Research Trends and Applications of Data Augmentation Algorithms

Joao Fonseca[1*], Fernando Bacao[1]

[1]NOVA Information Management School, Universidade Nova de Lisboa

*Corresponding Author

Postal Address: NOVA Information Management School, Campus de Campolide, 1070–312 Lisboa, Portugal

Telephone: +351 21 382 8610

1 Introduction

Introduction goes here.

2 Theory

Jürgen Schmidhuber's group shows that a simple MLP architecture can achieve state-of-the-art performance on computer vision benchmarks given strong enough data augmentation [1,2].

[1] Better digit recognition with a committee of simple Neural Nets. Meier, Cires, Gambardella and Schmidhuber 2011 [PDF]

[2] Handwritten Digit Recognition with a Committee of DeepNeural Nets on GPUs. Ciresan, Meier, Gambardella and Schmidhuber 2011 [PDF]

3 Methodology

In this section we describe the procedures defined for the literature collection, data preprocessing and literature analysis. The analysis of the literature was developed with 3 different approaches. Throughout the analyses, data preprocessing and hyperparameter tuning was developed iteratively. The procedure adopted in this manuscript is shown in Figure 1.

The literature collection procedure is described in Subsection 3.1. The data and text preprocessing is described in Subsection 3.2. The exploratory data analysis described in Subsection 3.3 was done to understand which manuscripts, journals and conferences are most significant within the field of Data

Augmentation. The manuscripts' keywords were used to construct a network of keywords (described in Subsection 3.4) and study the different communities of keywords found in the network. The topic modelling and parameter tunning is described in Subsection 3.5. The abstract embeddings procedure is described in Subsection 3.6.

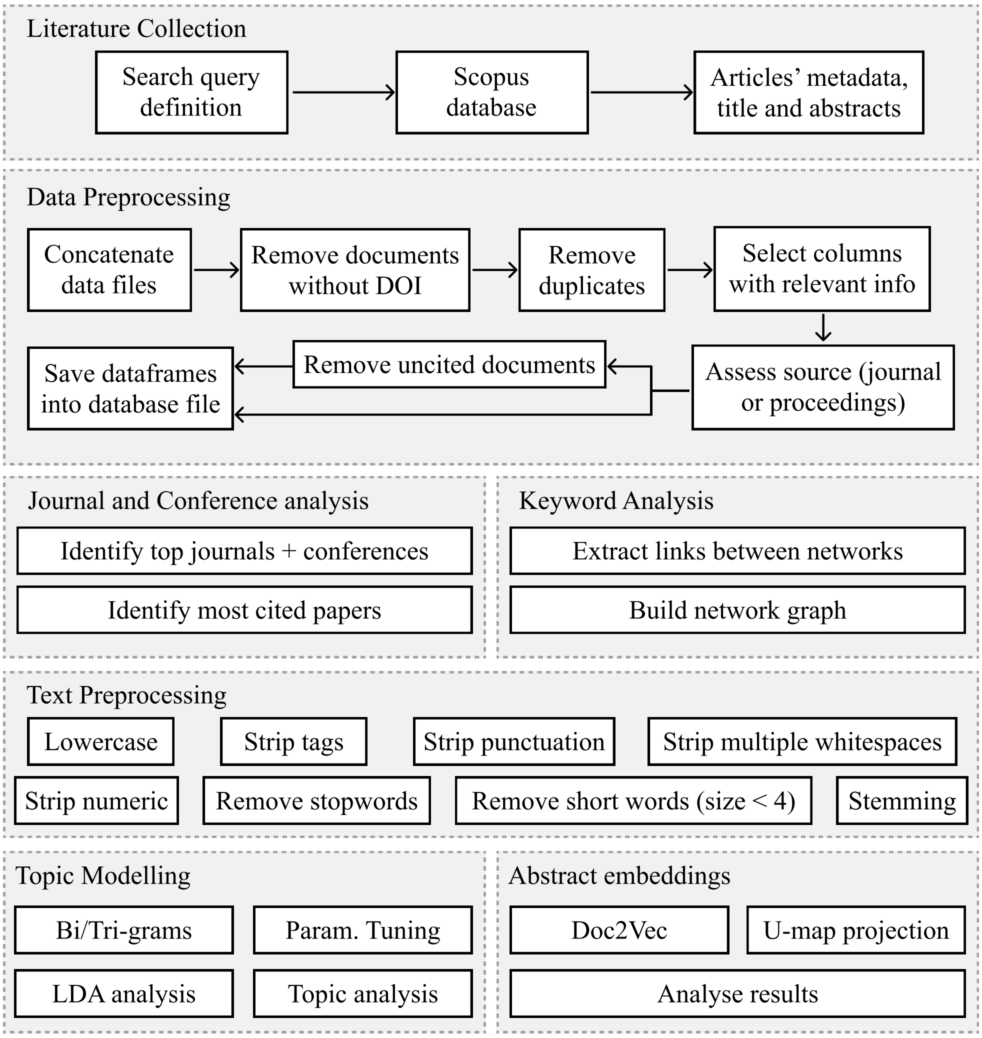

Figure 1: Diagram of the proposed literature analysis approach.

3.1 Literature Collection

The focus of this literature analysis is to understand the different algorithms, domains and/or tasks that employ data augmentation techniques. Therefore, we use the keyword "data augmentation" in order to ensure an unbiased analysis. The results were then limited to conference papers and journal articles written in English that were published in the past 15 years. Due to the large amount of results found, using using solely the Scopus database was found to be sufficient. The resulting query is shown below:

```
KEY ( "data augmentation" )  AND  ( LIMIT-TO ( LANGUAGE ,  "English" ) )
```

```
AND ( LIMIT-TO ( DOCTYPE ,  "cp" )  OR  LIMIT-TO ( DOCTYPE ,  "ar" ) )
AND  (
        LIMIT-TO ( PUBYEAR ,  2021 )  OR  LIMIT-TO ( PUBYEAR ,  2020 )
    OR  LIMIT-TO ( PUBYEAR ,  2019 )  OR  LIMIT-TO ( PUBYEAR ,  2018 )
    OR  LIMIT-TO ( PUBYEAR ,  2017 )  OR  LIMIT-TO ( PUBYEAR ,  2016 )
    OR  LIMIT-TO ( PUBYEAR ,  2015 )  OR  LIMIT-TO ( PUBYEAR ,  2014 )
    OR  LIMIT-TO ( PUBYEAR ,  2013 )  OR  LIMIT-TO ( PUBYEAR ,  2012 )
    OR  LIMIT-TO ( PUBYEAR ,  2011 )  OR  LIMIT-TO ( PUBYEAR ,  2010 )
    OR  LIMIT-TO ( PUBYEAR ,  2009 )  OR  LIMIT-TO ( PUBYEAR ,  2008 )
    OR  LIMIT-TO ( PUBYEAR ,  2007 )  OR  LIMIT-TO ( PUBYEAR ,  2006 )
)
```

The resulting data selection/filtering pipeline is shown in Figure 2. Due to the limitations in the Scopus data export (maximum 2000 documents per export), the data was split in four different time periods and exported separately: 2006 until 2018, 2019, 2020 and 2021, which produced four CSV files.

3.2 Data Preprocessing

The each step of the data preprocessing stage and amount of documents dropped is represented in Figure 2. The data was first concatenated into a single data frame. During this process, we found that one of the exported references had a corrupted line, which caused the loss of one additional document. References without a DOI were disregarded from further analysis. Since they are used as unique identifiers for intellectual property [1], we used this variable to detect and remove duplicate references from the dataset.

Removed references without a single citation, 2259 results. Exception made for journals and conference analysis.

For the network analysis:

Out of the 2259 results, 1923 contained keywords in Scopus' database.

Keyword combinations showing up in only one document are removed from further analysis.

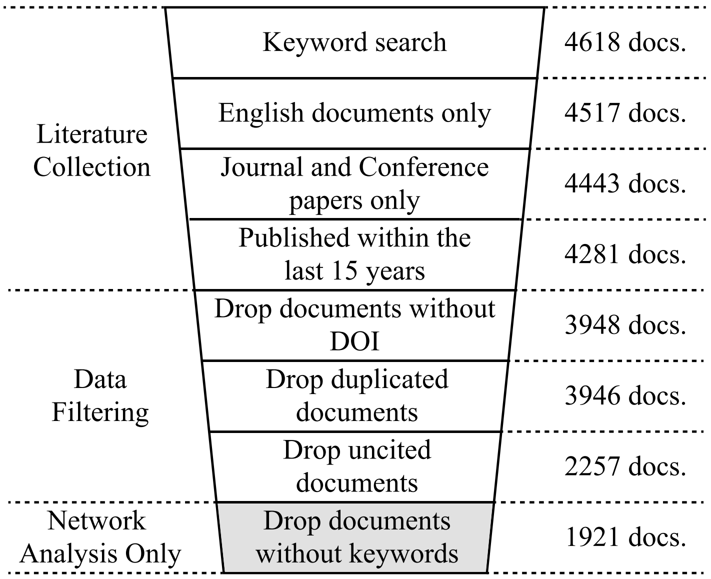

Figure 2: Data filtering pipeline.

The network consists of an undirected graph whose weights consist of the following formula: $Keyword--pairweight = \log(Avgcites * Nbrofdocuments)$ to avoid a disproportional bias of highly cited research articles.

- Discuss data preprocessing done.
- Concatenate data, extract features (if possible), data cleaning etc

3.3 Journal and Conference analysis

3.4 Keyword Analysis

3.5 Topic Modelling

3.6 Abstract embeddings

3.7 Software Implementation

The analysis and modelling was developed using the Python programming language, along with the Scikit-Learn [2], Gensim [3], Umap-Learn [4] and Networkx [5] libraries. The final network analysis and visualization was done with Gephi [6]. All functions, algorithms, analyses and results are provided in the GitHub repository of the project.

4 Results

Results go here.

4.1 PRISMA Flow Diagram

- Create a flowchart with the data cleaning process and describe it.

4.2 Terms Frequency

4.3 Topics Discovered

LDA analysis goes here

4.3.1 Main Journals

4.3.2 Main Conference Proceedings

4.4 Author Co-occurrence Analysis

Not sure whether to keep this one.

4.5 Title and Abstract Text Occurrence Analysis

This can be done by text occurrence network visualization or embedings

4.6 Most Cited Publications

4.7 Application and Method Analysis

5 Discussion

Discussion goes here.

5.1 Research Question Discussion

5.2 Research Gap Discussion

5.3 Study Limitation Discussion

6 Conclusions

References

[1] N. Paskin, "Toward unique identifiers," *Proceedings of the IEEE*, vol. 87, pp. 1208–1227, July 1999.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[3] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

[4] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[5] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11–15, 2008.

[6] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 3, 2009.