to synchronize an external file with the CodinGame editor, so you can use your own code editor.

Postman

Figure 2.4: The default state of the extension's browser



Chrome | chrome://extensions

Extensions

Search extensions

Developer mode

Load unpacked | Pack extension | Update

AdBlock — best ad blocker 5.4.0
Block ads and pop-ups on YouTube, Facebook, Twitch, and your favorite websites.
ID: gighmmpiobklfepjocnamgkkbiglidom
Inspect views background page

CodinGame Sync - Ext 1.1.1 Ext
CodinGame Sync allows you to synchronize an external file with the CodinGame editor, so you can use your own code editor.
ID: ldjnbdgcceengbjkalemckffhaajkehd
Inspect views background page

DownAlbum 0.20.7.1
Download Facebook (Album & Video), Instagram, Pinterest, Twitter, Ask.fm, Weibo Album.
ID: cgjnhhjpfcdhbhlcmmjppicjmgfkppok

Fatkun Batch Download Image 5.11.4
An useful image batch download extension.You can filter by resolution or link.
ID: nnjjahlikiabnchcpehcpkdeckfgnohf
Inspect views background.html

Google Docs Offline 1.53.0
Edit, create, and view your documents, spreadsheets, and presentations — all without internet access.
ID: ghbmnnjooekpmoecnnnilnnbdlolhkhi
Inspect views background page (Inactive)

Grammarly: Grammar Checker and ... 14.1095.0
Improve your writing with Grammarly's communication assistance—including spell check, grammar check, punctuation check, and
ID: kbfnbcaeplbcioakkpcpgfkobkghlhen
Inspect views service worker

LastPass: Free Password Manager 4.108.1
LastPass, an award-winning password manager, saves your passwords and gives you secure access from every computer and mobile device.
ID: hdokiejnpimakedhajhdlcegeplioahd
Inspect views background.html

NordVPN - VPN Proxy for Privacy and Se... 2.72.0
One click, and you're connected to the fastest VPN out there — NordVPN. Stay safe online with our VPN proxy extension for Chrome.
ID: fjoaledfpmneenckfbpdfhkmimnjocfa
Inspect views background page

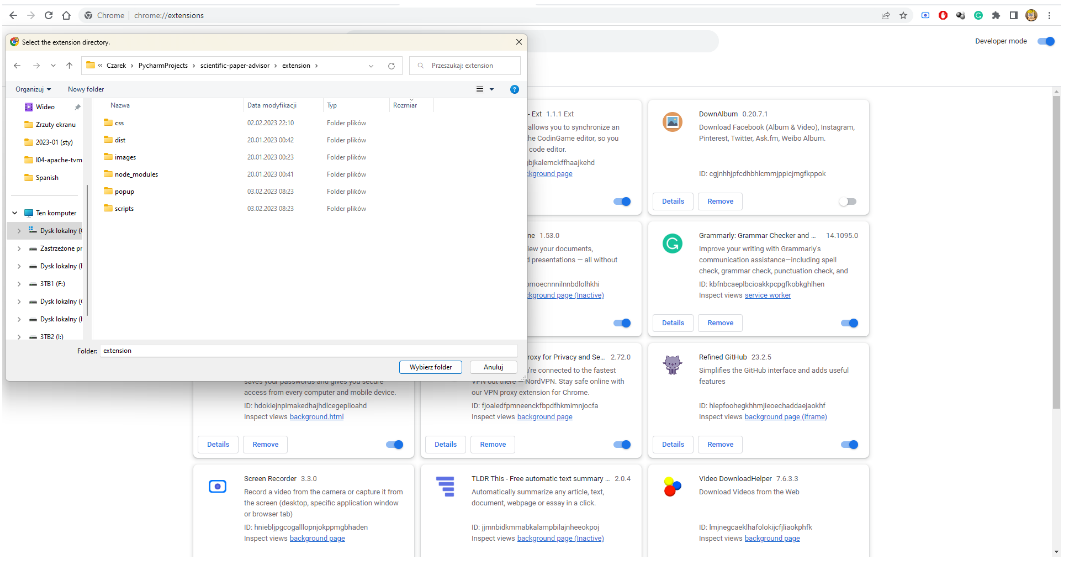
Refined GitHub 23.2.5
Simplifies the GitHub interface and adds useful features
ID: hlepfoohegkhhmjieoechaddaejaokhf
Inspect views background page (Inactive)

Screen Recorder 3.3.0
Record a video from the camera or capture it from the screen (desktop, specific application window or browser tab)
ID: hniebljpgcogalllopnjokppmgbhaden
Inspect views background page

TLDR This - Free automatic text summary ... 2.0.4
Automatically summarize any article, text, document, webpage or essay in a click.
ID: jjmnbidkmmabkalampbilajnheeokpoj
Inspect views background page (Inactive)

Video DownloadHelper 7.6.3.3
Download Videos from the Web
ID: lmjnegcaeklhafolokijcfjliaokphfk
Inspect views background page

Figure 2.5: The options mentioned in the tutorial are marked with red rectangles.


Chrome | chrome://extensions

Developer mode

Select the extension directory.

« Czarek › PycharmProjects › scientific-paper-advisor › extension ›

Przeszukaj: extension

Organizuj ▾ Nowy folder

Nazwa	Data modyfikacji	Typ	Rozmiar
css	02.02.2023 22:10	Folder plików	
dist	20.01.2023 00:42	Folder plików	
images	20.01.2023 00:23	Folder plików	
node_modules	20.01.2023 00:41	Folder plików	
popup	03.02.2023 08:23	Folder plików	
scripts	03.02.2023 08:23	Folder plików	

Wideo, Zrzuty ekranu, 2023-01 (sty), I04-apache-tvm, Spanish, Ten komputer, Dysk lokalny (, Zastrzeżone pr, Dysk lokalny (I, 3TB1 (F:), Dysk lokalny (, Dysk lokalny (I, 3TB2 (I:)

Folder: extension

Wybierz folder   Anuluj

DownAlbum 0.20.7.1
Download Facebook (Album & Video), Instagram, Pinterest, Twitter, Ask.fm, Weibo Album.
ID: cgjnhhjpfcdhbhlcmmjppicjmgfkppok

Grammarly: Grammar Checker and ... 14.1095.0
Improve your writing with Grammarly's communication assistance—including spell check, grammar check, punctuation check, and
ID: kbfnbcaeplbcioakkpcpgfkobkghlhen
Inspect views service worker

Refined GitHub 23.2.5
Simplifies the GitHub interface and adds useful features
ID: hlepfoohegkhhmjieoechaddaejaokhf
Inspect views background page (iframe)

access from every computer and mobile device.
ID: hdokiejnpimakedhajhdlcegeplioahd
Inspect views background.html

roxy for Privacy and Se... 2.72.0
NordVPN. Stay safe online with our VPN proxy extension for Chrome.
ID: fjoaledfpmneenckfbpdfhkmimnjocfa
Inspect views background page

Screen Recorder 3.3.0
Record a video from the camera or capture it from the screen (desktop, specific application window or browser tab)
ID: hniebljpgcogalllopnjokppmgbhaden
Inspect views background page

TLDR This - Free automatic text summary ... 2.0.4
Automatically summarize any article, text, document, webpage or essay in a click.
ID: jjmnbidkmmabkalampbilajnheeokpoj
Inspect views background page (Inactive)

Video DownloadHelper 7.6.3.3
Download Videos from the Web
ID: lmjnegcaeklhafolokijcfjliaokphfk
Inspect views background page

Figure 2.6: System explorer should be directed to the directory with the extension's manifest.

2.2.2 Custom backend – optional step

If you want to host your own backend server for this extension, our repository is prepared for that.

The recommended way of setting up a custom backend is using the docker-compose with the *stack.yml* file. Before you run your setup you need to fill in the configuration. The only required part is the `SCRAPERAPI_KEY` variable from the *.env* file – you need to put your ScraperAPI account's key in there to use proxy servers[5][6]. When your configuration is complete, you can run create the container setup with the command `docker compose -f stack.yml up` run in the */backend* directory.

When the building process is done your server shall be listening on the *127.0.0.1:8000* address. You need to switch your extension to use that URL instead of the default server for SPA. You can do that by clicking on the extension's icon in the upper right corner of your browser and selecting the 'local' and 'http' options from the popup menu.

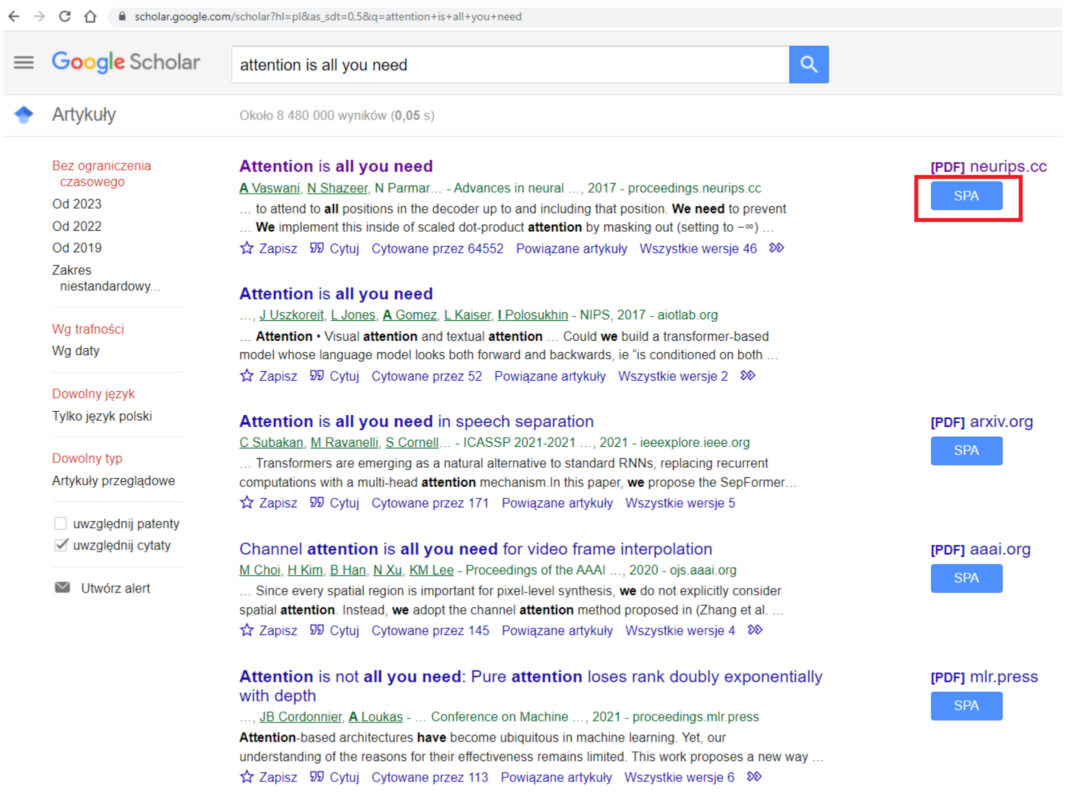
2.2.3 Example of usage

Let's use our extension on the well-known paper 'Attention is All you Need'[6]. Once the user searches for it on Google Scholar the results page will contain a 'SPA'

[5]ScraperAPI is the only proxy service we support at the moment. If that changes in the future, the setup options will be updated in the repository's README

[6]Proxy service is required. Scholar will quickly block your backend without it for bot-like behavior.

button. By clicking on it, the extension is notified the user is interested in learning more about the article and a graph will be displayed.


scholar.google.com/scholar?hl=pl&as_sdt=0,5&q=attention+is+all+you+need

Google Scholar | attention is all you need

Artykuły | Około 8 480 000 wyników (**0,05** s)

Bez ograniczenia czasowego
Od 2023
Od 2022
Od 2019
Zakres niestandardowy...

Wg trafności
Wg daty

Dowolny język
Tylko język polski

Dowolny typ
Artykuły przeglądowe

uwzględnij patenty
✓ uwzględnij cytaty

Utwórz alert

Attention is **all you need** — [PDF] neurips.cc — SPA
A Vaswani, N Shazeer, N Parmar... - Advances in neural ..., 2017 - proceedings.neurips.cc
... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent ... **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) ...
Zapisz Cytuj Cytowane przez 64552 Powiązane artykuły Wszystkie wersje 46

Attention is **all you need**
..., J Uszkoreit, L Jones, A Gomez, L Kaiser, I Polosukhin - NIPS, 2017 - aiotlab.org
... **Attention** • Visual **attention** and textual **attention** ... Could **we** build a transformer-based model whose language model looks both forward and backwards, ie "is conditioned on both ...
Zapisz Cytuj Cytowane przez 52 Powiązane artykuły Wszystkie wersje 2

Attention is **all you need** in speech separation — [PDF] arxiv.org — SPA
C Subakan, M Ravanelli, S Cornell... - ICASSP 2021-2021 ..., 2021 - ieeexplore.ieee.org
... Transformers are emerging as a natural alternative to standard RNNs, replacing recurrent computations with a multi-head **attention** mechanism.In this paper, **we** propose the SepFormer...
Zapisz Cytuj Cytowane przez 171 Powiązane artykuły Wszystkie wersje 5

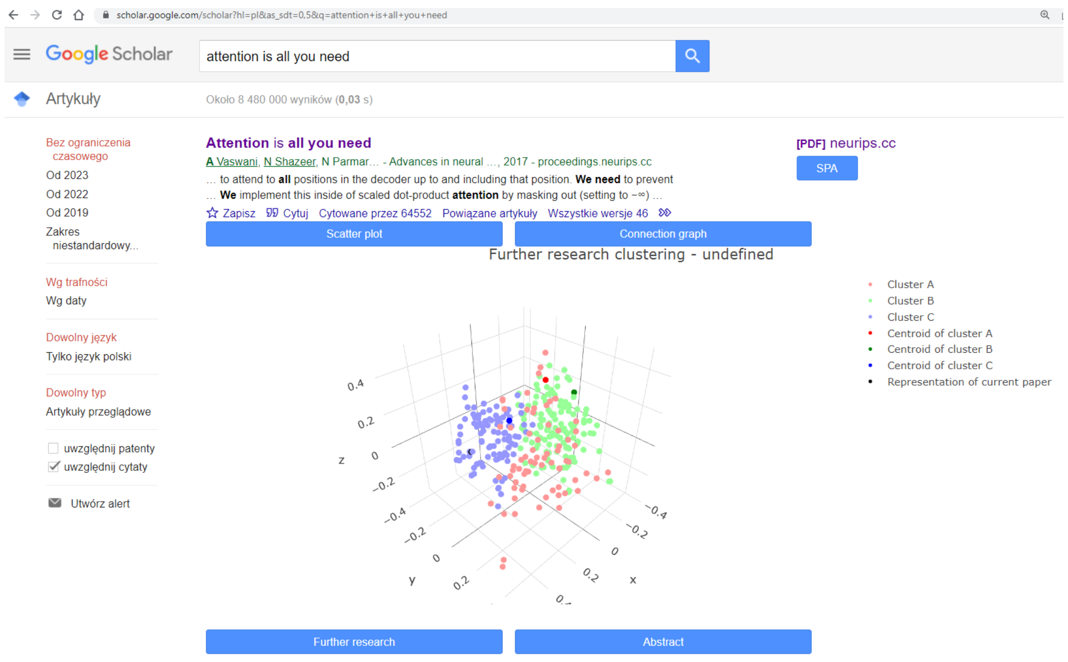
Channel **attention** is **all you need** for video frame interpolation — [PDF] aaai.org — SPA
M Choi, H Kim, B Han, N Xu, KM Lee - Proceedings of the AAAI ..., 2020 - ojs.aaai.org
... Since every spatial region is important for pixel-level synthesis, **we** do not explicitly consider spatial **attention**. Instead, **we** adopt the channel **attention** method proposed in (Zhang et al. ...
Zapisz Cytuj Cytowane przez 145 Powiązane artykuły Wszystkie wersje 4

Attention is not **all you need**: Pure **attention** loses rank doubly exponentially with depth — [PDF] mlr.press — SPA
..., JB Cordonnier, A Loukas - ... Conference on Machine ..., 2021 - proceedings.mlr.press
Attention-based architectures **have** become ubiquitous in machine learning. Yet, our understanding of the reasons for their effectiveness remains limited. This work proposes a new way ...
Zapisz Cytuj Cytowane przez 113 Powiązane artykuły Wszystkie wersje 6

Figure 2.7: Extension will add new elements to Google Scholar result pages. The marked button will start the extension's interaction with the article.

This graph contains information about a collection of suggestions for further research from recently publicized articles in the same scientific category as 'Attention is All you Need'.

The graph nodes are divided into three clusters, color-coded accordingly. Clusters are created based on the articles' similarity and for each of those clusters a representative is chosen. Details of the clusterization procedure for further research suggestions and the papers' abstracts are described in Section 5.2.

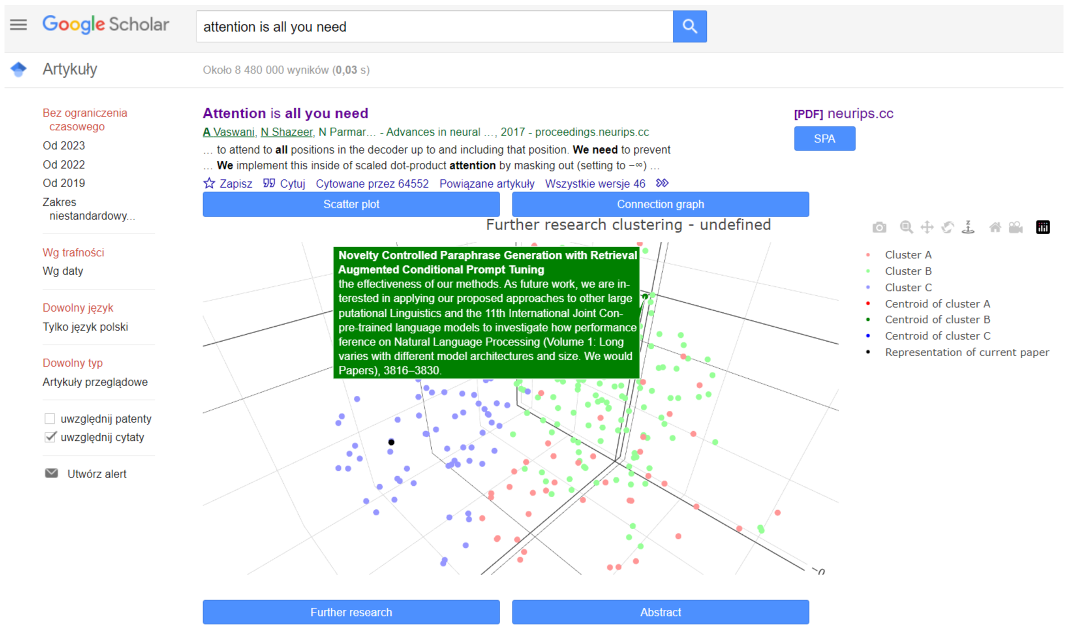


Attention is all you need

A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

… to attend to all positions in the decoder up to and including that position. We need to prevent … We implement this inside of scaled dot-product attention by masking out (setting to −∞) …

Figure 2.8: The first graph our extension return is a graph with further research suggestions from the scientific category.



Attention is all you need

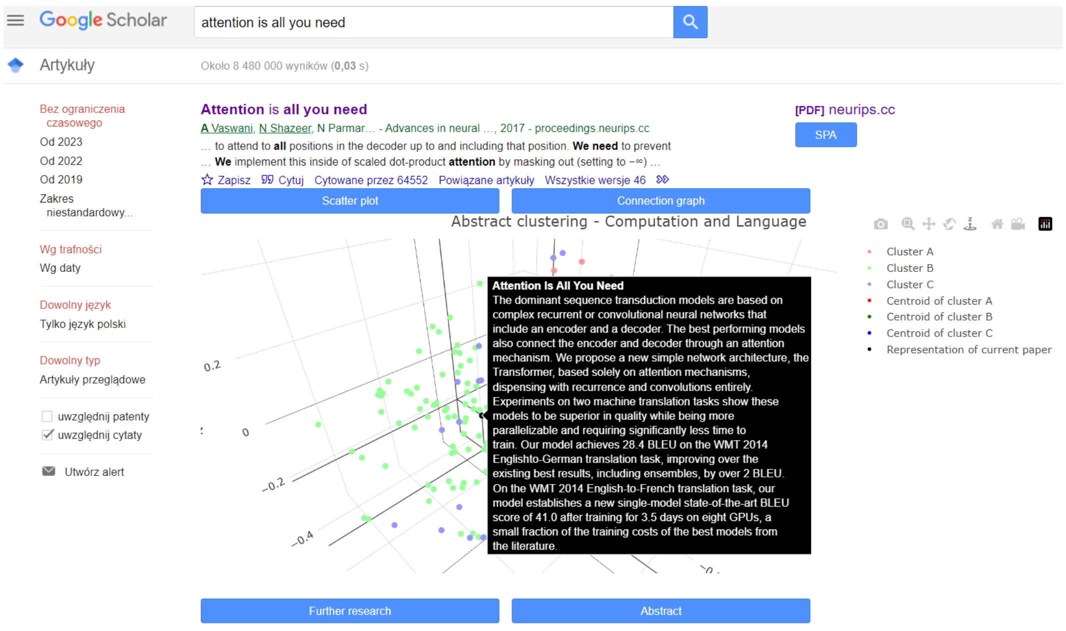
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

… to attend to all positions in the decoder up to and including that position. We need to prevent … We implement this inside of scaled dot-product attention by masking out (setting to −∞) …

Novelty Controlled Paraphrase Generation with Retrieval Augmented Conditional Prompt Tuning
the effectiveness of our methods. As future work, we are interested in applying our proposed approaches to other large putational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long pre-trained language models to investigate how performance varies with different model architectures and size. We would Papers), 3816–3830.

Figure 2.9: An example of suggestion discovered by the extension in a paper.

If the user wants to shift the focus of the graph they can use the 'Abstract' button. This will recreate the graph, but this time the data shown and the clusterization performed will focus on the abstracts of the publications.


Attention is all you need

A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

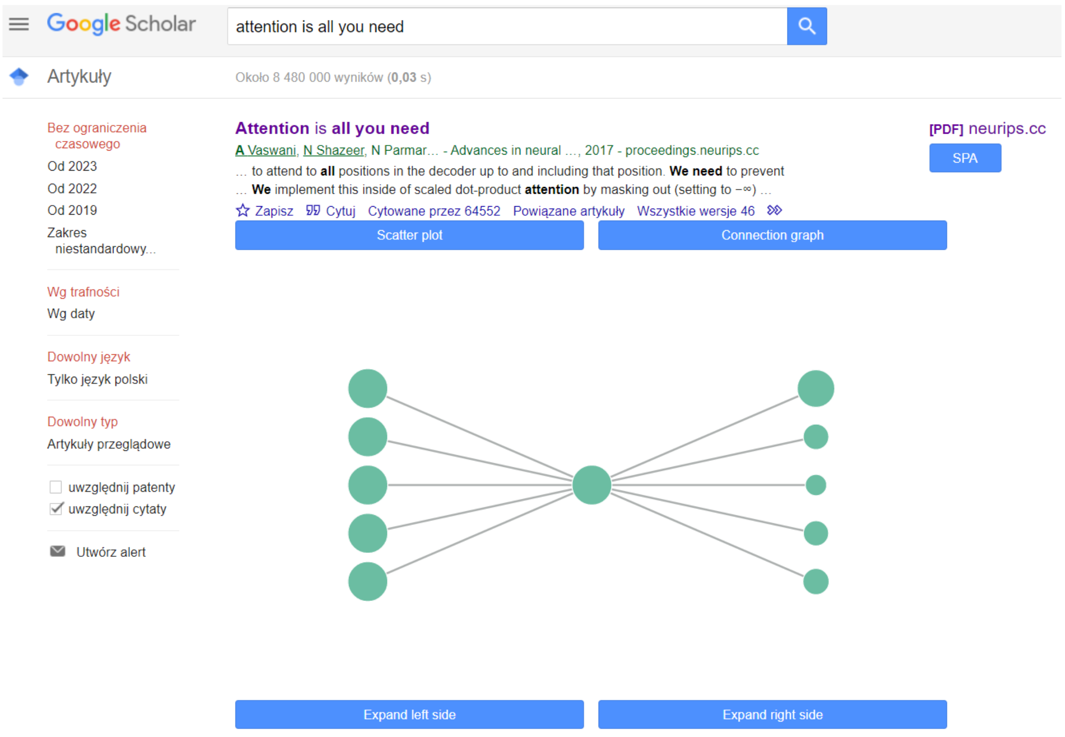
… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
… **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

Attention Is All You Need
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 Englishto-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Figure 2.10: The graph can represent abstracts instead. The main article's abstract will be included in the representation.

The other type of graph we offer is specific to the selected article, not the whole category. It shows the user the influence of the chosen article on the scientific landscape. The user can see the references this article uses and publications that cite it. Those relations are represented by the edges of the graph – an article on the right end of the edge cites connected publication on the left side.

Additional information is coded in the size of the nodes. The diameter of the node represents the number of citations of the corresponding paper. The size of the node grows logarithmically and reaches a maximum at 1100 citations.


Google Scholar

attention is all you need

Artykuły

Około 8 480 000 wyników (**0,03** s)

Bez ograniczenia czasowego
Od 2023
Od 2022
Od 2019
Zakres niestandardowy...

Wg trafności
Wg daty

Dowolny język
Tylko język polski

Dowolny typ
Artykuły przeglądowe

☐ uwzględnij patenty
☑ uwzględnij cytaty

Utwórz alert

Attention is **all you need**

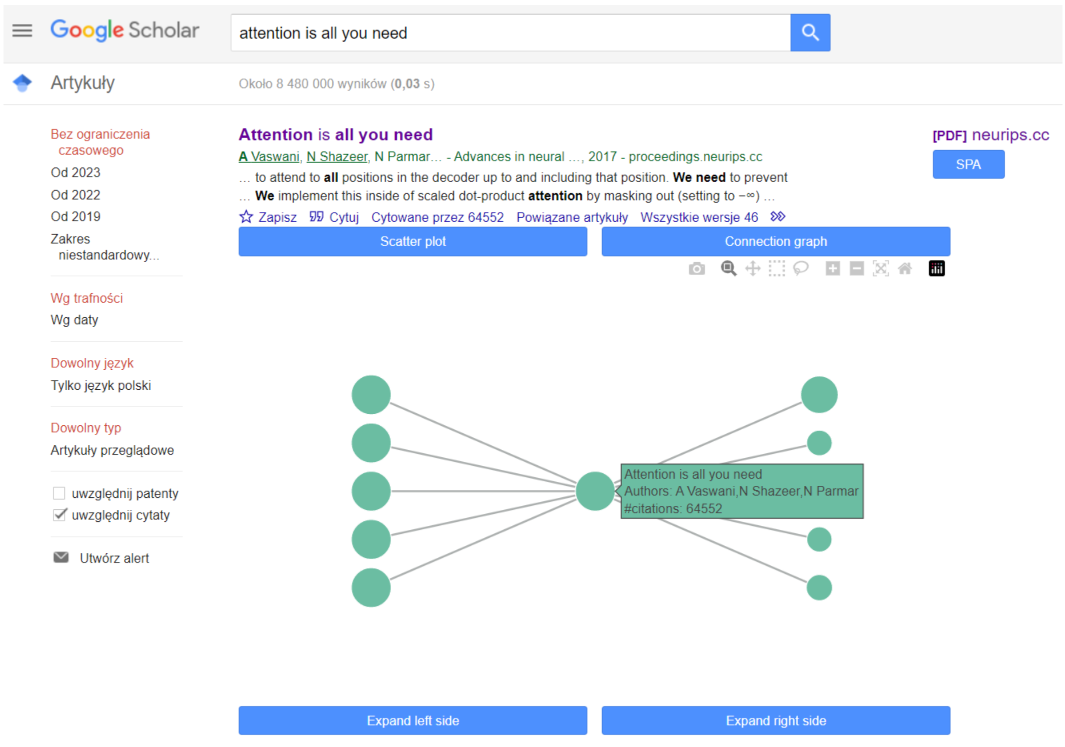
A Vaswani, N Shazeer, N Parmar... - Advances in neural ..., 2017 - proceedings.neurips.cc

... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
... **We** implement this inside of scaled dot-product **attention** by masking out (setting to $-\infty$) ...

☆ Zapisz   Cytuj   Cytowane przez 64552   Powiązane artykuły   Wszystkie wersje 46

[PDF] neurips.cc

Figure 2.11: Connection graph is another mode in which the extension can operate.



attention is all you need

Około 8 480 000 wyników (**0,03** s)

Bez ograniczenia czasowego
Od 2023
Od 2022
Od 2019
Zakres niestandardowy...

Wg trafności
Wg daty

Dowolny język
Tylko język polski

Dowolny typ
Artykuły przeglądowe

☐ uwzględnij patenty
☑ uwzględnij cytaty

Utwórz alert

Attention is **all you need**

A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

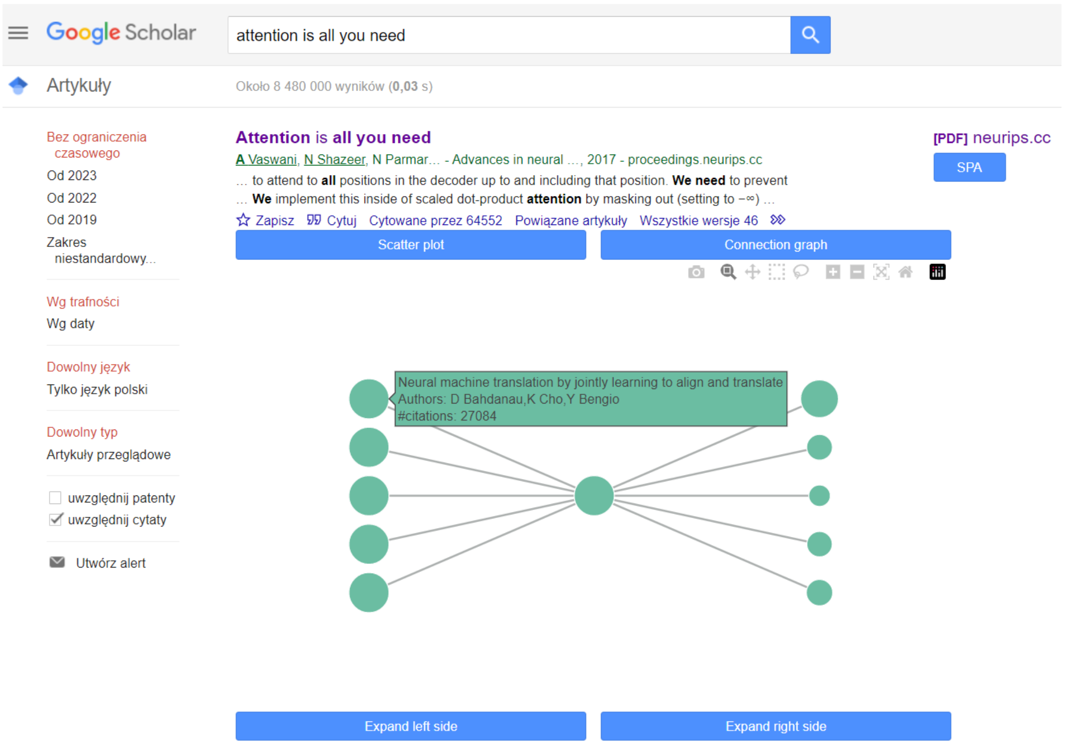
… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
… **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

☆ Zapisz 99 Cytuj Cytowane przez 64552 Powiązane artykuły Wszystkie wersje 46

[PDF] neurips.cc

Attention is all you need
Authors: A Vaswani,N Shazeer,N Parmar
#citations: 64552

Figure 2.12: The graph representation visualizes connections between articles with a special focus on the chosen article. It is put in the center of the graph, in its own column.



attention is all you need

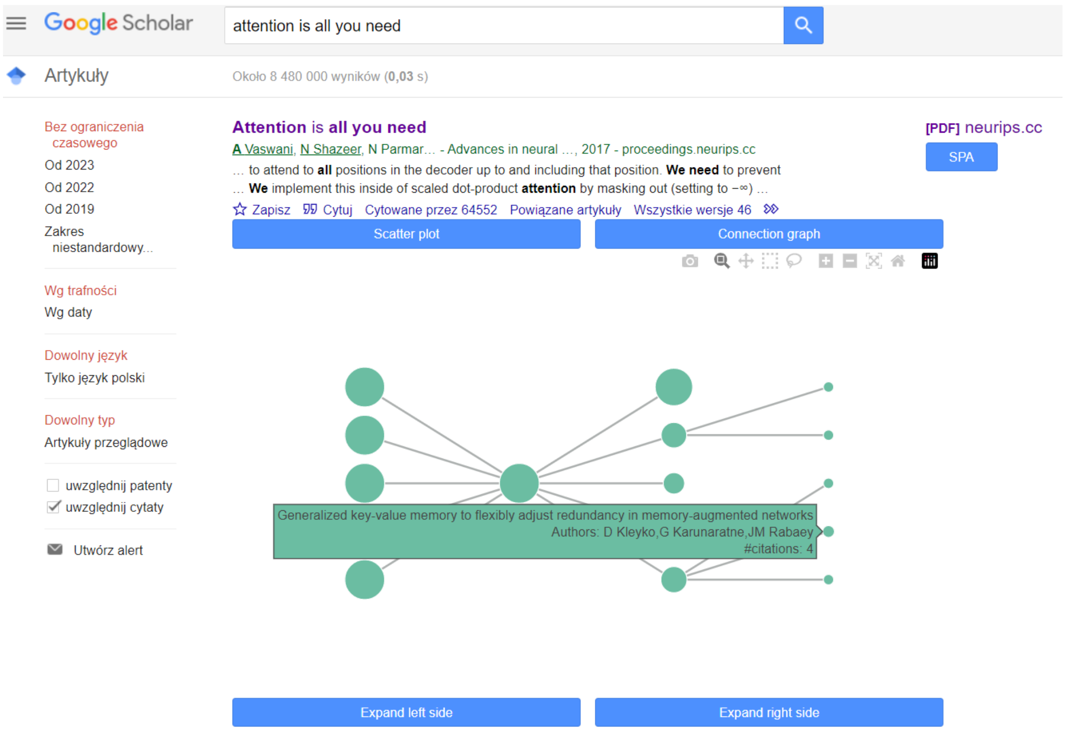
Około 8 480 000 wyników (**0,03** s)

Attention is all you need
A Vaswani, N Shazeer, N Parmar... - Advances in neural ..., 2017 - proceedings.neurips.cc
... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
... **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) ...

Neural machine translation by jointly learning to align and translate
Authors: D Bahdanau,K Cho,Y Bengio
#citations: 27084

Figure 2.13: Publications referenced by the main article are put on the left. Papers that cite the main article are on the right

If the user wishes to, they can expand the graph in either direction. Clicking on the 'Expand left side' will give the user a new column on the left with secondary references to the article, and 'Expand right side' will give a new column on the right with publications citing the publications from the previous rightmost column.


Google Scholar

attention is all you need

Artykuły

Około 8 480 000 wyników (**0,03** s)

Bez ograniczenia czasowego
Od 2023
Od 2022
Od 2019
Zakres niestandardowy...

Wg trafności
Wg daty

Dowolny język
Tylko język polski

Dowolny typ
Artykuły przeglądowe

☐ uwzględnij patenty
☑ uwzględnij cytaty

Utwórz alert

Attention is **all you need**
A Vaswani, N Shazeer, N Parmar... - Advances in neural ..., 2017 - proceedings.neurips.cc
... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
... **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) ...
☆ Zapisz 99 Cytuj Cytowane przez 64552 Powiązane artykuły Wszystkie wersje 46

[PDF] neurips.cc

SPA

Scatter plot

Connection graph

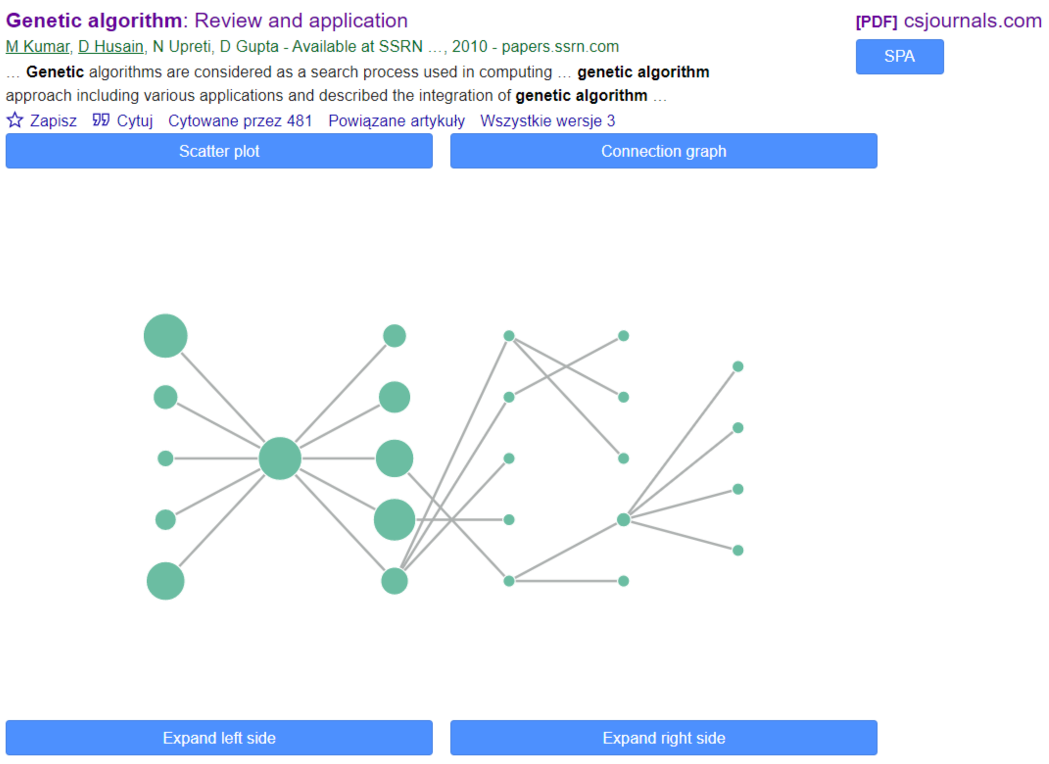
Generalized key-value memory to flexibly adjust redundancy in memory-augmented networks
Authors: D Kleyko,G Karunaratne,JM Rabaey
#citations: 4

Expand left side

Expand right side

Figure 2.14: The user can use the 'Expand right side' and 'Expand left side' buttons to add new columns on the corresponding sides. This example shows a new column on the right with articles that cite the publications with which they share an edge.

The user can just keep going with expansions as long as there are articles that meet the requirements of the column.

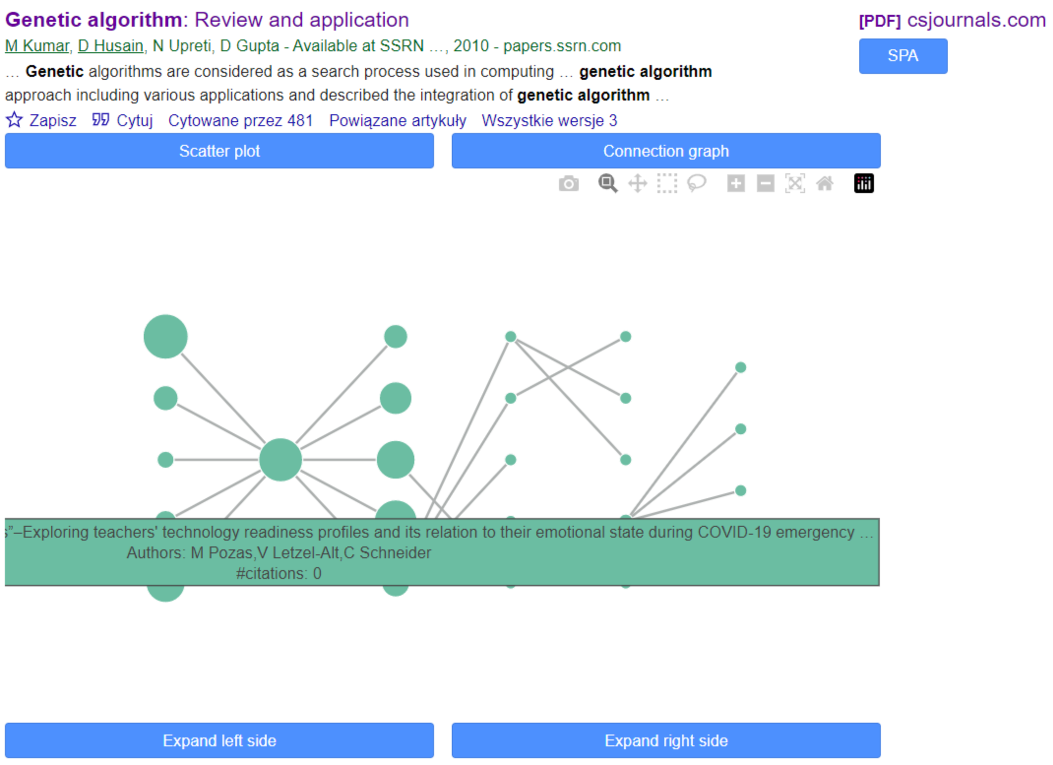


Genetic algorithm: Review and application

M Kumar, D Husain, N Upreti, D Gupta - Available at SSRN …, 2010 - papers.ssrn.com

… Genetic algorithms are considered as a search process used in computing … genetic algorithm approach including various applications and described the integration of genetic algorithm …

Figure 2.15: Graph can be extended multiple times, as long as there are articles to be used for the next column. This figure represents an article where after three extensions we reached the limit.



Genetic algorithm: Review and application

M Kumar, D Husain, N Upreti, D Gupta - Available at SSRN ..., 2010 - papers.ssrn.com

... **Genetic** algorithms are considered as a search process used in computing ... **genetic algorithm** approach including various applications and described the integration of **genetic algorithm** ...

Zapisz Cytuj Cytowane przez 481 Powiązane artykuły Wszystkie wersje 3

[PDF] csjournals.com

;"–Exploring teachers' technology readiness profiles and its relation to their emotional state during COVID-19 emergency ...
Authors: M Pozas,V Letzel-Alt,C Schneider
#citations: 0

Figure 2.16: Additional extensions won't yield any results, because the rightmost column contains only articles with no citations. It also contains only four articles, because the previous column had just one article with any citations, precisely four

Chapter 3

Implementation details

The goal of this chapter is to describe the development process undertaken during this project. The form we chose, i.e. a browser extension, comes with a significant amount of limitations and specificity. The description of methods and tools utilized while working on this task might prove useful for developers facing similar tasks in the future.

3.1 Data gathering and web scraping

The central feature of our solution is to present the data about scientific papers and their respective fields. Selecting the most pressing research suggestions and estimating the popularity potential of a paper can be done but it requires extensive data analysis.

Our approach assumes collecting three sets of different information: the details of an article chosen by the user, including references and a list of available papers citing the chosen article, the suggestions for further research presented in the papers from recent years, and a large corpus of articles with citation statistics.

3.1.1 Google Scholar

One of the goals set for this extension is for it to work with any article Google Scholar stores in form of a PDF file. It would be impossible to have all of the data collected in advance, so the application needs to be able to gather this information dynamically.

When the user requests a connection graph for a given article, the frontend extracts the title and list of authors from the Google Scholar search result. That information is sent to the backend and the server tries to find the article details. The Advanced Search option was helpful in finding the user's chosen article with the least amount of queries.

Scholar does not have an official public API so we decided to use an unofficial one called scholarly. This python library helped us greatly with handling communication with Google Scholar. It passes our queries to the browser and parses the response into a list of Publication objects. Thanks to that functionality we do not have to attempt parsing the Google Scholar output by ourselves. Those Publications objects contain all the information we need for our functionalities, including the number of citations and abstract. On top of that scholarly provides us with a *citedby* function that when given a Publication object returns a list of papers that reference the corresponding print.

A piece of information which turned out to be especially useful in our endeavor was a list of articles citing the requested paper. This data allows us to create the right side of the connection graph, the side representing the articles that stem from the chosen one.

The backward snowballing process required a different approach. Neither Google nor any other public system shares records of articles' references that we can use for arbitrary articles in real-time. Thus, the solution we decided to go with was extracting that information from the article text itself. Frontend includes the PDF link to the article in the request sent to the backend. This link is used for downloading the PDF content and passing it to the parsing model. We found a service designed for this exact case –Science-parse-api. This tool extracts the bibliography from the given PDF and parses it into a list of references. We can use that list to collect the titles and authors of referenced articles, which in turn allows us to find those publications with scholarly and make them into nodes in the connection graph.

To create additional columns in the graph2.14 the whole process was repeated for each of the representatives from the previous layer. Each elaboration effort is carried out in a separate processing thread. In the end, the results are gathered and a constant-size-subset is chosen, based on the number of citations of the found articles (or the prediction of the number of citations, in the case of more recent articles).

A significant obstacle we encountered in this step was Google Scholar's scrapping prevention mechanisms. Bot-like demeanor quickly leads to verification requests in form of CAPTCHA tests. To work around this we utilized proxy servers masking the patterns of the automated behavior. To attain those proxies we utilized ScraperAPI. They handled solving CAPTCHA and rotating proxies. An additional cost of this solution was a hit to performance. Requests going through ScrapperAPI's servers have a delay of up to 14 seconds, a time the service needs to work around the Scholar's safeguards[1]. This is the sole reason for the noticeable waiting time the

[1]The extension is now officially distributed in Chrome's Web Store and passed the Google screening process, so we assume the company is fine with our current approach. We will be lobbying for a release of an official API for Google Scholar which could be used in such tools without the need

user might experience when requesting a connection graph for an article the backend hasn't worked on before.

3.1.2 Further research database

The goal of this database was to gather and analyze all of the suggestions for research projects given by scientists in recently released papers. We defined those proposals as sentences containing specific phrases: *further research*, *further study*, *future work*, *additional research*, *further analysis*, *further examination*, *additional investigation* or *additional studies*. Those suggestions will be sorted according to arXiv's category taxonomy and for each category, representatives will be selected.

The first step to do was to collect the texts for processing. We tried doing that with an official arXiv API but the limitations on the number of requests established in the API's terms of service made this process slow and inefficient. We found an alternative in the form of a Kaggle database kept up to date by arXiv. This is one of their suggested methods of bulk access. With this resource, we downloaded 250.000 papers released in the year 2022 from various categories.

Exact information to which category the paper belongs was extracted with the help of the API. This information was useful for two reasons – we were interested in sorting the research suggestions by this metric but we needed to create a model that would assign the categories to papers not stored in arXiv.

The final result was a database containing further research requests we could use in our feature recommending topics for new studies.

3.1.3 Article popularity database

Another feature offered by our browser extension is predicting the future popularity of recently added articles. To do that we needed to collect a database of articles with citation statistics from a chosen timeframe. Besides the information about the number of citations, we wanted to collect data like the paper's authors and their affiliations, where the research was released, the abstract, and the category of the research.

All of this information is offered by Google Scholar, but the aforementioned long request processing times imposed by the proxy servers and the limited number of uses our ScraperAPI package is allowed made this approach undesirable for collecting a large set of data.

We requested access to some proprietary databases. This effort resulted in obtaining an API key for Semantic Scholar. This organization provides a vast selection of information about international academia and shares it with researchers, with a

for a proxy server as a workaround.

special focus on AI development. This resource proved invaluable in developing the popularity prediction model.

Semantic Scholar's data collection can be accessed in two ways: with an official RESTful web API and an unofficial Python client. Initially, we hoped the Python client would be sufficient for all our needs but during the development process it turned out to be unreliable with some types of data, like details about the authors. We filled in that blank by communicating directly with the web API via Python's requests library.

3.2 Frontend

Scientific Paper Advisor's frontend acts as an interface between the user and a backend server. Its job is to integrate all of our functionalities into Google Scholar in an intuitive and non-invasive way.

3.2.1 Browser extension

Our client was developed as a browser extension. We decided to use this option to make the installation process as seamless as possible for the users - anyone who wants to use our tool just needs to find it in the Chrome Web Store (images **??**) and they are ready to go.

The object defining an extension is its manifest. This file defines when the functionalities will be allowed to run and lists all the files the extension consists of. Our tool gets activated only when the user enters pages with Google Scholar results, so there are no situations where we would waste the computer's resources on processing pages without scientific papers.

Figure 3.1: The icon representing Scientific Paper Advisor

The process of adding an extension to the Chrome Web Store required four steps. Firstly, we had to create Google Developer Account. Then we were able to upload the extension code. The initial validation requires adding descriptions and promotion materials. In this process, we were also asked to explain all the space and access requirements that our extension needs. After adding a lengthy description the extension was sent for verification. After a few days it was available publically in the store. Each new update required repeating those steps.

3.2.2 Layout

After installing our application users will notice new elements added to the page with Google Scholar search results.

Interaction with our features starts when the user clicks on a 'SPA' button found right under a link to the paper's PDF. Clicking on it is a signal to the extension that the user wants to investigate the connections this paper has in its field of study.

Clicking the button will cause a new element to show up, a menu presenting graphical representations of the data gathered about the article. The user can use the buttons on top of the menu to switch between two graphs: the connection graph and the further research graph. The further research graph is the one chosen by default, so it will show up first when the menu is displayed.

When the user switches to the connections graph two more buttons will become available. They are located below the graph and labeled as 'Backward snowballing' and 'Further citations'. They will cause the expansion of the graph on the left side or the right side, respectively. The details behind this process were described in Section 3.1.1 of this paper.

Collecting all of the information, especially for the connection graph, takes some time. The extension will signal to the user that it is in the process of collecting data with a spinner animation. This animation will be automatically replaced by the graph once the graph is ready. This is to let the user know that the extension is working on their request and did not just crash.

3.2.3 Backend communication

Collecting the information for the construction of either of the graphs is not a task for the frontend. Execution times would be too dependent on the client's hardware, processing of the data would be unhandy from the developers' perspective and we would have to share sensitive information with the client, such as API keys to our proxy server.

Because of it, the forntend is tasked with gathering basic information available locally on the loaded page, without additional requests. This information includes the title of the user's selected article, a list of authors, and a link to the paper's PDF. This data is then sent to the backend. The backend's side of the processing is described in Section 3.3.1.

The backend responds with a JSON representation of the graph. It contains all the information frontend needs for a graphical representation of the requested aspect.

Further communication with the backend will take place if the user decides to use the 'Backward snowballing' or 'Further citations' buttons. In this case, the frontend sends the schema of the currently rendered graph so that the backend does not have to create it from the very beginning but just needs to elaborate on the current state. The response is a JSON with the updated graph.

3.2.4 Results visualization

We use Plotly.js library for creating the graphs from the JSON files received from the backend.

An important, extension-specific detail we had to keep in mind is that we are restricted to using 'strict' versions of the packages.

Importing modules in the content script (the main script of the extension) is heavily restricted, so to be able to use the Plotly object we had to find a workaround. The solution came in form of the webpack module bundler. It allowed us to convert all the resources used by our content script into one static asset, the state matching Chrome's extension standards.

The diameter of the markers reflects the popularity of the article it represents. Users might click on them to open the webpage with the requested publication.

3.3 Backend

The backend part of our application is responsible for finding, collecting and storing the data about the articles requested by the user.

3.3.1 Django-based application

The backend of the extension was created using Django[11] framework. We decided to use it based on prior familiarity, support for our chosen database management system, and the detailed documentation it provides. Our Django project has two functional parts ('applications' in this framework's terminology), each responsible for one type of graph we provide.

During the development we relied on the lightweight server provided by Django as for the production environment we decided to go with the Gunicorn.

articlegraph application

The application responsible for the connection graph provides three endpoints corresponding to three methods of expanding graph information.

- \articlegraph is the endpoint to which the frontend sends a request for the base of the graph, which consists of the article chosen by the user, its direct references, and works citing the central publication. The request contains the article's title, list of the authors, and the link to the PDF.
- \articlegraph\expandleft - this endpoint lets the frontend add the next column

of references to the left side of the graph. This column will contain publications used in the previous rightmost column. The request contains the JSON representation of the current state of the graph.

- \articlegraph\expandright - similar to the \articlegraph\expandleft endpoint. The frontend sends here the JSON representation of the current state of the graph. The response is the JSON with a new row to the right that contains articles citing the articles from the previous leftmost column.

The details of the article are found based on the title and list of authors. We create a query that is run in Google Scholar with the use of the scholarly. This yields a Publication object that contains details about the article. The scholarly lets us use this Publication object to get a list of citations of the represented paper and one of the details that object contains is the link which can be used to download the contents of the article. The list of citations is sorted by popularity and truncated to a constant maximum size. The list of references is extracted from the article's PDF with science-parse-api library, then they are converted into the Publication objects.

When one extends the graph, the operations described above are applied to each of the articles from the previous column. The results are concatenated, sorted by popularity and truncated.

The edge in the graph represents the citation relationship. The article on the right side of the edge cites the publication on the left.

To keep track of the edges and create the layout for the visual representation of the graph, we utilized NetworkX library. It provides a function *multipartite_layout* that returns coordinates which frontend's Plotly.js will use to create a graph with evenly spaced nodes and columns.

frquesions application

The application connected with a display of the 'further research' plot has only one endpoint \frquestions. It receives the URL of the currently analyzed article and passes it to the backend. The server then sends a request to Google Scholar for a full publication PDF, which is later parsed, analyzed and sent to frontend with other articles from the same domain.

3.3.2 PostgreSQL database

Gathering information by scrapping the web is a time-consuming process. Once the information about the article is collected, it is stored in the database to speed up the resolution of user requests. This allows us to retrieve it locally in case it is needed in the future.

Data is stored in a PostgreSQL database, running in the Docker container (more on the usage of Docker in this project in section 3.3.3). Django supports this database management system, so we were able to use its Models approach to integrate the databases into the project.

We created Models representing an article, a citation relation between pieces, and a result of processing a PDF for the further research section. The getter functions we constructed for articles or further research results first check for the objects in the database before trying to get them by other means. If the objects are not present in the PostgreSQL, at the end of the getter function, we save them.

Articles are saved in the byte stream format with a help of the pickle library. This allows us to save and load the objects from the database while keeping the properties of the objects storing the data. We use it to always keep the information about an article as an instance of a Publication class. The process of using byte streams with the database is easy thanks to the PickledObjectField, table column type added with the django-picklefield library.

3.3.3 Dockerization

Scientific Paper Advisor's backend uses Docker[3] to quickly, and regardless of hardware, deploy services it depends on. The two services in question are the PostgreSQL database and the Science Parse server.

Science Parse is a service allowing to swiftly analyze scientific papers' PDFs. It handles extracting parts important for our application — title, authors, abstract and list of references. It uses a model trained specifically for handling scientific papers, outperforming other PDF text extractors. However, for proper functioning, it requires its own server. The recommended method of deploying that server is with a docker container with a port published for communication. Due to the fact that we were already considering Docker for handling our database needs, this additional container could be added without much additional development work.

Having a dockerized database is beneficial for many reasons, especially during development. Using Docker containers guarantees configuration consistency across the developers, so we can be certain that the mechanisms we create will work regardless of the deployment site. It allows us to quickly reinitialize the whole database in case we want to start from a clean slate. It is also secure – we can isolate the container so it is only accessible from the machine running the server and is not exposed to any outside ports.

Putting those two services in the containers persuaded us to go one step further and put the main application server in a container as well. We added a Dockerfile that creates the environment required for the server to run and exposes it, so it can have its ports mapped to the host and work as if it was deployed on a standard machine. The main goal guiding us in this endeavor was to make the deployment process as easy as possible. Without the dockerization anyone who would want to run our server would have to provide a properly configured database, Python with all the requirements, and on top of that they would have to install Docker anyway because it is required for the Science Parse component. This process is counterproductive if the goal of the project is supposed to gain any traction and encourage other developers to use and contribute to the project. Such an approach would only overwhelm potential contributors with all the requirements.

The current docker setup reduces the deployment to filling in the configuration and running a single command. With `docker compose -f stack.yml up` docker will prepare a collection of containers with a main one running our server and listening for any requests from the frontend.

3.3.4 Google Cloud deployment

We want our extension to be an out-of-the-box solution. The first step to achieving this was getting it available in the Web Store, but the other necessary component is a central, publicly available server that anyone not wanting to run their own backend instance can use.

To achieve this we decided to use the services of Google Cloud. This cloud platform allowed us to create a virtual machine and assign an external, static IP that can be reached by any instance of our frontend. This is where the dockerization turned out useful again – deployment of our backend was this freshly created machine, which was swift and painless. We just had to install the docker-compose from its repository, clone the source code from the repository and order the docker to initialize containers. We had the backend working in no time and updating it after the code changes was equally straightforward.

The only new hurdle posed by this remote backend was securing the connection. Google accepted insecure connections to the localhost but for the external IP https was strictly required. Because of this, we had to go through the process of getting our instance valid SSL certificates. In the end, we secured them with the Let's Encrypt provider.

Chapter 4

Popularity prediction

It is fairly easy to tell whether a scientific publication is worth reading when we see its impact on the community. For papers that have five years or more, we have strong indicators of popularity like the number of citations, journals or conferences in which it occurred, and some alternative metrics like the number of downloads or the number of web searches. Meanwhile, fresh papers can not be measured in the same way as there is not a sufficient amount of data. It turned out that we had to find a solution for this situation in case a user of our extension wants to explore more recent study papers.

The problem of automatic citability prediction is well known in scientometrics – the field of study concerned with the measurement and analysis of scientific literature. Yet we could not find any pre-trained model or even its pre-implemented architecture resolving that issue. Therefore we were forced to create our one using natural language processing methods[2].

4.1 Dataset

Research papers are usually publicly accessible, frequently with extensive side information like the citation count, the date and place of publication, the field of study, etc. It seems to be very user-friendly as long as you are not trying to get too much data. We tested multiple different APIs from several different electronic repositories of scientific content – arXiv, Elsevier, and Web of Science, to name some of them. We also tried some independent scraping programs like ScraperApi or Scholarly. However, none met our expectations. Each one had its unique drawbacks that prevented collecting data. We had problems with slow response time, lack of information about citation count, captcha blockades, and poorly designed search engines. Nevertheless, we were finally able to collect a significant amount of data thanks to Semantic Scholar. It is an online repository of scientific literature, which answered our request and granted us access to their API for scientific research

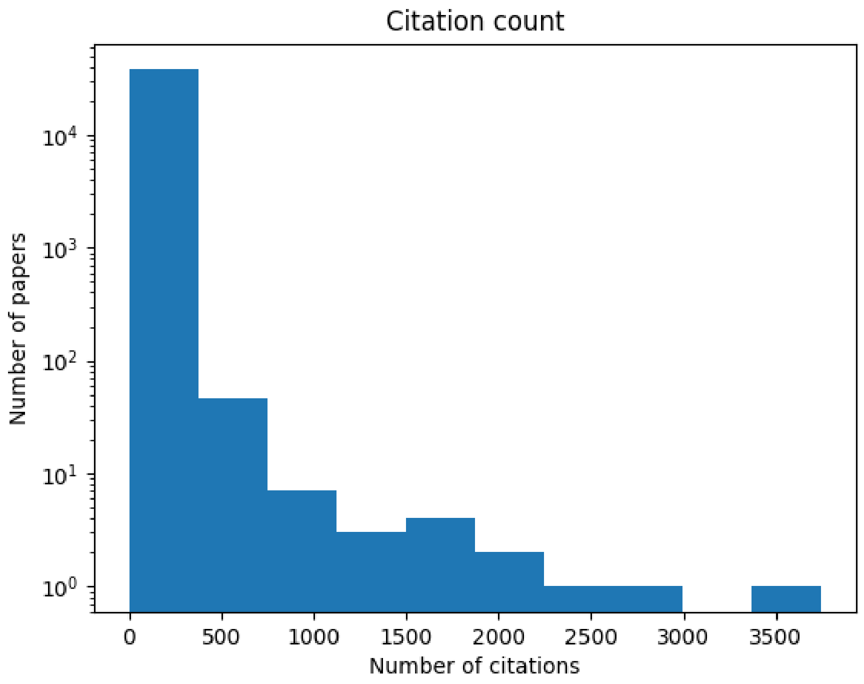


Figure 4.1: Citability distribution in the collected dataset. The number of overall samples might explain the histogram discontinuity and roughness as they are not as prominent in larger datasets[9].

purposes. We collected more than 130 thousand papers, from which about 20% were removed in pre-processing. As a result, we obtained records with information about the following features: title, abstract, authors, authors' affiliations, venue, "is open access", publication year, journal, and citation count.

4.2 Objective

In the best-case scenario, we would like to create a model that can accurately predict the number of citations, especially identifying state-of-the-art publications. However, if we would give any predictive algorithm a goal to minimize MSE it would most likely always answer 0 and achieve a very high score. It is due to the uneven distribution of citation count among publications. As one may suspect, there are very few popular papers meanwhile many are less noticeable. Namely 25% of papers have 0 citations and only 10% have more than 32. The whole distribution of our dataset is presented in Figure 4.1.

To overcome the problem of high skewness around lower values described broadly in literature[5], we decided that our goal would be to guess the interquartile range of distribution rather than a particular number of citations. In practice, it means that we created roughly equal size classes of papers with 0; 1 to 4; 5 to 16; and more than 16 citations, which lead us to a problem of classification instead of regression.

4.3 Feature engeneering

Analyzing any text with machine learning models requires creating a representation of its content and side information that can be understood by a computer. There are plenty of pre-trained models that can create word embeddings. However, there are a few that can be applied to articles or even sentences, not to mention the side information embeddings, which have to be custom-made.

4.3.1 Stylistic features

According to the research of Sergio Jimenez et al. about the stylometry of publication titles and abstracts[9], stylistic features are surprisingly strong predictors of paper citability. For each sample, they were extracting 3578 stylistic features – simple measurements of text like the number of sentences, the average number of tokens in a sentence, the percentage of alphabetic characters, the percentage of numeric characters, the average length of tokens, and so on. The study looked at the impact of individual features on the final prediction. Based on their analysis, we used three metrics with the highest correlation to the result: the POS-tag frequency, the POS-tags bigrams frequency, and Corpus Spectral Signatures.

The first two sets of attributes refer to the analysis of word parts of speech (POS). The process of feature extraction was initialized by splitting the title and the abstract separately into lists of tokens. Each list accordingly was mapped on the equivalent list of POS tags. Next, the frequency of each POS tag was computed and stored as a final feature. We also calculated the frequencies of the 25 POS-tag bigrams that, according to mentioned research, were most promising.

Corpus Spectral Signature (CSS) is a text metric introduced by Sergio Jimenez et al. [9], that takes into account the whole corpus in which it is located. It involves a significant amount of pre-processing, however afterwards we get fast and easy-to-get text characteristics. We will describe the process of applying CSS based on its use in a particular case. For a further explanation, we recommend turning to mentioned publication.

In order to create a bigram Corpus Spectral Signatures, we have to extract all bigrams (consecutive pairs of characters) from the whole corpus. Next, we calculate the negative log frequency of each and split them into ten bins based on that value. The threshold for each bin is evenly spread from 0 to the value of the most common bigram. Now, to calculate the 'signature' of the text we have to assign each of its bigrams to a bin. The number of elements in each bin separately creates ten features which constitute text Spectral Signature.

For simplicity, we described the process of creating CSS with fixed hyperparameters. However, we do not have to and we did not use it only in that way. The number of bins as well as the number of characters in the n-gram can be adjusted.

Table 4.1 presents CSS analysis on three different levels. **a)** shows the distribution of bin values across the whole corpus for bigrams and trigrams. It tells us that the CSS method is applicable to our problem as we get almost the same distribution as the inventors of this metric. The wider look at the spectral analysis presented in **b)** gives us the impression that it in fact gains some valuable information or at least is a strong category indicator. **c)** shows how radically different the Spectral Signatures of low-cited papers are when compared to the most popular ones. Even though the sample here is very small and might not be representative, we know from the work of Sergio Jimenez et al. that the CSS of the publication title and abstract in fact are strongly correlated with the number of citations.

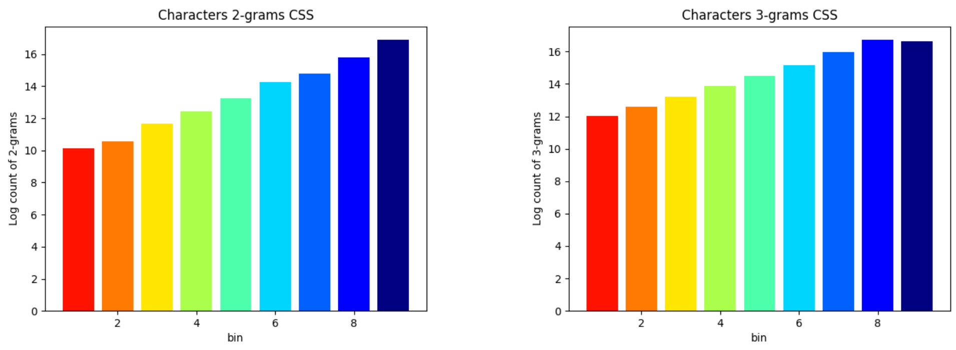
4.3.2 Semantic features

We can imagine that when a scientist specializing in a certain domain reads a title and abstract of a publication, he already knows or at least has strong predictions about the popularity that this publication is going to gain. Therefore analyzing the meaning of its content with an artificial intelligence model seems like the right way to predict citability. We used transformer model[7] to separately create a title and abstract embeddings. We also tested a pre-trained transformer model[10] made for the question-answering problem. Although it gets slightly better results, the time complexity was not acceptable.

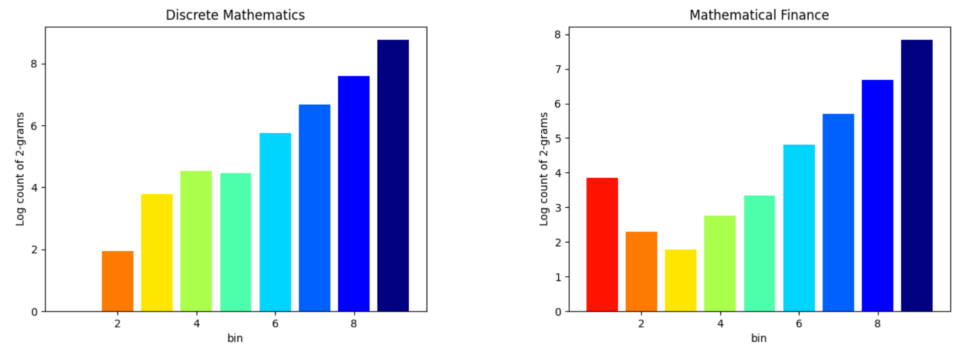
4.3.3 Side information

With no surprise, we found out that such article characteristics as authors' names, their affiliations, and place of publication (journal/conference name) are strong predictors of the number of citations. In our corpus on average 64% of authors' publications were in the same class of citability (0; 1-5; 6-16; ¿16). With affiliations, this score was even higher – 72%. Therefore we considered using this information for predicting popularity.

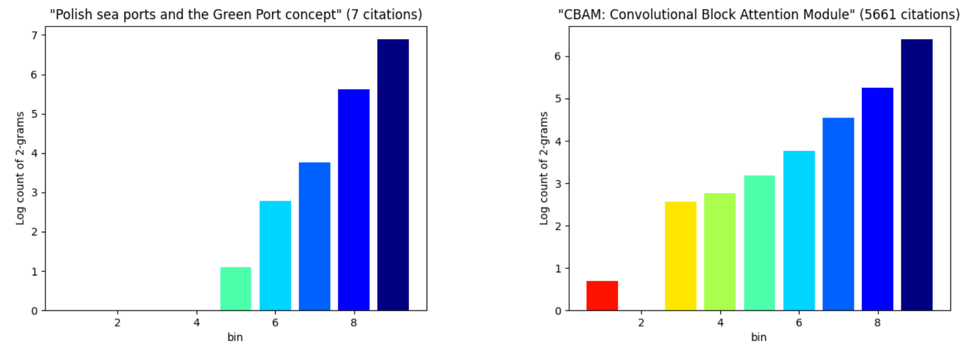
In order to create article characteristics applicable to the predictive model we created three rankings: average citability of authors, average citability of affiliations and average citability of publication venues. Then any new article gets three indicators of being in the top 10% of the ranking.



a) CSS of the whole corpus.



b) CSS of a category.



c) CSS of a single publication.

Table 4.1: CSS analysis. The colours are representing the general frequency of bigrams in a bin. Blue ones are the most frequently seen, meanwhile moving toward red we get the least common.

Multi-layer Perceptron					
Stylistic features	Semantic features	Side information	Training set accuracy	Test set accuracy	Architecture – widths of consecutive layers
X			57.92%	36.02%	40, 40, 40, 40, 40
	X		99.58%	37.52%	900, 500, 200, 100, 50
		X	32.78%	32.71%	70, 40, 20
	X	X	99.66%	38.91%	900, 500, 200, 100, 50
X		X	58.10%	36.88%	40, 40, 40, 40, 40
X	X		99.17%	38.08%	900, 500, 200, 100, 50
X	X	X	98.70%	40.36%	250, 250, 125

Table 4.2: The influence of features on accuracy of MLP classifier

4.4 Model – training and evaluation

All of the abovementioned text characteristic methods create 929 numerical features for each publication, which were used as a basis for our prediction. We tested a few different approaches. Firstly a linear model but it gave us hardly any results even after applying L2 normalization. Next, we moved to more sophisticated ones. As a final choice, we implemented a multilayer perceptron[4] consisting of five fully connected layers of neurons. The layers' width differs accordingly to the number of features extracted from the article.

We tested the accuracy over different combinations of features. The results are presented in Table 4.2. It turned out that the single stronges predictor is the semantic representation of the title and the abstract. That is also highlighted in the combinations of two sets of features. Stylistic embedding used with side information indicators gave worst predictions than any prediction that included semantic analysis. It is kind of surprising that the encountered problem of overfitting in almost every case of training neural networks on semantic embeddings does not cross out the whole attempt. However, if solved, it could lead to much greater results. Nevertheless, the general data behaviour seems to be quite understandable – the more features, the better results.

Our final score of prediction direct class is not as high as in the research of Sergio Jimenez et al. [9] (51.5%). The difference might have a few causes. In the original paper, they were using about 3500 stylistic features, we took from that only the top 150. Moreover, their dataset contains 750 000 publications, which is seven times the size of ours. It might affect the process of training. On the other hand, we used semantic features and external informations about publications, which should lead to better results. Nevertheless, we are satisfied as we do not need an exact classification of scientific papers, but only their popularity ranking, to show users top results.

Chapter 5

Analysis of 'further research' segments

To provide a better understanding of current problems in a certain domain of interest we introduce a new feature that extracts the most crucial of present problems and shows them in an easy-to-access way. When the user wants to analyze an article on Google Scholar with our extension, he or she now has the possibility to check its category and the category field.

5.1 Category prediction

A popular online repository of scientific publications (arXiv) suggests not too broad and yet a complete set of classes that cover the majority of the research topics from natural and formal sciences. Based on their classification, we created a model that can predict the domain of a publication with high accuracy.

For training and testing purposes we used the arXiv dataset, which contains more than 200GB of scientific publications in PDF form. As a way of pre-processing, we extracted the titles, abstracts, and information about the arXiv categories. Then we split the data in proportions of 80% to 20% into training and testing datasets.

The model itself is a combination of a text embedding model and a classifier. In order to get text vectors we downloaded a pre-trained transformer model[7]. Then we used the embeddings to train the K-nearest neighbors classifier. The testing process shown in Figure 5.1 revealed the best hyperparameters that gave model accuracy of 74%.

In order to assess how accurate the results can be about our predictions we calculated 95% confidence intervals. Assuming that we are dealing with a binomial distribution, the accuracy is the maximum likelihood estimator of success in the Bernoulli trial i.e. guessing the right category. To calculate the confidence intervals

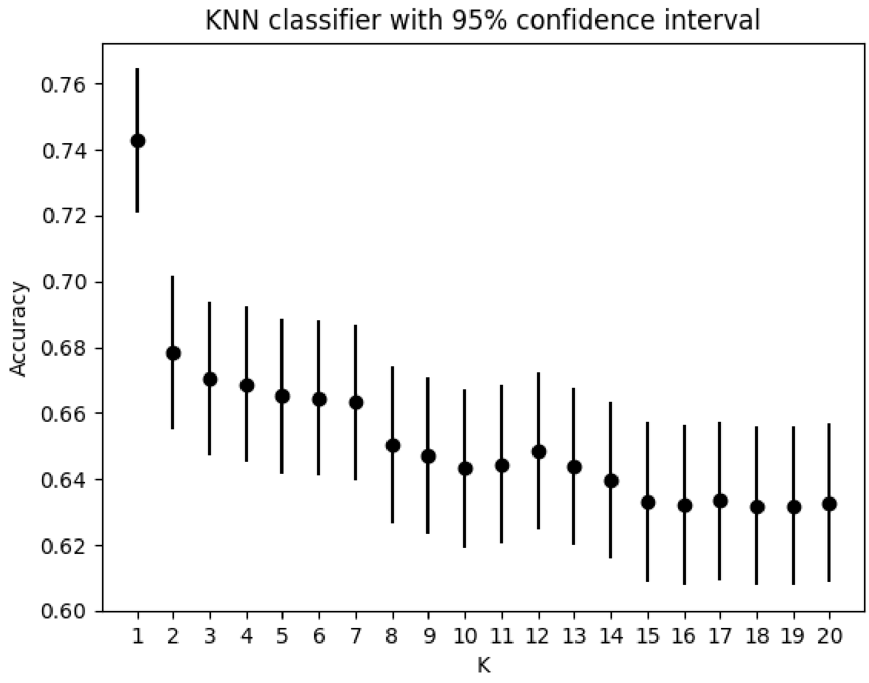


Figure 5.1: Category prediction model testing over hyperparameters

of this estimator we used the Wald method as it is most commonly applied. Results can be seen both in Figure 5.1 and 5.2. The second plot is especially important as the sizes of categories were not equal. In this case, it is possible that the model can achieve high accuracy just by guessing categories with the highest number of samples, treating others as outliers. However, as we can see it does not happen here. The expected accuracy of random predictions is one over the number of categories which is approximately 0.66%. Here even categories with the highest margin of error had significantly better results.

When the category of the publication is revealed we can move to the next step in which similarities between the current publication and the other ones from this domain are presented. For each of over 150 categories, on average we gathered more than a thousand most promising research papers from the last two years. For each domain separately we performed a clusterization procedure to retrieve the most representative papers.

5.2 Clusterization procedure

Each category has far too many samples to present to the user in an ordinary way. Creating a ranking of similar papers would neither be a novelty nor in any way interesting. For that reason, we decided to show the results on a plot where each article would be represented as a point in three-dimensional space. The distances

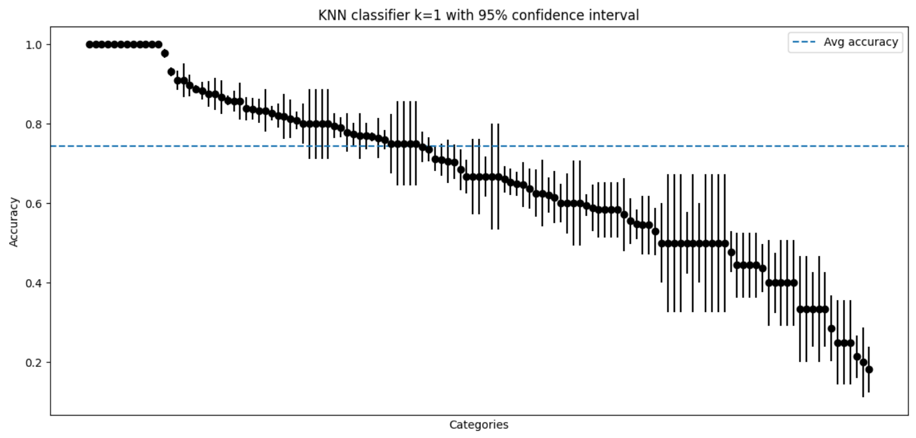


Figure 5.2: Category prediction model testing over categories. To make it easier to read, the results were sorted according to how they scored.

between points are inversely proportional to the semantic similarity of the articles linked to them. It means that not only are we able to show plenty of publications at the same time, but also their sophisticated correlations can be presented for comparison. Still, a standard display 5.3 might seem to be overwhelming. Therefore we propose a hierarchy – three clusters for each category, that groups all publications into three groups with highlighted centroids. This approach suggests to the user to check what the three main topics in a certain domain are as well as show the user a smaller group of articles which are semantically similar to the one that is currently analysed.

The process of clusterization in detail goes as follows. From each paper, we extracted segments that contain information about researchers' struggles and plans for overcoming encountered problems. In order to achieve this we extracted all parts of the papers that contained one of the formulas: 'further research', 'further studies', 'future work', and several others. As a result of this process, we ended up with a list of paragraphs representing each category. Then we used a pre-trained transformer model[7] to create text embedding in 384-dimensional space. Each of those vectors was passed to a K-means algorithm which allocated them into one of three classes. In the following step, we found centroids of each cluster and returned them as the most representative article in a domain. Next, we mapped each embedding into lower-dimensional space and displayed them on a scatter plot 5.4. For additional information, we repeated this process with abstracts instead of previously collected 'further research' sections.

5.2.1 Dimensionality reduction

Dimensionality reduction is a well-known problem, which nowadays can be easily solved with dedicated and pre-implemented algorithms like t-SNE[1]. However, it is not the case here as we have to find a trade-off between the accuracy of the model,

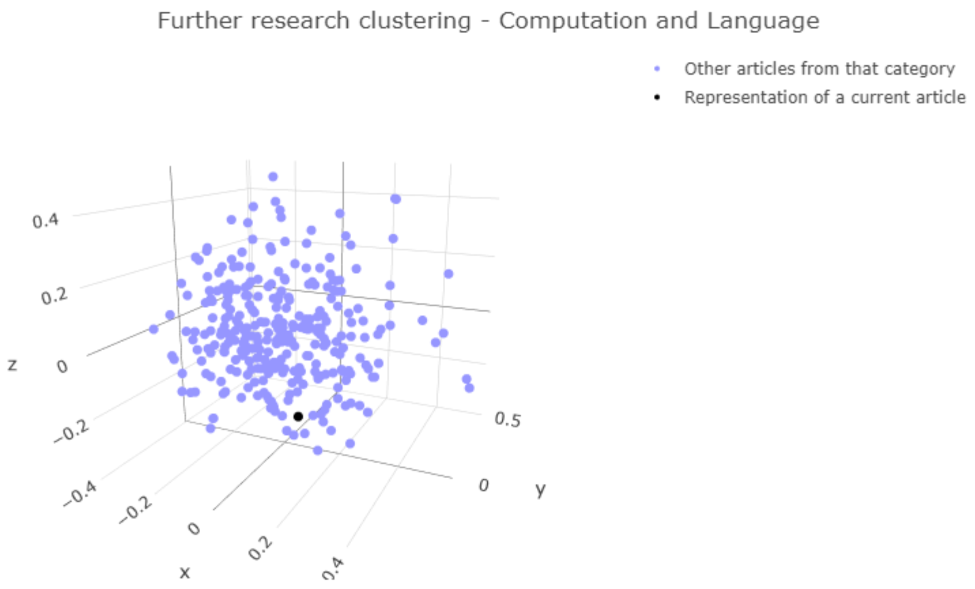


Figure 5.3: Category mapping without clusterization

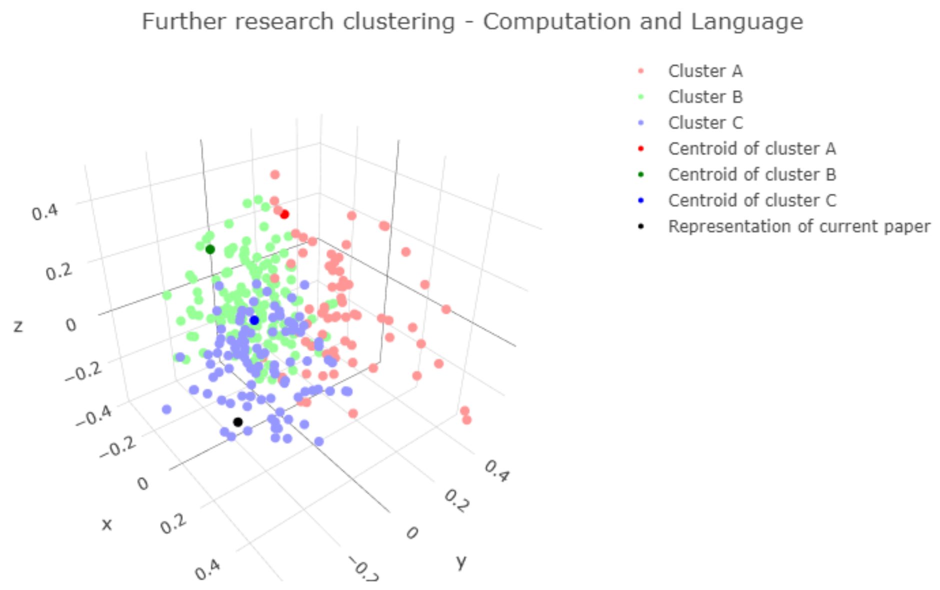


Figure 5.4: Category mapping with clusterization. Points with higher colour intensity are centroids.

its speed, and space requirements. Our algorithm has to be efficient as it will be run on a server during the processing of a user request. t-SNE in this case is either too slow or requires too much space.

When the user wants to analyze some article, our extension predicts its category and extracts a segment that points to further research. Then it creates embedding of the found segment. Now we face the problem of combining it with other publications from this category and mapping it into two or three dimensions. Possible scenarios:

- We compute the embeddings of each paper from the current category. We merge them with the requested by user article embedding. Then we apply t-SNE $\rightarrow$ too time-consuming.
- We store pre-computed embeddings for each article from each category. We create embedding of the requested article, then we apply t-SNE $\rightarrow$ too space-consuming.

To overcome these problems, we assumed a different approach. We used the Principal Component Analysis algorithm to extract three main features of the data. It is not perfect due to loss of information. As a side note, removing all the other dimensions is the cause of displaying centroids apart from the cluster centre. Nevertheless, PCA[8] has a unique property that when it once finds the matrix of transformation, there is no longer need to remember all the data that was used in training. Therefore the solution seems to be a logical compromise in our application. It takes a marginal amount of space - a 3-dimensional vector for each stored article and a matrix of size 384×4 for each category. The time complexity of this approach is reduced to a minimum, which is running a transformer to create text embedding and multiplying it by a matrix of transformation.

5.2.2 t-SNE vs PCA

We performed some tests to see the difference in the performance of the two algorithms in question. Results are displayed in Figures 5.5 and 5.6. To our surprise, PCA seems to look better. Not only are the clusters maintained, but also the data is less spread through space. In order to even be able to display t-SNE results properly we had to remove outliers. It does not mean that PCA is better in general, it is just more user-friendly in this case.

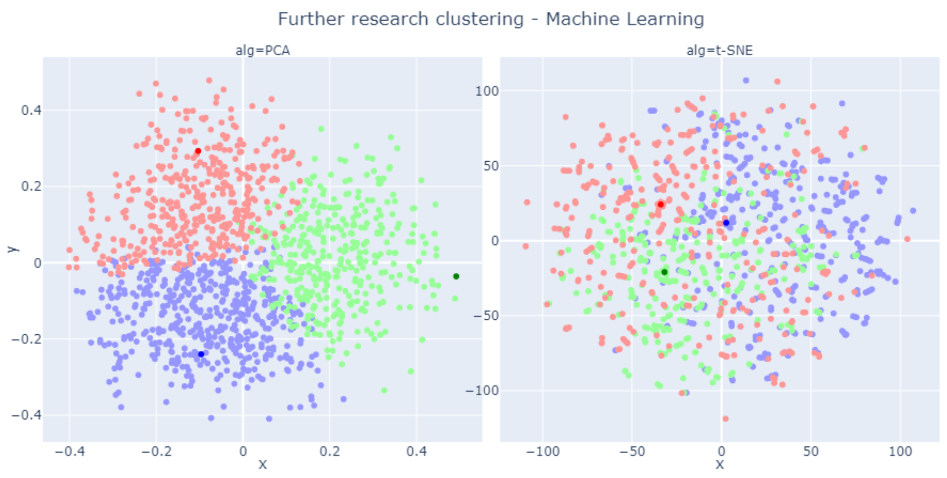


Figure 5.5: Publication mapping onto 2D space with PCA and t-SNE. Colours indicate clusters. Points with higher colour intensity are centroids.

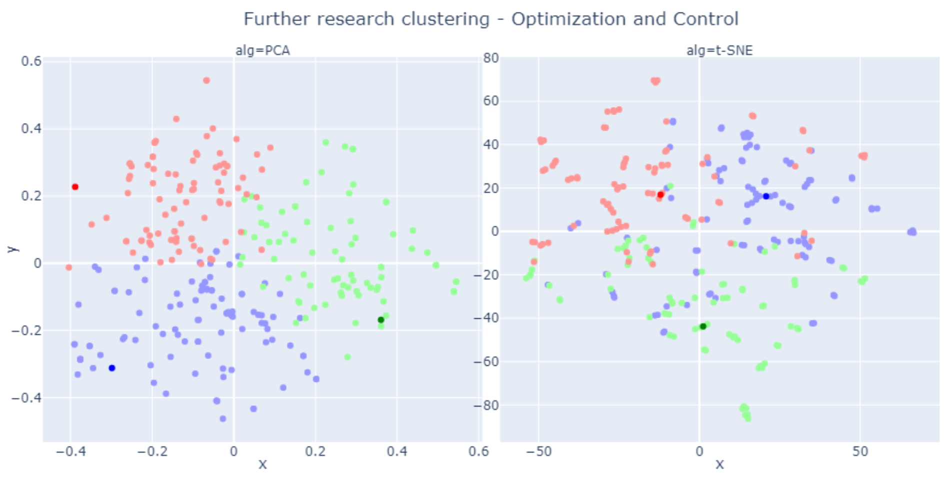


Figure 5.6: Publication mapping onto 2D space with PCA and t-SNE. Colours indicate clusters. Points with higher colour intensity are centroids.

Chapter 6

Conclusions

We achieved all the objectives that we set in the project declaration. We developed an application that might actually be useful for us and others. There were many problems to overcome, and many important development decisions to make. However, we believe ended up successful.

6.1 Further development

Further development is needed in a segment of graph generation. Time required for creating this graph is far from optimal and strongly affects the quality of use. The method of extracting and presenting 'further research' segments might be also improved as it is not always meaningful and is hard to understand without context. Also, we would like to spend more time creating and analyzing the popularity prediction model. It seems like a really interesting problem, which if solved might have a huge impact on overall research proceedings.

Contributions - distinct
Cezary Troska
Frontend - extension skeleton - integrating our features into the Google page - introduction and configuration of webpack - integrating Plotly.js into the extension. - scraping relevant information from the Google Scholar result - Connection Graph part of the extension - asynchronous connections to the backend - loading animations - CSS Backend - Django project skeleton and configuration - Django 'articlegraph' application : ¬ gathering information about the articles in real-time ¬ parsing information into the graph form ¬ introducing proxy servers ¬ parallelization of the process - introduction of the database into the backend - dockerization of the project: ¬ creating containers out of the components of the project ¬ network configuration of this setup - introduction of the Gunicorn server - deployment on Google Cloud: ¬ preparing cloud instance to run the backend ¬ obtaining static IP and configuring the network ¬ SSL certification - automated tests Research - collecting bulk amounts of arXiv articles - collecting bulk amounts of article popularity data from Semantic Scholar - creating efficient tools for mass paring of the data collected and parsing the raw data Document - Abstract - Application overview

Contributions - distinct
Antoni Dąbrowski
Popularity prediction: - sending requests for access to APIs of online scientific repositories - data preprocessing - feature engineering - testing different models - testing the MLP model over different sets of features - testing different architectures of the MLP model - creating visualizations for this document Category prediction: - data preprocessing - testing model over different hyperparameters - extended analysis of the results - creating visualizations for this document 'Further research' analysis: - data preprocessing - text embeddings - solving the problem of dimensionality reduction - clusterization - processing all categories Backend: - establishing communication with frontend - 'further research' plot - creating database for processed articles - developing fast way to process article - preparing data for the plot in the frontend - adding all pretrained model and preprocessed data to a pipeline Frontend: - handling communication with the backend - 'further research' plot - creating the 'further research' plot - adding switching buttons - adding a popup for switching hosts Deployment: - creating Google Developer account - uploading extension - creating all extension descriptions and promotion images - convincing Google staff that our extension is not harmfull This document: - 'Further research' analysis - Popularity prediction - Conclusions - Contributions (template)

Contributions - common		
Cezary Troska	**Contribution**	**Antoni Dąbrowski**
●●●○○	Researching database for citability prediction	●●○○○
●○○○○	Writing chapter: *Introduction*	●●●●○
●●●●○	Writing chapter: *Implementation details*	●○○○○

●○○○○– small contribution, ●●●●○– large contribution

Bibliography

[1] Laurens van der Maaten and Geoffrey Hinton. „Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.

[2] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Upper Saddle River, N.J.: Pearson Prentice Hall, 2009. ISBN: 9780131873216 0131873210. URL: `http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y`.

[3] Dirk Merkel. „Docker: lightweight linux containers for consistent development and deployment". In: *Linux journal* 2014.239 (2014), p. 2.

[4] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* Cambridge, MA, USA: MIT Press, 2016. URL: `http://www.deeplearningbook.org`.

[5] Lutz Bornmann and Loet Leydesdorff. „Skewness of citation impact data and covariates of citation distributions: A large-scale empirical analysis based on Web of Science data". In: *Journal of Informetrics* 11.1 (2017), pp. 164–175. URL: `https://EconPapers.repec.org/RePEc:eee:infome:v:11:y:2017:i:1:p:164-175`.

[6] Ashish Vaswani et al. „Attention is All you Need". In: *Advances in Neural Information Processing Systems.* Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[7] Nils Reimers and Iryna Gurevych. „Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Nov. 2019. URL: `https://arxiv.org/abs/1908.10084`.

[8] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning.* Cambridge University Press, 2020.

[9] Sergio Jimenez et al. „Automatic prediction of citability of scientific articles by stylometry of their titles and abstracts". In: (Dec. 2020). URL: `https://doi.org/10.1007/s11192-020-03526-1`.

[10] Vladimir Karpukhin et al. „Dense Passage Retrieval for Open-Domain Question Answering”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: `10.18653/v1/2020.emnlp-main.550`. URL: `https://www.aclweb.org/anthology/2020.emnlp-main.550`.

[11] Django Software Foundation. *Django*. Version 4.1. Feb. 1, 2023. URL: `https://docs.djangoproject.com/en/4.1/`.