



HAL
open science

Scientific search engines : From the categorization to the information retrieval

Bastien Latard

► To cite this version:

Bastien Latard. Scientific search engines: From the categorization to the information retrieval. Computers and Society [cs.CY]. Université de Haute Alsace - Mulhouse, 2019. English. NNT : 2019MULH2986 . tel-03463570

HAL Id: tel-03463570

<https://theses.hal.science/tel-03463570v1>

Submitted on 2 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE HAUTE-ALSACE

DOCTORAL THESIS

**Scientific Search Engines: From the
Categorization to the Information
Retrieval**

Author: Bastien LATARD

Examiners:

Prof. Dr. Frédérique LAFOREST

Dr. Mathieu ROCHE

Prof. Dr. Julien LONGHI

Supervisors:

Prof. Dr. Michel HASSENFORDER

Prof. Dr. Germain FORESTIER

Dr. Jonathan WEBER

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Sciences*



September 19, 2019

Declaration of Authorship

I, Bastien LATARD, declare that this thesis titled, "Scientific Search Engines: From the Categorization to the Information Retrieval" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“If ease of use was the only valid criterion, people would stick to tricycles and never try bicycles.”

Douglas Engelbart

“One never notices what has been done; one can only see what remains to be done.”

Marie Curie

“Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.”

Albert Einstein

UNIVERSITÉ DE HAUTE-ALSACE

Abstract

Université de Haute-Alsace

Doctor of Philosophy

Scientific Search Engines: From the Categorization to the Information Retrieval

by Bastien LATARD

Nowadays, drown by the data abundance, scientists and other readers spend more and more time in bibliographic phase. Searching for relevant reading is a tedious task given that scientific search engines tend to be either selective (i.e., incomplete) or to offer basic search functionalities. The aim of this thesis is to add semantics into scientific search engines in order to propose related articles which are semantically highly similar. Moreover, we have the conviction that adding semantics to search engines will help, in a longer term, to be more innovative and to offer new interesting features to the scientific community. On top of our heads, the building of a neutral and inclusive cross-publisher journal suggester or a tool to prepare a pool of the most relevant articles about a topic.

The first step of such an ambitious goal exploits a knowledge base to categorize articles keywords. This categorization disambiguates and validates the context of these keywords by identifying the categories they have in common. An article can only be categorized when at least two keywords are connected by one of their categories (i.e., they share a common category).

The second step augments the data by extracting all related neighbors—*from the knowledge base*—sharing the same category. Then, a new similarity score is computed involving all types of relationships among keywords and their related neighbors, for each article pair.

Finally, this thesis sits astride several text mining tasks such as word sense disambiguation, categorization or other information retrieval tasks. It achieves satisfactory and promising results for each of those, and competes with state-of-the-art approaches such as the word mover distance exploiting words embeddings.

Acknowledgements

Even though a thesis is an individual work, there were plenty direct or indirect contributors helped me throughout its way. I hereby sincerely acknowledge all of the different actors.

First of all, I would like to thank my thesis director, Prof. Michel Hassenforder and my supervisors Prof. Germain Forestier and Dr. Jonathan Weber. I humbly learned about the laborious path towards becoming a scientist from all of our intensive exchanges around my research and the writing of our papers. I will try to remember your valuable tips and your rigor in scientific analysis.

I am also entirely grateful to Dr. Shu-Kun Lin and the whole management team of MDPI for accepting to fund this thesis. Also a special thank you to Franck Vazquez who helped me to draft the PhD proposal, and Milos Cuculovic, the manager of MDPI IT team.

I also want to acknowledge the precious role of my thesis examiners, Prof. Frédérique Laforest, Dr. Mathieu Roche and Prof. Julien Longhi for their valuable feedbacks. They helped me to make this manuscript clearer and more explicit about the accomplished work.

I also thank my colleagues at MDPI who made my time at the office more pleasant. Namely, Juha, Matthias, Oliver, Séline, Sara, Martyn and all the others. Also a special thanks to the past and present Scilit developers, who helped me to maintain and further develop Scilit during my PhD: Mladen, Darko, Nemanja, Stefan and Nenad. Finally, I also wanted to mention the good work of Nolwenn who helped me to explore some parallel leads that I did not have time to explore within my PhD, and also made the evaluation part more generic.

Another bench of colleagues that I would like to acknowledge are by Lab' mates: Trung, Florent, Houda, Heng, Hassan, Mounir, Baptiste, Tsega, Mac and Robin. This international team open a few interesting discussions, scientifically and humanly. Good luck for your respective PhDs and for your futures, you deserve to be successful. We stay in touch.

This is now time to enter into my private circle and it is with emotion that I do it. First of all, I want to mention my lovely wife who dedicated her entire time to support me on this trip and who freed me a huge amount of time to achieve my objectives. I am very grateful for everything you did, without any complain (or almost none :p). More seriously, I would never have managed to finish it without you. Also a tons of hugs and kisses to my three little loves, Kiara, Livia and Lenzo who remind me every day how good it is to be their father. You four are my balance, my motor and much more!

Another big THANK YOU goes to my entire family for their abundant love. This particularly includes my parents for their education, my siblings for their support and all of my relatives. You all helped me to become who I am. Also, all of my gratitude to my family-in-law who accepted me as their own child. You all contributed to this achievement.

The last but not the least, my mates from my private friends circle, from my football club and from my lovely neighborhood. I always have a good distraction time with all of you. I will not enumerate all names because I would need another bench of pages and I will surely forget some, but the real ones would recognize themselves!

Contents

Declaration of Authorship	iii
Abstract	vii
Acknowledgements	ix
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Publications	2
1.4 Outline	3
2 State of the Art	5
2.1 Information Extraction (IE)	5
2.1.1 Text Pre-processing	5
2.1.2 Bag-of-Words	6
2.1.3 N-gram	6
2.1.4 Words Embeddings	7
2.1.5 Part of Speech (POS)	8
2.1.6 Word Sense Disambiguation (WSD)	8
2.2 Information Retrieval (IR)	8
2.2.1 Term Frequency–Inverse Document Frequency (tf-idf)	9
2.2.2 Vector Space Model (VSM)	9
2.2.3 Dimension Reduction Techniques	10
2.2.4 Distance and Similarity Measures	10
2.2.5 Word Mover Distance (WMD)	10
2.2.6 Classification	11
2.2.7 Clustering	11
2.2.8 Categorization	12
2.3 Recommender Systems	13
2.3.1 Content-Based Filtering	13
2.3.2 Collaborative Filtering	14
2.4 Conclusion	14
3 Categorization	15
3.1 Our Approach	15
3.1.1 BabelNet	15
3.1.2 General Overview	16
3.1.3 Keyword Usage	16
3.1.4 Exact Search	17
3.1.5 Further Search: Split	17
3.1.6 Synsets Connections	18
3.1.6.1 Connection matrices	19

3.1.6.2	Connection— <i>common categories</i>	22
3.1.6.3	No connection— <i>no common categories</i>	23
3.1.6.4	No connection— <i>synsets from the same keyword</i>	25
3.1.6.5	No connection— <i>synsets from the same sub-keywords</i>	26
3.1.6.6	Connection— <i>common categories from different sub-keywords</i>	27
3.1.6.7	No connection— <i>same synset</i>	29
3.1.7	Noise Filtering	30
3.2	Evaluation	32
3.2.1	Offline Evaluation	32
3.2.1.1	Dataset	32
3.2.1.2	Categorization	32
3.2.2	Online Evaluation	34
3.2.2.1	Dataset and extraction overview	34
3.2.2.2	Methodology	35
3.2.2.3	Results	36
3.3	Experiments	39
3.3.1	Threshold Parameter	39
3.3.2	Description and Effects of Used Options	40
3.3.2.1	Split – Classical/Modified n-grams	40
3.3.2.2	Two words exception (2W)	41
3.3.2.3	Numbers filtering (NF)	41
3.3.2.4	Soft filter (SF)	41
3.3.3	Description and Effects of Unused Options	42
3.3.3.1	One letter filtering (1L)	42
3.3.3.2	Single words (SW)	43
3.3.4	Variants Results	43
3.3.4.1	Basic modes	43
3.3.4.2	Further options effect analysis	44
3.4	Future work	48
3.4.1	Problematic Cases	48
3.4.1.1	Valid but non-representative connections	48
3.4.1.2	Poor keywords	48
3.4.1.3	BabelNet limits	49
3.4.2	Solutions and Perspectives	49
3.4.2.1	Extend to text	49
3.4.2.2	Dynamic filtering per journal	49
3.4.2.3	Lemmatization	50
3.4.2.4	Further use of domains	50
3.4.2.5	Lesk algorithm	50
3.5	Conclusion	50
4	Information Retrieval	53
4.1	Our Approach	53
4.1.1	General Overview	53
4.1.2	Data Augmentation	53
4.1.2.1	Generalities	54
4.1.2.2	Neighbors selection – <i>Use case of artificial intelligence</i>	54
4.1.3	Similarity Computation	56
4.1.3.1	Keywords intersection	57
4.1.3.2	Keyword–Neighbor intersection	57
4.1.3.3	Neighbors intersection	58

4.1.3.4	Intersection example	58
4.1.3.5	Summary and big data perspectives	59
4.2	Evaluation	61
4.2.1	Offline Evaluation	61
4.2.1.1	Dataset	61
4.2.1.2	Visualization of synsets and neighbors	61
4.2.1.3	Distance matrix	62
4.2.1.4	Metrics analysis	63
4.2.2	Online Evaluation	66
4.2.2.1	Overall results	67
4.2.2.2	Analysis	69
4.2.2.3	Summary	70
4.3	Experiments	72
4.3.1	Variants Results	72
4.3.1.1	Overall analysis	72
4.3.1.2	Precision / recall exploitation	74
4.3.1.3	ROC exploitation	74
4.3.2	Neural Network	76
4.3.2.1	Features	77
4.3.2.2	Dataset statistics	79
4.3.2.3	Perceptron	79
4.3.2.4	Multi-Layer Perceptron	81
4.3.3	Probabilistic Methods – Word2Vec	83
4.3.3.1	Word Mover Distance	84
4.3.3.2	Comparison	84
4.3.4	Further Investigation of BabelNet Tree	89
4.3.5	Summary	89
4.4	Future work	91
4.4.1	More Data Connected	91
4.4.2	Similarity	92
4.4.3	Benchmarks	92
4.4.4	Neural Network	93
4.4.5	Probabilistic Models	94
4.4.6	An Interactive Evaluation Protocol	94
4.4.7	Industrial Perspectives	94
4.5	Conclusion	95
5	Conclusion / Discussion	97
6	Résumé	101
6.1	Introduction et Motivation	101
6.2	Contributions	101
6.2.1	Catégorisation	102
6.2.2	Extraction de l'information	103
6.3	Conclusion	105

List of Figures

3.1	BabelNet dictionary architecture.	16
3.2	Simplified workflow of our categorization approach.	16
3.3	Split phase: Decreasing n-gram generated after stopwords removal.	17
3.4	Theoretical representation of a category connection.	21
3.5	Legend.	22
3.6	Simple connection—synsets sharing common categories are the meaningful entries.	23
3.7	No connection—Categorization fails because related synsets do not share any category.	24
3.8	No connection—Filtering of common category shared by synsets from the same keyword.	25
3.9	Sub-keywords—Filtering identical sub-keywords (<i>segmentation</i>). No connection, no category extracted.	26
3.10	Sub-keywords—Filtering and connection between <i>segmentation</i> and another sub-keyword (<i>random walker</i>) from the same two keywords.	27
3.11	Sub-keywords—Filtering and connection between <i>segmentation</i> and another keyword <i>watershed</i>	28
3.12	Sub-keywords—Filtering and connection among other sub-keywords (<i>plane</i> and <i>line</i>) from the same two keywords. The correct but not most representative category <i>Mathematical_concepts</i> is extracted.	29
3.13	No connection—Filtering of identical synsets from two different keywords/sub-keywords, and connection with categories from another keyword.	30
3.14	Noise filtering. Extra constant noisy categories (in gray) are de-activated.	31
3.15	Offline evaluation – articles journal distribution	32
3.16	Publisher distribution.	35
3.17	Evaluation—categories and domains.	37
3.18	Categories ratings overview.	38
3.19	Distribution of ratings per articles.	38
3.20	Split—Classical vs. Enhanced n-grams.	40
4.1	General workflow of our complete approach.	53
4.2	Neighborhood of Artificial intelligence synset.	55
4.3	Theoretical similarity intersections.	57
4.4	Articles connection from keywords' (purple), neighbors' (blue) and keyword-neighbor (pink) intersections.	59
4.5	Global scheduling of our processes.	60
4.6	Offline evaluation – articles journal distribution	61
4.7	A tag cloud of journals' synsets and neighbors. Gray: <i>Religions</i> / Purple: <i>Symmetry</i> / Green: <i>Viruses</i> / Blue: <i>Toxins</i>	62

4.8	Distance matrices with (A) / (C) three distinct journals and (B) / (D) four journals including two related ones, for thresholds: 0.98 and 0.995. The purple, green, blue, and orange points represent articles in abscissa respectively from the journal <i>Symmetry</i> , <i>Viruses</i> , <i>Religions</i> , and <i>Toxins</i> .	64
4.9	Metrics of the 4 journals predictions ($\alpha = 4$, $\beta = 2$, $\gamma = 1$)	66
4.10	Evaluation–Keywords intersection , or another one with neighbors-neighbors...	67
4.11	Evaluation–Ratings distribution	68
4.12	Metrics / Curves of the 4 journals predictions (weights variations)	73
4.13	ROC and F1 curves of the 4 journals predictions (weights variations)	75
4.14	Distance matrices for best point from the ROC curves	77
4.15	General workflow of our approach using neural network. Keywords are first categorized and augmented. Then, a neural network learns to predict similarity, based on their journal belonging.	78
4.16	Perceptron predictions accurateness percentage.	80
4.17	Perceptron – Similarity matrices of train and test data sets.	81
4.18	MLP predictions accurateness percentage.	83
4.19	MLP – Similarity matrix of test data (TP: green, TN: light green, FP: red, FN: light red)	83
4.20	Curves for our approach (pink line) and WMD ones. WMD approaches are based on Google News Word2Vec model (gray line) and slim one (green line)	85
4.21	Distance matrix for WMD using Word2Vec slim model – threshold: 1.271 (best point from ROC curve)	87
4.22	Extrapolated time for approach scalability for WMD-W2V (gray), WMD-W2V (slim – green) and our approach (pink)	88
6.1	L’aperçu général de notre approche.	102
6.2	Exemple d’intersections héritées de notre approche.	104

List of Tables

3.1	Sample of the dataset.	33
3.2	Categorization—metrics.	34
3.3	Domain overlapping.	36
3.4	Threshold parameter (α) defines the restriction of the selection criteria.	39
3.5	Categorization—metrics (domains).	42
3.6	Categories - Precision – 3 modes.	43
3.7	Categories - Recall – 3 modes.	44
3.8	Categories - Coverage – 3 modes.	44
3.9	Categories - Harmonic mean – 3 modes.	45
3.10	Categories - Precision – all options effects.	45
3.11	Categories - Recall – all options effects.	46
3.12	Categories - Coverage – all options effects.	46
3.13	Categories - Harmonic mean – all options effects.	47
4.1	Top 2 synsets and neighbors per journal	62
4.2	Predictions naming. Y: Yes, N: No, TP: True Pos., FP: False Pos., TN: True Neg., FN: False Neg.	63
4.3	Predictions with best precision	65
4.4	Predictions with best precision/recall compromise	65
4.5	Predictions with best ROC curve	65
4.6	Weight variants and their legend	72
4.7	Most precise threshold selection.	74
4.8	Best point from the precision / recall curve	75
4.9	Best point from the ROC curve.	76
4.10	Distribution of article pairs according to their type of intersection within the entire dataset	79
4.11	Distribution of article pairs (and accuracy) according to their higher type of intersection within the test set and the set of bad/good predictions	80
4.12	Distribution of bad/good predictions and accuracy of the MLP results	82
4.13	Word2Vec-WMD vs. BabelNet – Predictions with best precision.	86
4.14	Word2Vec-WMD vs. BabelNet – Best point from the precision / recall curve.	86
4.15	Word2Vec-WMD vs. BabelNet – Best point from the ROC curve.	86

Dedicated to my beloved family without whom the achievement of this thesis would have never been possible.

Chapter 1

Introduction

People speak about big data since the early days of Internet¹. However, digital data grew exponentially and nowadays we create every two days as much data as was ever created up to 2013². Moreover, the size of Internet doubles every two years³. The same behavior is encountered in scientific literature, where the number of yearly published articles constantly increases⁴. Even though its growth is less exponential, the trends seems to be a constant growth factor. Web users, as well as researchers easily get drowned within this data deluge [43] and searching for relevant information is nowadays a tedious task. In despite of the numerous scientific search engines, the bibliographic phase is still a complex and tremendous task that researchers regularly face. This thesis aims to help scientists and other readers to facilitate this mining process by proposing some approaches to add semantics into scientific search engines.

1.1 Context

MDPI⁵ is the first fully open access (OA) publisher in terms of number of articles yearly published since 2016. It is the third bigger one (behind Springer and Elsevier) when hybrid publishers are considered. I built up Scilit⁶ during an internship at MDPI for my master thesis in 2013. The idea of Scilit is to daily integrate metadata of scientific articles as soon as a DOI (Digital Object Identifier) is allocated or whenever articles are deposited to PubMed⁷. Different extra sources are crawled and data from those are merged into a single and centralized database. On top of it stands the Scilit Solr⁸ search engine, used for its speed and efficiency for retrieving documents based on full text search.

Searches on Scilit are basic and only a classical keywords search is possible. In addition, a basic related articles widget based on keywords exact matching is implemented in articles pages. No other possibilities are currently conceivable because

¹see this ngram viewer graph for the occurrence of *big data* in books corpus: https://books.google.com/ngrams/graph?content=big+data&year_start=1960&year_end=2008

²<https://techcrunch.com/2010/08/04/schmidt-data/>

³<https://www.live-counter.com/how-big-is-the-internet/>

⁴<https://www.scilit.net/statistic-publishing-market-article>

⁵<https://www.mdpi.com/about>

⁶<https://www.scilit.net/>

⁷<https://www.ncbi.nlm.nih.gov/pubmed/>

⁸Solr (<http://lucene.apache.org/solr/>) is an open source search platform built on Apache Lucene. It is particularly powerful for full-text search, database integration, faceted search and other indexing/querying purposes. Solr is also highly scalable with its SolrCloud feature. All of this makes it a trusted search platform.

no relation between articles are identified, except that they are from the same journal/publisher or that they have words/keywords in common. So the main drawback of Scilit is that articles are not grouped together by field of application and that searches—even though search functionalities are as complete as possible—remain limited.

To tackle these problems, this thesis arises the following questions:

- Is it possible to automatically categorize scientific articles, on any topic, without supervision?
- Is there any way to connect scientific articles with some levels of semantics, rather than using probabilistic models?

We believe that categorizing articles can help users of scientific platforms to browse and navigate through articles more easily. In addition, it may also help to recommend articles within the same topic/context. Finally, it could even help users to refine their ambiguous searches when the requests embrace articles from different distinct fields. We have the intuition that a fast, complete and semantic search engine for the scientific literature may be of interests to scientists and other readers.

1.2 Motivation

Adding semantics into search engines not only potentially adds new and more meaningful related articles pairs but also offers a way to explain predictions. Then, the reader can judge the relevance of the suggestion and decide whether to click it or not. Recent approaches tend to be based on probabilistic models, hence they suffer from the black box effect from which it is really hard—*seen impossible*—to explain suggestions. Therefore, the challenging objective of this thesis is to propose related articles by exclusively using a knowledge base. Moreover, we aim to compete with existing probabilistic approaches.

1.3 Publications

This thesis aroused publications in national and international peer-reviewed conferences, as well as one journal article.

Categorization – in international conferences

– Bastien Latard, Jonathan Weber, Germain Forestier, Michel Hassenforder. Towards a Semantic Search Engine for Scientific Articles. *TPDL*, pp. 608-611. Springer (2017)[67]

– Bastien Latard, Jonathan Weber, Germain Forestier, Michel Hassenforder. Using Semantic Relations between Keywords to Categorize Articles from Scientific Literature. *ICTAI*, pp. 260-264. IEEE (2017)[68]

Categorization – in national conferences

– Bastien Latard, Jonathan Weber, Germain Forestier, Michel Hassenforder. Catégorisation d'articles scientifiques basée sur les relations sémantiques des mots-clés. *EGC*, pp. 371-372. (2018)[66]

Information Retrieval – in international peer reviewed journal

– Bastien Latard, Jonathan Weber, Germain Forestier, Michel Hassenforder. Categorization of Scientific Articles for Data Expansion and Semantical Linking. (submitted to *Information Processing and Management*, 2019).

Information Retrieval – in international conferences

– Nolwenn Bernard, Jonathan Weber, Germain Forestier, Michel Hassenforder and Bastien Latard. Knowledge-Based Categorization of Scientific Articles for Similarity Predictions. (submitted to *WSDM*, 2020).

– Hojjat Rakhshani, Bastien Latard, Mathieu Brévilliers, Jonathan Weber, Julien Lepagnot, Germain Forestier, Michel Hassenforder and Lhassane Idoumghar. Automated Machine Learning for Information Retrieval in Scientific Articles. (submitted to *WSDM*, 2020).

1.4 Outline

Chapter 2 – State of the Art

This thesis starts with a brief introduction of the text mining, given that this thesis is clearly inscribed in the scope of this field. From the information extraction tasks to the information retrieval, touching on categorization and other recommender systems, the main concepts of most common tasks embraced into text mining are introduced.

Chapter 3 – Categorization

The second chapter presents our first contribution which is the categorization of scientific articles. The details about our chosen methodology and its implementation logic are given in Section 3.1. After that, the results of our evaluations are presented in Section 3.2, followed by an overview of our different lab experiments (Section 3.3). Then, some ideas about how to improve our results and a few perspectives are discussed in Section 3.4. And finally, a brief summary of the categorization is given to conclude this chapter (Section 3.5).

Chapter 4 – Information Retrieval

Chapter 4 describes the second main contribution of this thesis, the retrieval of related articles. It follows the same structure as Chapter 3 by starting with the details of our novel approach, which connect articles with a new semantic similarity metric (Section 4.1). Section 4.3 mentions different experiments implemented to compare the results of our chosen approach, going from probabilistic to neural network approaches. Evaluations of the different approaches tested are discussed in Section 4.2, and a recital of future works is given in Section 4.4. Finally, a little conclusion close this chapter (Section 4.5).

Chapter 5 – Conclusion

Chapter 5 is a general conclusion of this thesis. Given that local conclusions are already given in their dedicated contributions sections in Chapter 3 and Chapter 4, this chapter makes the bridge between by concluding this three years work and mentioning a few perspectives which would improve the proposed approach.

Chapter 2

State of the Art

Digital libraries are nowadays larger and larger and this is only the beginning. Indeed, 90% of the worldwide data has been produced the last two years, according to a report from IBM¹. Consequently, web users can easily get drowned among the huge amount of data those represent, and assisting them to search and discover relevant content became crucial. The same assessment can be observed for scientific literature, from which more than 100 millions of works have already been published [56]. Indeed, printing 1 page of every scientific paper will lead to a stack bigger than the Mount Everest². Consequently and thankfully, a lot of research aroused within the field of text mining (TM) which can be decomposed into two distinct but related areas, namely the information extraction (Section 2.1) and the information retrieval (Section 2.2). Algorithms used in TM are widely applied to various other related domains such as recommender systems (Section 2.3), natural language understanding (NLU) and many more applications (e.g., data analysis, search engines, etc).

This chapter has not for objective to be exhaustive, but rather aims to provide core concepts of most common methods realizing these tasks. For interested users, more exhaustive surveys on TM have been written [5, 47, 125].

2.1 Information Extraction (IE)

The goal of information extraction (IE) is to extract meaningful information from unstructured document. Its domain of application is broad given that we may find sub-divisions of IE for the processing of videos, images or any other document types (e.g., audio, html, and so on). Only main approaches applied to TM are presented in this section in order to narrow the bibliography to our topic.

2.1.1 Text Pre-processing

In order to mine a collection of documents, a pre-processing step is often needed [96], even mandatory. Most of the text mining approaches model a document with a few features (i.e., word, bag-of-words, etc.) representing it, often using a vector [47]. For this purpose, a pre-processing step removing meaningless words or determining most important ones is most of the times preliminary applied to document collections. This reduces the size of the dictionary used to represent documents (i.e., vectors), and fastens their processing (i.e., analysis, comparison and so on).

¹<https://tinyurl.com/ibm-internet>

This growth takes all digital data into account (sensors included), not only Internet data.

²The Mount Everest currently culminates at 8,848m whereas a stack of 100 millions paper sheets will be 10,000m high – 100 millions * 0.1mm (basic paper thickness)

The *stopwords filtering* is commonly used given that they are meaningless. Indeed, common words with a very little meaning such as prepositions, linking words or articles (definite or indefinite) do not bring any information about the context of a document and do not help to distinguish or group documents.

The *lemmatization* is another method aiming to reduce the size of the dictionary. Its goal is to return the lemma of the word, which is its canonical form (i.e., without inflectional ending). For example, the words *writing*, *wrote* or *writes* all have the same lemma: *write*. Its drawback is that the *lemmatization* needs an understanding of the words, thus often the usage of a dictionary, a controlled vocabulary or a knowledge database.

Stemming is close to the *lemmatization* but it is much simpler given that it rather removes affixes (i.e., end of the words), rule based, without attempting to build an existing word. This method is usually much faster than lemmatization given that it does not need any word analysis (see Porter algorithm [94] for more details). An example of the difference between both algorithms is that the *stem* of the word *biological* is *biolog*, whereas its lemma is *biology*.

Techniques aiming to extract knowledge from text [122], or even the entire IE field [79] is sometimes also considered as part of the pre-processing step in the literature. Even though it could make sense to embrace the whole IE field while considering the knowledge extraction as a preliminary step of the information retrieval, we rather distinguish IE and other TM tasks from pre-processing step.

2.1.2 Bag-of-Words

Bag-of-words (BoW) has been widely used in text mining field [8] as a preliminary step of various tasks such as classification, similarity computation and word sense disambiguation [84, 113, 117]. Its objective is to represent a document (i.e., text) as a vector of unique words, usually with a counter of the number of occurrences for each word. This is particularly useful to reduce document size and hence also makes textual analysis more efficient and scalable.

Let us take the following two sentences as an example:

sen_A: "The information about alive buried victims extraction by the firemen is amazing"

sen_B: "This approach uses information extraction to augment the data"

The dictionary for both sentences will include all unique words (i.e., no duplicated terms):

{"the", "information", "about", "alive", "buried", "victims", "extraction", "by", "firemen", "is", "amazing", "this", "approach", "uses", "to", "augment", "data"}

And their respective bag-of-words vector representations will be the following:

bow(*sen_A*): [2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]

bow(*sen_B*): [1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]

In order to reduce the vector sizes, pre-processing methods may be applied (see Section 2.1.1). Then, only meaningful and optionally normalized terms will remain and comparing vectors should be more efficient. The drawback of the BoW logic is that multi-words context is lost. For example, the concept inherited from the words *information extraction* will be lost while splitting it into two separated words.

2.1.3 N-gram

The n-gram (or ngram) logic tackles the loss of context for multi-words by creating a bag of *n* words. Originally, this approach was mainly used for determining the most

probable word to come in regard to the previous words (i.e., computational linguistics or probabilistics). Nowadays, approaches using n-gram might also be relevant to extract frequent expressions or word combinations within a context. For example, extracting bigrams (two words) and/or trigrams (three words) from a corpus of text and determining a threshold to cut off marginal associations could be a way to extract meaningful multi-words. Therefore, the interesting bi-grams (i.e., 2-grams) *buried victims* and victim extraction would be found from sen_A , as well as meaningless ones such as *information about* or *about alive*. Consequently, its usage is broad and covers all tasks of TM, such as text classification, text summarization, recommender systems and so on [24, 32, 85].

2.1.4 Words Embeddings

A word embedding is a vectorial representation of words neighborhood. This is based on Firth's idea stating that "*a word is characterized by the company it keeps*" [34]. Nowadays, words embeddings are most of the time associated with *Word2Vec* [81], a machine learning approach using a two-layers neural network to learn semantic word proximity. Indeed, *Word2Vec* has been used for various TM tasks including clustering, keywords extraction, classification and many others [27, 37, 97, 128]. The models trained by *Word2Vec* learning techniques represent the syntactic probabilities of words co-occurrence and can be used to predict next words in a sequence. They return a vector of 300 dimensions (300D) per word, from which the cosine distance is usually calculated in order to extract similarity (or proximity) with vectors of other words.

These models also offer the possibility to search for words analogy by doing vectors operations, such as the equation $vector('Paris') - vector('France') + vector('Italy')$ which returns a vector close to $vector('Rome')$. Two different learning techniques are proposed by Mikolov et al. [81], namely continuous bag-of-words (CBOW) or skip-gram. The core logic of these techniques is described in the following paragraphs.

CBOW.

The CBOW learning mode uses a sliding window and considers each word together with its surrounding words (i.e., the context). If we take a context window of size 2, the word *victims* in sentence sen_B (from Section 2.1.2) has the context [buried, extraction]. Models inherited from CBOW are used to predict a word given a given context, based on co-occurrence probability. I.e., the word *victims* might be expected given the context "*buried ??? extraction*".

Skip-gram.

Models generated by the skip-gram approach aim to achieve the inverse of those from CBOW models. Indeed, their goal is to predict the context (i.e., the probability of surrounding words) of a given word. For that, every *k-skip-n-grams* [41] are extracted, where *k* is the number of skippable words to create the pool of n-grams. The word *victims* from the sentence sen_B has the following *1-skip-bi-grams*: *alive victims*, *buried victims*, *victims extraction* and *victims by*.

2.1.5 Part of Speech (POS)

The part of speech (POS) is the semantical class of a word (i.e., noun, verb, preposition, etc.) in a given sentence. The extraction of POS is nowadays often performed with existing syntactical analyzer such as CoreNLP [75], SyntaxNet [7] or Spacy [46]. Widely used in the literature, these tools not only return the POS of each word within the sentence in entry, but also its syntactic tree (i.e., the relationships between words). Most of the time, named entities are also recognized. Extracting the POS is especially interesting because it helps for a better understanding of the context where a word is given.

2.1.6 Word Sense Disambiguation (WSD)

Word sense disambiguation (WSD) is the task determining the sense of words in regard to the context where they are used. Navigli [87] stated that WSD can be seen as a classification task, where classes are the senses and the classifier assigns related classes by taking into consideration the context where words appear. WSD is often implemented with unsupervised techniques relying on knowledge databases such as specialized thesauri, dictionaries or ontologies, but may also be based on supervised classifiers (e.g., decision trees, naïve bayes, support vector machines). However, supervised WSD would need a big amount of labeled data—*even though some approaches aim to tackle this problem by automatically labeling the sense of unlabeled corpus* [100]—hence most of the WSD approaches are unsupervised knowledge-based [22].

WordNet [82] is commonly used in WSD tasks because it combines advantages of dictionaries and ontologies [77, 90, 115]. Indeed, it can be used both as a dictionary from which several senses are returned for a given word or as an ontology given that all words in relation are also proposed from each dictionary entry. Several approaches use WordNet as a graph and identify connections within the graph to predict the correct meanings (i.e., realize the WSD) [102] [80].

The Lesk algorithm [69] placed on top of a knowledge database may also be used for WSD. This counts the number of words overlapping among all potential senses of two words, and assumes that the higher intersections cardinality validates both meanings and their corresponding entries. Even though this approach remains interesting, computing overlapping among all combinations of all words is too expansive and would not scale well.

Another approach achieving a kind of WSD is the word embeddings where words are projected into several dimensions in regard to morphological relationships learned from a corpus. Section 2.1.4 provides more details about this approach.

2.2 Information Retrieval (IR)

The field of information retrieval (IR) is a part of the TM area, hence most of the techniques used in IR are directly inherited from TM. The difference is in the fact that IR aims to retrieve most relevant documents from a corpus regarding a given query. Therefore, IR often combines TM techniques and similarity measures in order to retrieve closest documents. In this section, the most commonly used approaches are introduced, such as *tf-idf* (Section 2.2.1), the *vector space model* projection (Section 2.2.2) but also dimension reduction techniques (for scalability optimization – Section 2.2.3) and classical similarity measures (Section 2.2.4). We decided to rather

focus on classical IR tasks, but some progress was made in semantic IR [29, 58, 72] and in neural IR [49, 89, 107].

2.2.1 Term Frequency–Inverse Document Frequency (tf-idf)

The term frequency–inverse document frequency (tf-idf) [109] is one of the most used statistics in TM, IR and RS [13]. A tf-idf weight reflects the importance of a term in regard to its frequency within a document (i.e., term frequency) and its overall usage by all the documents corpus (i.e., inverse document frequency). The term frequency (tf) measures the frequency / importance of a term within a document. Several different variants exist but let us introduce the most commonly used one which divides the occurrence frequency of a term t in a document d (i.e., $f(t,d)$) by the total number of terms in this document ($|d|$).

$$tf(t,d) = \frac{f(t,d)}{|d|} \quad (2.1)$$

The inverse document frequency (idf) measures the importance of a term t in a document corpus D by reducing its weight when occurring often in the corpus. $|D_t|$ represents the number of documents where t appears.

$$idf(t,D) = \log \frac{|D|}{|D_t|} \quad (2.2)$$

Finally, tf-idf is the product of tf and idf ($tfidf(t,d,D) = tf(t,d) * idf(t,D)$). In other words, the more a word is represented within a document, the higher the tf weight. In contrary, the more common it is in the entire corpus, the lower the idf. Given that tf and idf are multiplied, the higher tf-idf weight is obtained for a very well represented word in a document while staying rare in regard to all documents.

2.2.2 Vector Space Model (VSM)

Vector space model (VSM) [108] is the representation of the corpus into a $m * n$ matrix where columns represent the corpus documents and rows embrace all terms of the entire corpus vocabulary. It might be seen as a concatenation of all documents' BoW (see Section 2.1.2) using the entire corpus vocabulary in index. This vectorization is specifically useful to build a probabilistic semantic model somehow representing the word occurrence (or co-occurrence) in a specific context. Often, the matrix elements represent the terms occurrences per document, hence the entry $X_{i,j}$ is the number of times the term i occur in document j . Sometimes, the matrix is binary and represents the term presence, but it may also include the tf-idf in order to reflect the term distribution in regard to the entire corpus.

Finally, VSM provides the ability to project vectors into space and operates vectors comparisons or other operations. This is the reason why it has been intensively used in text mining [17, 47], recommender systems [13, 73], and other tasks involving some similarity detection (i.e., classification, clustering, etc) [36, 50, 51, 119]. The most common operation is the cosine of the angle between two vectors in order to determine their similarity. More similarity metrics are given in the dedicated section (Section 2.2.4).

2.2.3 Dimension Reduction Techniques

Nowadays, living in the big data area, classical IR techniques struggle both to keep providing real time solutions and to scale well for big libraries. For that reason, a bench of dimension reduction approaches emerged to fasten the comparison and the retrieval of documents from (too) big matrices. The single value decomposition (SVD) [39] is one of the most used reduction techniques in TM [98]. It decomposes a complex matrix into singular values (a square root of eigenvectors). For that, the $m * n$ term per document matrix M is decomposed such as $M = U \Sigma V^*$, where U and V are respectively the unitary matrices of size $m * m$ and $n * n$. M is a rectangular diagonal matrix of size $m * n$ containing singular values (see [39, 40] for mathematical computation details).

Other dimension reduction techniques exist—those are mostly adaptation of the SVD logic—such as the principal component analysis (PCA) [124], latent semantic indexing (LSI) [28], probabilistic LSI [45], latent dirichlet allocation (LDA) [20], latent semantic analysis (LSA) [65], probabilistic LSA (pLSA) [45], non-Negative Matrix Factorization (NMF) [4], linear discriminant analysis (LDA) [10], and so on. The growing quantity of data implies that all of those were widely used in text mining, recommender systems and other information retrieval tasks [6, 95, 110, 123],

2.2.4 Distance and Similarity Measures

We consider similarity to be a synonym of relatedness within this thesis, even though there might be a slightly difference between both terms because they tend to be used interchangeably in the literature. Beel et al. [13] defined the *similarity* as the number of features two items have in common, in contrary to *relatedness* which express how close two items are. As an example, they consider that a *paper* is not similar to a *pen* but they are related.

The simplest and most classical distance is the Euclidean one (Equation 2.3). The sum of the squared difference of all n features (i.e., coordinates) of two items (a and b) is calculated. Finally, the distance is the root square of this sum.

$$d_{Euc}(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (2.3)$$

Another commonly used metric is the cosine similarity, which computes the cosine of the angle that two vectors (a and b) form (Equation 2.4). In other words, the division of the vectors dot product (\cdot) by the product of their norms gives the cosine similarity.

$$d_{cos}(a, b) = \frac{a \cdot b}{\|a\| \|b\|} \quad (2.4)$$

A multitude of other distance or similarity metrics may be used by clustering algorithms to build their clusters. Among others, the Pearson correlation, Jaccard coefficient, the Mahalanobis, the Dice's coefficient and so on. Users willing to discover more distance metrics or more details on those might read the following surveys [6, 44, 48, 120].

2.2.5 Word Mover Distance (WMD)

Given that the word embeddings (Section 2.1.4) and similarity measures (Section 2.2.4) have been discussed, let us introduce the word mover distance (WMD) [63]. This similarity metric may be seen as an instance of Earth mover's distance (EMD) [106],

which estimates the minimal cost of a transportation from one set of elements to another one—*i.e.*, the minimal moves/changes that need to be applied to one set to perfectly match the other one.

WMD may be used to compute a distance among Word2Vec vectors in order to measure how dissimilar two items are. For that purpose, WMD takes vectors for all words of a document D_1 and computes the euclidean distance (*i.e.*, cost) with another document's (D_2) vectors—it builds the $n * m$ matrix where n and m are respectively the sizes of the unique words vectors from D_1 and D_2 . Then the distance between these two documents is nothing else than the sum of the minimum costs for every vector from D_1 to "travel" until its closest vector from D_2 .

2.2.6 Classification

Classification approaches tend to classify items into one of several predetermined and finite number of classes. They often rely on supervised machine learning processes from which models are trained to predict class belonging for unclassified documents. A huge amount of classifiers have already been implemented. The most commonly used are k-nearest neighbors (kNN), decision trees (DT), naïve Bayes (NB), support vector machines (SVM), neural networks.

Amatriain et al. [91] discussed about text mining approaches used in preliminary steps of recommender systems. Clear definitions of these most commonly used classifiers are given, providing an interesting global overview of their core methodologies. Menaka and Radha [77] identify keywords from tf-idf and obtain their senses by querying WordNet. After this disambiguation, the articles are classified into five different classes by three classifiers, namely kNN, NB and DT. The results obtained by the DT achieved the best results on their dataset.

A typical and complete classification example is presented by Romeo et al. [103]. The first steps realize the POS of each document and a WSD algorithm is applied. After that, the BoW of each documents is calculated and dimensionally reduced via the latent semantic indexing (see Section 2.2.3 for dimension reduction techniques). Then, a similarity graph is built using the cosine similarity metric, kNN computes the nearest neighbors and classes are assigned in a way that favor class likelihood.

Different various combinations can be adopted to build an entire classifier architecture. Indeed, some approaches may associate different information extraction tasks (n-gram, POS, words embeddings and so on), various data reduction techniques (SVM, LSI, LSA, LDA, etc) [99], and plug into these preliminary steps any classification algorithms (kNN, NB, DT, or even neural network classifiers [70, 129]). A big amount of research aroused in the classification field and this section does not aim to enumerate all possible approaches. However, we recommend interested users to read these surveys [1–3].

Having a finite number of classes may be a disadvantage of classification approaches, given that it implies to have a general *a priori* knowledge. Moreover, they may have the drawback of being hard to scale for similarity-based algorithms given that the distance between all pairs of documents need to be computed (*i.e.*, the entire similarity matrix) [91]. Clustering approaches can be put in place to tackle these problems. Those tend to be more efficient but the accuracy decreases.

2.2.7 Clustering

In contrary to classification, clustering is an unsupervised learning approach. Clustering algorithms automatically group items based on their similarity (or distance),

and therefore create as many groups as needed. Beel et al. [13] stated that clustering approaches are faster than other classification ones but those are less accurate. Amatriain [6] agreed but moderated this statement by saying that accuracy can only be improved when dimensionality reduction methods are used. Indeed clustering approach often consider BoW, use VSM as documents representative vectors and find closest items within this dimension space. The most commonly used clustering algorithm is the k-means [74], as stated in [47]. Steinbach et al. [119] made a comparison of k-means algorithm, one of its derived versions (bisecting K-means), as well as a few hierarchical clustering approaches. They concluded that k-means algorithms are faster while being as good as hierarchical approaches.

The k-means algorithm takes a projection of elements within a vector space and starts by randomly defining k items as centroids. Then each point is assigned to the closest cluster, based on the smallest euclidean distance between the point coordinate and the cluster average mean (i.e., centroid). Cluster's mean is re-calculated at each assignment as the mean of every cluster element and a new centroid is created. Several derived k-means algorithms have been implemented, such as bisecting k-means, fuzzy k-means, Fuzzy c-means and so on (see Jain's survey [52] for more details).

Lot of other clustering algorithms exist, such as—among others—k-medoids, density based, hierarchical, Bi-Section-k-means, Self Organizing Map, fuzzy k-means, Co-clustering, Fuzzy Clustering and so on [6, 47, 126]. All of these approaches are based on a different criterion to judge about the pairwise similarity between documents.

Some clustering approaches might have a few particularities when applied to big data [118], or short text [99] in order to counter respectively the data abundance or the lack of context. However, Biemann [18] stated that theoretically, every clustering method can be applied to any representation (VSM or other matrices). For more details about clustering algorithms, interested users are recommended to have a look at these well written surveys [3, 6, 16, 53, 126].

2.2.8 Categorization

Even though categorization and classification are often interchangeably used in the scientific literature (e.g., [112, 127]), they might be considered as slightly different fields [54]. Indeed, the categorization is seen as the task of extracting categories from text, either by identifying most representative keywords or relying on thesaurus / ontology. Therefore, it can be associated to labeling, where categories (i.e., word or group of words) representing the most a document are selected. We differentiate categorization with classification and clustering because no learning phase is needed, hence it can be seen as an unsupervised approach without necessarily any machine learning algorithm behind it. Indeed, neural classifiers would be hard to apply in categorization approaches given that there would be too many potential categories and the network would not have enough representative ground truth entries to converge into an efficient model.

Janik and Kochut [54] “correctly” used the term *categorization* for their approach realizing an ontology-based entity recognition, from which all potential senses of a sentence terms can be printed into their *semantic graphs*. Then relationships among senses can be identified and the most connected topic (i.e., category) is selected as the most representative of the document / paragraph. Other articles also used the term categorization in the same sense we are attending [36, 42, 90].

2.3 Recommender Systems

Web users are used to get recommendations, relatively targeted, from any website they browse, recommending what to read, listen, watch, buy, follow and so on. From the Amazon's famous "*Customers who bought this item also bought*" to the Netflix recommendations, RS assist users to take better decisions faster. RS became so profitable that Netflix even launched a challenge and offered \$1,000,000 to the person who was able to outperform by 10% their recommendations algorithm on their dataset containing 100 millions anonymous movie ratings. This might be justified by the fact that 60% of Netflix DVD renting revenues are coming from suggestions [64].

Overall, recommender systems (RS) aim to bring relevant items to their users. Many good and complete surveys have been written on this topic [9, 13, 21, 64]. Different types of RS have been developed, and were differently grouped in the literature, depending on the main interest of the authors writing those. Indeed, some surveys grouped RS by the algorithm or metric(s) used to compute similarity whereas others rather focus on the type of data used to compute recommendations, or by the field of applications. We decided to group them more generally and only consider two main groups of RS, namely the *content-based filtering (CBF)* [13, 73, 92] and the *collaborative filtering (CF)* [31, 111, 121]. From those, a third group combining both CBF and CF approaches (*hybrid* approach) may be derived in order to tackle their respective weaknesses. Within these groups, more sub-categories might be identified, such as item-based, user-based, graph-based, co-occurrences approaches among others. The aim of this section is not to be exhaustive and describe all details from experimental/future/obsolete/marginal implementations of RS but rather to provide a global understanding of the common ways to recommend items.

2.3.1 Content-Based Filtering

A content-based filtering RS (CBF) aims to propose to its users items close to items that he used to like or access. In other words, item-item similarity are computed and items close to the user's interests are proposed. The classical workflow of such RS combines an item representation, optionally a user profile and finally a ranking algorithm to propose the most relevant items. Many of the content-based techniques are directly inherited from the IR field and both fields are therefore really close [78]. Actually, CBF RS can be seen as an advanced classification tasks [73, 78]. Consequently, tf-idf (see Section 2.2.1) is widely used to model textual features occurrences. It tends to favor terms that occur frequently in one document (TF: term frequency), but which remain rare in the rest of the corpus (IDF: inverse-document-frequency) [73]. This term weighting representation approaches are usually combined with Vector Space Model (see Section 2.2.2) to build items and users profiles. Other approaches providing bag-of-words (or b-o-term/b-o-concept) per document matrices—*such as the latent semantic indexing (LSI) or latent Dirichlet allocation (LDA)*—may be used to represent documents or users in a vector space. On top of these vector space representations, dimension reduction techniques (Section 2.2.3) are often used to reduce the matrix size and fasten the similarity computation. Indeed, the ranking algorithm usually relies on a similarity metrics, such as the cosine similarity which measures the angle between user/item or item/item vectors, from the VSM. Other ranking distance metrics might be used such as Pearson, Jaccard index or Euclidean distance (see Section 2.2.4 for more similarity or distance metrics).

CBF approaches have been widely applied to scientific literature [32, 55, 71, 85, 102]. Beel et al. [13] stated that they were the most used for scientific RS (55% of the

62 reviewed approaches). The main advantages of CBF approaches is their transparency, given that connectivity criterion (i.e., common features) can easily be identified and displayed to the user in order to give him the choice whether he/she decides to follow the recommendation or not.

2.3.2 Collaborative Filtering

Collaborative filtering RS (CF) recommend to a user items that similar users liked. In background, user-user or user-item matrix is often built and used to find link-minded users, hence potential items of interest. The main advantage of these approaches compared to CBF ones is that they do not need data analysis and can therefore be applied similarly to system recommending movies to watch than for systems suggesting items to buy. Indeed, given the fact that recommendations are based on user activities (i.e., ratings, readings and so on), CF mainly relies on user similarity matrices. Those are used to find link-minded users (i.e., users who shared similar activities) in order to predict items a user U may potentially like.

The same similarity measures as CBF might be used to identify link-minded users such as Pearson correlation coefficient or cosine similarity. Also k-nearest neighbors (kNN) or other classification approaches (see Section 2.2.6) might be used to identify closest or group of similar users. CF user-based approaches tend to provide good results but do not scale well and suffer from the lack of data (sparsity and cold-start problem). The main reason of this drawback is the huge number of potential users' ratings, which leads to enormous matrices. To tackle the performance speed of huge matrices processing, several dimensionality reduction techniques are applied, such as SVD, PCA, and so on (see Section 2.2.3 for more techniques) [57, 110]. Koren and Bell [61] were also innovative to avoid sparsity related problems given that they implemented a method interpolated unobserved ratings by averaging ratings done on similar items rated by the user.

CF approaches were also applied to scientific RS [76, 83, 93], but much less than CBF ones—only 18% VS. 55%—according to [13]. They explained this phenomena with the fact that CF RS suffers too much both from the cold start problem (i.e., no data when launching the algorithm, or on insertion of new entry into the system) and from the low participation motivation (i.e., users' ratings).

2.4 Conclusion

This chapter presented an introduction and a general overview of the main text mining (TM) tasks. Main knowledges of the information extraction (IE), as well as the information retrieval (IR) fields have been introduced, going through some of the mostly used approaches to extract information (bag-of-words, n-grams, part-of-speech, etc) to some of the most classical data representation techniques for efficient IR (vector space model, dimension reduction technique, similarity measures, etc). We also introduced two of the most used recommender systems (namely content-based and collaborative filtering ones) which embrace tasks both from IE and IR.

Chapter 3

Categorization

3.1 Our Approach

The first contribution of this thesis is the categorization of scientific articles. This part would later offer the possibility to meaningfully group articles (e.g., by categories) in contrary to the previous situation where belonging to a journal or publisher were the only unifying elements. Word sense disambiguation (see Section 2.1.6) naturally becomes a preliminary pre-processing step to achieve this goal while extracting a category is hard (if not impossible) from ambiguous set of words. For that purpose, a knowledge database (BabelNet [88]) is used and our approach categorizes scientific articles by identifying the categories shared by the keywords.

3.1.1 BabelNet

The multilingual lexicographic and encyclopedic database BabelNet [88] is a smart superposition of semantic lexicons (WordNet, VerbNet) and other collaborative databases (Wikipedia and other Wiki data). The embeddings of all standard knowledge databases commonly used for classical WSD approaches makes it a complete and really good candidate to provide the needed information to disambiguate keywords. BabelNet generously provides offline indexes and APIs to query those, freely for research purpose, which makes it fast and really efficient for different applications.

The research community widely validated the suitability of BabelNet data in every aspect of text mining. Extraction of the most meaningful data from scientific articles [37, 104], summarizing documents [101], language detection [116] and WSD for multilingual document classification [103] are illustrating a little panel of its different applicative usage domains.

We used BabelNet mostly as a dictionary where searching for a word provides a list of entries with different senses and meanings. Figure 3.1 shows the architecture of our approach where the different meanings (called synsets in BabelNet) can belong to one or several categories and domains. Domains are considered as the higher entity in the knowledge graph—also called *ontology*—and are therefore more general than categories. Actually, there are only 34 general domains (e.g., ‘health and medicine’ or ‘physics and astronomy’) in contrary to categories which are mostly inherited from Wikipedia—*Wikipedia contains around 290,000 categories* [90]. The diversification of these categories may be very precise such as ‘*peripheral nervous system disorders*’ or ‘*exact solutions in general relativity*’ while others may be really general like ‘*technology*’ or ‘*knowledge*’. This is probably the consequence of the collaborative maintaining of Wikipedia hierarchy, where users can add / edit / delete categories. Some users might be experts on a field and add very specific categories where other users will add more general categories embracing for example the domain of application.

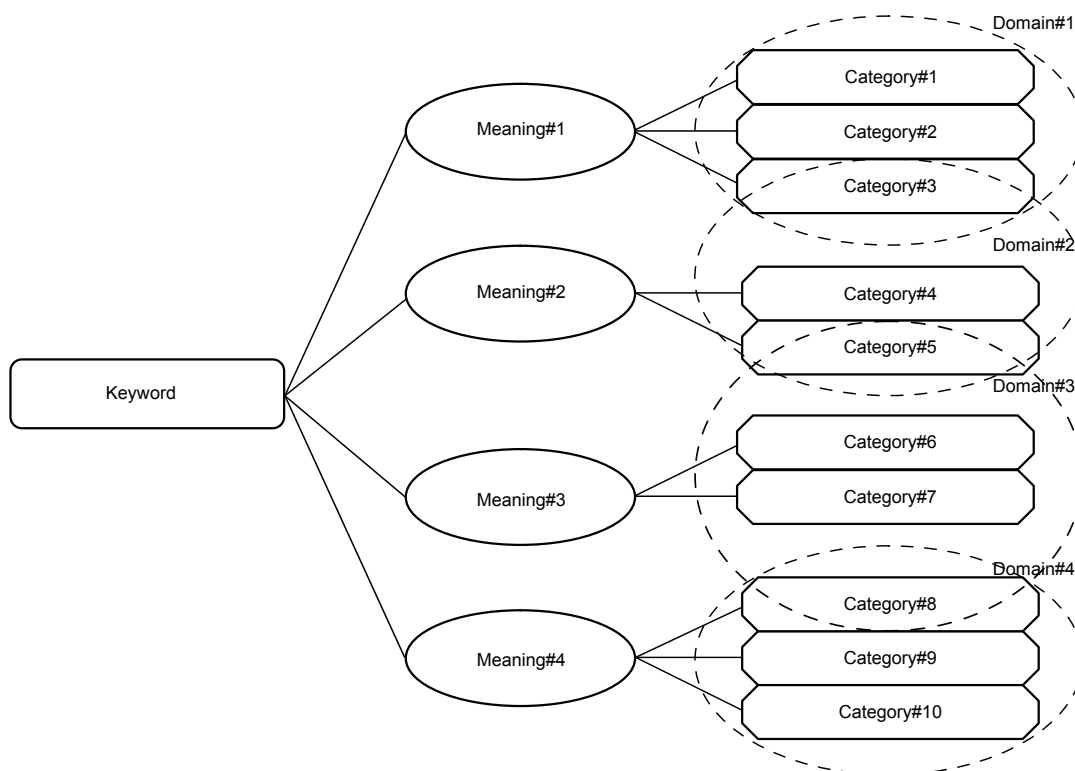


FIGURE 3.1: BabelNet dictionary architecture.

3.1.2 General Overview

In order to categorize an article, we first need to disambiguate its representative words. We made the choice to focus on keywords both because they are legitimate (see Section 3.1.3) and for a scalability reason (it is faster to disambiguate keywords than the abstract or other parts of an article).

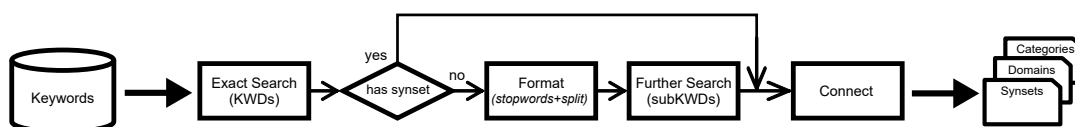


FIGURE 3.2: Simplified workflow of our categorization approach.

As illustrated in Figure 3.2, our approach starts by searching BabelNet synsets for origin keywords, without any pre-processing (exact search – Section 3.1.4). Another step (further search – Section 3.1.5) is activated for those returning no data, where keywords are split and different combinations of words are tested to extract potential BabelNet synsets. Finally, connecting synsets from the different steps by their categories realises the disambiguation part (Section 3.1.6) and returns the disambiguated synsets together with their corresponding categories and domains.

3.1.3 Keyword Usage

As discussed in the previous section, our approach uses only keywords as representative words to categorize articles. Titles, abstracts and other articles' metadata might potentially also be an interesting source of information, but we assumed that keywords are the most representative words of the articles. Shah et al. [114] investigated about the legitimacy to only use abstracts to generate key terms of biological

scientific articles and concluded that abstracts contain the best ratio of key terms per total of words. Gil-Leiva and Alonso-Arroyo [38] analysed the scientific literature in several articles and stated that keywords provided by authors are a very meaningful source of information. Therefore, this confirm the legitimacy of using keywords from our dataset given that they are either from the authors or generated by a topic extractor (MAUI [59]) from the concatenation of article’s title and abstract. Finally, we define \mathcal{A}_K as a set of n keywords from an article \mathcal{A} :

$$\mathcal{A}_K = \{k_1, \dots, k_n\} \quad (3.1)$$

3.1.4 Exact Search

The exact search is the first way of finding potential synsets from BabelNet. This is the naive approach where requests are sent without any pre-formatting. This provides precise synsets because the keyword’s sense given by the author is never altered. However, the difficulty of this step is finding BabelNet entries for multiwords keywords, which are highly represented in the scientific literature. Indeed, the longer the keyword (i.e., the more words composing it), the lower the chance that it has a related BabelNet entry. Synsets inherited from these searches bring satisfactory results (95% precision and 93% recall) while used for the categories connections, but converges for only 22% of the articles (see Section 3.2.1 for detailed results).

3.1.5 Further Search: Split

To counter the exact search drawback which is its difficulty to extract synsets from multiwords keywords, the further search is activated when no results is obtained. Hence, stopwords¹ and isolated numbers² are removed from these problematic keywords. Then, the remaining keywords are split on spaces and punctuation marks. This exploding phase provides new potential sub-keywords to search against BabelNet.

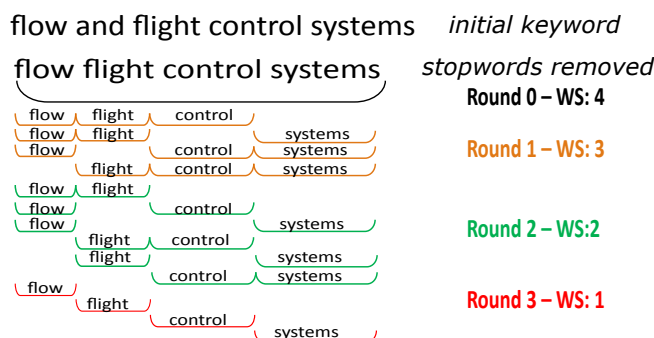


FIGURE 3.3: Split phase: Decreasing n-gram generated after stopwords removal.

Figure 3.3 shows the split logic. The stopword *and* is removed from the initial keyword in the pre-processing step. Then, our enhanced decreasing n-gram approach generates all possible word combinations. Classical n-grams ‘*flow flight control*’ and ‘*flight control systems*’ are extracted, but also the linear combinations ‘*flow*’

¹Stopwords are common words frequently used in a language (e.g., the, of, and, etc.) and which are therefore useless in WSD.

²Numbers contained within a word are conserved, such as for the molecule H_2O

flight systems' and *'flow control systems'*. This enhancement relies on the skip-gram model [41], where some tokens (i.e., words) may be skipped in the n-gram generation. These linear combinations may create new sub-keywords unrelated to the context of the original keyword (especially for long keywords), but our experiments show that it improves results. It is especially useful to find potential matchings for keywords containing several adjectives or nouns.

Our approach starts from the largest possible window (original keyword after stopwords removal – *Round 0*) and requests BabelNet. The window size (WS) is decreased as long as no result is obtained, at the end of each round. A round is never interrupted, even if synsets are found from a sub-keyword because other sub-keywords sharing the same size might also provide other synsets. In this example, the 4-grams and 3-grams generated respectively by the Round 0 et Round 1 do not bring any result from BabelNet. However, the 2-grams *flow control*, *flight control*, and *control systems* from the Round 2 respectively return 2, 3, and 1 synsets. Precision significantly decreases when the splitting process goes up to the 1-grams explosion because single words are more ambiguous and therefore bring more potential unrelated synsets (i.e., noise). In this example, BabelNet would respectively return 42, 38, 56 and 3 synsets while searching for the words *flow*, *flight*, *control* and *systems*. This is the reason why the split process is stopped at the end of a round as soon as some synsets are retrieved. Finally, the 6 synsets found in the Round 2 are kept as potential representative entries of the initial keyword and will be used for the disambiguation (connection phase – Section 3.1.6).

The default behavior of this step is that only synsets from multi-words sub-keywords are kept in order to be as precise as possible. However, if categories (or domains) from single-words sub-keywords are connected among the same keywords, their corresponding data is used for the keywords categories connection. A few options were implemented in order to bring more flexibility and include synsets from single-words searches (see Section 3.3).

3.1.6 Synsets Connections

Because of the architecture of BabelNet and the richness of its results, there is a need to filter out unrelated synsets returned by both the exact search (i.e., initial keywords) and the further search (i.e., sub-keywords). This section describes the connection of potential synsets which is the most important step of our categorization process. Indeed, given that searching for a term in BabelNet often returns a list of synsets having different senses, there is a need to find the context of the term usage. Hence, the selection of the synsets matching the article context is realized by identifying the common categories shared by synsets from different keywords. Thus, a connection represents the link between two synsets coming from two keywords sharing a common category. To illustrate the need of this disambiguation step, let us take the example of the keyword *Gold*. It can be used in scientific literature to express the precious gold metal³[26], but also as the atom⁴[30], the nanoparticle⁵[25] or even gold coins⁶[33]. In other cases, *Gold* may also be used as an adjective to express the color, favored topic (e.g., the golden age/past), etc. and all of these different meanings have their corresponding synsets in BabelNet.

³<https://tinyurl.com/babelMetal>

⁴<https://tinyurl.com/babelAtom>

⁵<https://tinyurl.com/babelNanoparticle>

⁶<https://tinyurl.com/babelCoins>

To disambiguate the potentially big amount of synsets returned —*Recall: a synset can be considered as a dictionary entry*— and select the meaningful sense in regards to the article’s context, synsets categories are exploited. We assume that the more synsets share the same categories, the more probable they are legitimate. For that purpose, we define the function S returning for each keyword k its m corresponding BabelNet’s synsets:

$$S(k) = \{s_1, \dots, s_m\} \quad (3.2)$$

A synset s contains its related categories C (noted $s.C$), domains D (noted $s.D$), and a set of neighbor synsets N (noted $s.N$) within the specific concept. We express it as follow:

$$s = \{C, D, N\} \quad (3.3)$$

For the purpose of disambiguating words in order to categorize articles, the categories and domains are the most interesting information from this synset data. However, as discussed in Section 3.1.1 domains are too general and too many connections would be extracted. Hence article’s categories A_C are inherited from the extraction of common categories across keywords’ synsets. The following equation shows the categories selection in its simplified version, when synsets are only coming from the exact search:

$$\begin{aligned} \text{Let } k_i, k_j \in \mathcal{A}_K, s_i \in S(k_i), s_j \in S(k_j) \\ \mathcal{A}_C = \{c \mid c \in s_i.C \wedge c \in s_j.C \wedge s_i \neq s_j \wedge k_i \neq k_j\} \end{aligned} \quad (3.4)$$

Equation (3.4) implicitly defines that only categories shared by at least two synsets are kept. Indeed, the automatic validation can not be ensured in case of unique category occurrences because no confidence can be expressed. There are in reality several different and sophisticated cases for the selection and the restriction of categories. The six most representative cases are developed in the following sections:

- Connection—*common categories*
- No connection—*no common categories*
- No connection—*synsets from the same keyword*
- No connection—*synsets from the same sub-keywords*
- Connection—*common categories from different sub-keywords*
- No connection—*same synset*

3.1.6.1 Connection matrices

The synsets to be connected are coming both from the exact search (Section 3.1.4) or from the further search (Section 3.1.5). Hence, originally distinct keywords may lead to same sub-keywords after the splitting phase. Also two synsets from the same keywords may share the same category. To avoid selecting categories over represented within one duplicated synset, keyword or sub-keyword, three different matrices were created, namely categories/keywords (\mathcal{M}_{kwd}), categories/sub-keywords (\mathcal{M}_{sub}), and categories/synsets (\mathcal{M}_{syn}).

Let’s give an example of the matrix \mathcal{M}_{kwd} used for the categories selection, which represents the number of occurrences of a given category for a specific keyword.

Example of connections

$$\mathcal{M}_{kwd} = \begin{matrix} & k_1 & k_2 & \dots & k_n & \\ \begin{bmatrix} c_1k_1 & c_1k_2 & \dots & c_1k_n \\ c_2k_1 & c_2k_2 & \dots & c_2k_n \\ \vdots & \vdots & \dots & \vdots \\ c_mk_1 & c_mk_2 & \dots & c_mk_n \end{bmatrix} & c_1 \\ & c_2 \\ & \vdots \\ & c_m \end{matrix}$$

This matrix counts the occurrences of all m categories inherited from all synsets of the n article's keywords. It provides the flexibility to automatically determine which categories (i.e., rows) are shared by several keywords (i.e. columns). The same is done for categories by synsets and sub-keywords matrices. When some synsets are found with the exact search, the keyword is not split. To identify valid connections, the synset is replicated to the sub-keyword matrix. Hence, $\mathcal{M}_{kwd} = \mathcal{M}_{syn}$ when all synsets are found exclusively with the exact search. Finally, only categories occurring in multiple rows of each matrix are selected. For that purpose, the function $F(\mathcal{M})$ computes the vector indicating whether or not each category is enough represented in the matrix \mathcal{M} .

$$F(\mathcal{M}) = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \quad (3.5)$$

$$\text{where } \alpha_i = \text{norm} \left(\sum_{j=1}^m \text{norm}(\mathcal{M}_{ij}) \right)$$

$$\text{and } \text{norm}(x) = \begin{cases} 1, & \text{if } x > 0. \\ 0, & \text{otherwise.} \end{cases}$$

After the computation of the category occurrence vectors for all matrices (i.e., with the function F), the Hadamard product⁷ of these vectors gives the selected article's categories \mathcal{A}_C . This entrywise product ensures that categories are only selected as soon as they are not over represented among a unique item (i.e., keywords, synsets or sub-keywords).

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{matrix} \quad (3.6)$$

⁷The Hadamard product is the entrywise product, i.e., the element-for-element product of both matrices.

Theoretical example – connection

$$\mathcal{M}_{kwd} = \begin{array}{ccc|c} & k_1 & k_2 & k_3 & \\ \hline & \mathbf{1} & \mathbf{0} & \mathbf{1} & c_1 \\ & 2 & 0 & 0 & c_2 \\ & \mathbf{0} & \mathbf{1} & \mathbf{1} & c_3 \end{array} \quad \mathcal{M}_{syn} = \begin{array}{ccccc|c} & s_1 & s_2 & s_3 & s_4 & s_5 & \\ \hline & 2 & 0 & 0 & 0 & 0 & c_1 \\ & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & c_2 \\ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & c_3 \end{array}$$

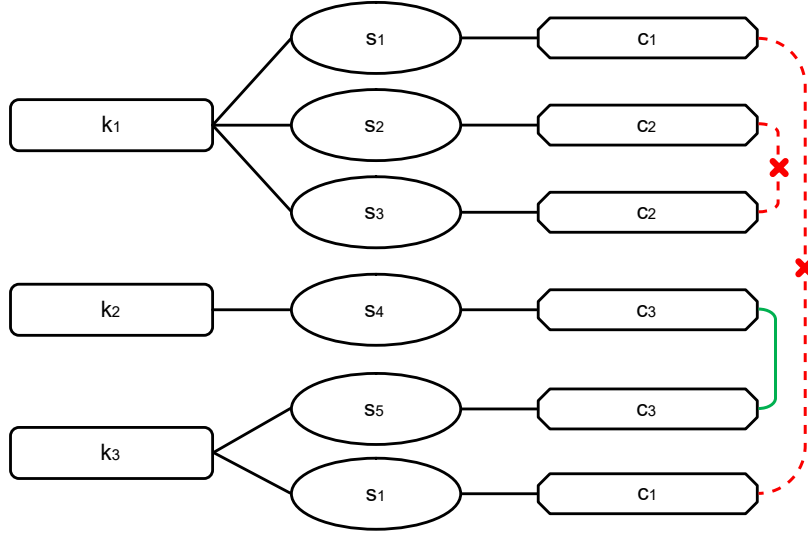


FIGURE 3.4: Theoretical representation of a category connection.

In this example, all results are coming from the exact search. The category c_3 is highlighted because it comes from two different synsets (s_4 and s_5 from \mathcal{M}_{syn}) and from two different keywords (k_2 and k_3 from \mathcal{M}_{kwd}). The categories c_1 and c_2 are not highlighted because c_2 appears from only one keyword (k_1) and c_1 from only one synset (s_1). As discussed previously, the categories/sub-keywords matrix (\mathcal{M}_{sub}) contains keywords categories when the split logic is not used. Hence, $\mathcal{M}_{sub} \leftarrow \mathcal{M}_{kwd}$ in this specific case, because exact search finds synsets for all keywords. We obtain the following respective category occurrences matrices:

$$F(\mathcal{M}_{kwd}) = F(\mathcal{M}_{sub}) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} \quad F(\mathcal{M}_{syn}) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}$$

The final article's categories are selected by the following Hadamard product:

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}$$

In other words, $\mathcal{A}_C = [c_3]$, which is the desired result given that s_4 (from k_2) and s_5 (from k_3) are connected by their common category c_3 . The category c_2 is represented in two different synsets (s_2 and s_3) and is therefore selected by $F(\mathcal{M}_{syn})$, but they both come from the same keyword k_1 so it is neutralized by the Hadamard product with $F(\mathcal{M}_{kwd})$. The same filter is applied for the category c_1 , even though it occurs in keywords k_1 and k_3 , given that it is coming from the same synset s_1 . Hence, the value

0 is assigned to c_1 in $F(\mathcal{M}_{syn})$ and the category is not kept as one of the article categories. This filter is needed in order not to boost over represented categories within a single keyword—*several synsets sharing the same category withing the same keyword does not mean that this has more chance to be the most representative of the given keyword*—or over represented synsets, for the same reason.

Theoretical example – no connection

$$\mathcal{M}_{kwd} = \begin{array}{ccc|c} & k_1 & k_2 & k_3 \\ \hline c_1 & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ c_2 & 0 & 1 & 0 \\ c_3 & 0 & 0 & 1 \end{array} \quad \mathcal{M}_{syn} = \begin{array}{ccc|c} & s_1 & s_2 & s_3 \\ \hline c_1 & 2 & 0 & 0 \\ c_2 & 0 & 1 & 0 \\ c_3 & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{array}$$

In this example, no category is selected, and the split method is still not used. Even though the category c_1 properly comes from two different keywords (k_1 and k_3 from \mathcal{M}_{kwd}), it comes from the same synset because no synset share any category (visible in \mathcal{M}_{syn}). Hence, Hadamard product of category occurrences matrices return an empty zero matrix.

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}$$

Therefore, no category will be proposed for this article, and $\mathcal{A}_C = []$. This is the expected role of this connection process to filter out unsafe categories when they are not coming from distinct keywords, sub-keywords and synsets.

3.1.6.2 Connection—common categories

The first and simplest way to disambiguate keywords is when several synsets from different keywords share the same category(-ies). The Figure 3.6 is a perfect illustration of these connections where the keywords *HIV* and *AIDS* have one common category *HIV/AIDS*. Other unconnected synsets are considered as local noise (i.e., unrelated) and are finally filtered out. However, they might be the correct synsets in another field.

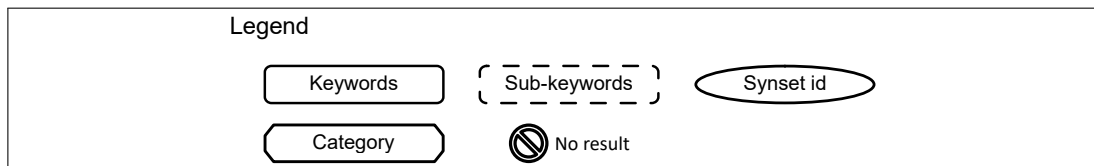


FIGURE 3.5: Legend.

Even though the sense of *HIV* is obvious for a human in this example, it remains ambiguous because BabelNet returns three different synsets when searching for *HIV*. The first entry does not have any category—*hence it will never be usable by our categorization approach*—but has the correct sense “*Infection by the human immunodeficiency virus*”. The second entry is the one we are searching for, with the sense “*The virus that causes AIDS; it replicates in and kills the helper T cells*”. Finally, the last one illustrates the diversity of BabelNet because it represents an iranian village and has

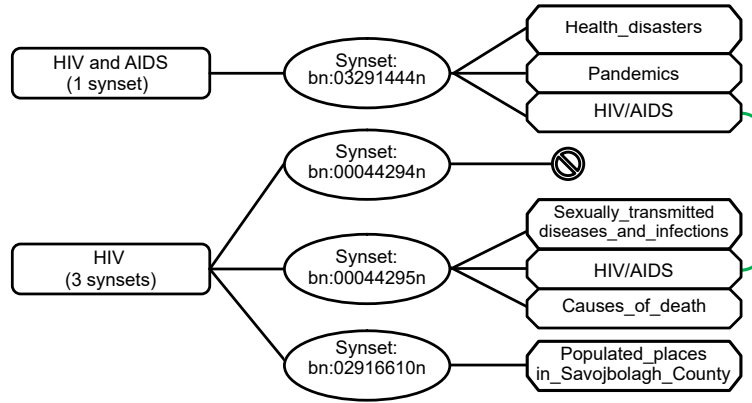


FIGURE 3.6: Simple connection—synsets sharing common categories are the meaningful entries.

the following sense “*Heev is a village in Heave Rural District, in the Central District of Savojbolagh County, Alborz Province, Iran*”.

In the example from Figure 3.6, only the categories/keywords (\mathcal{M}_{kwd}) and categories/synsets (\mathcal{M}_{syn}) matrices are used because there is no sub-keyword (i.e., $\mathcal{M}_{kwd} = \mathcal{M}_{sub}$).

$$\mathcal{M}_{kwd} = \begin{matrix} & \begin{matrix} AIDS & HIV \end{matrix} \\ \begin{bmatrix} 1 & - \\ 1 & - \\ \mathbf{1} & \mathbf{1} \\ - & 1 \\ - & 1 \\ - & 1 \end{bmatrix} & \begin{matrix} Health_disasters \\ Pandemics \\ HIV/AIDS \\ Sexually_transmitted.. \\ Causes_of_death \\ Populated_places.. \end{matrix} \end{matrix} \begin{matrix} s_1 & s_2 & s_3 & s_4 \\ \begin{bmatrix} 1 & - & - & - \\ 1 & - & - & - \\ \mathbf{1} & - & \mathbf{1} & - \\ - & - & 1 & - \\ - & - & 1 & - \\ - & - & - & 1 \end{bmatrix} & = \mathcal{M}_{syn} \end{matrix}$$

The category *HIV/AIDS* is properly selected as the representative category of these keywords because this is the only category having multiple occurrences in all matrices.

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} Health_disasters \\ Pandemics \\ HIV/AIDS \\ Sexually_transmitted.. \\ Causes_of_death \\ Populated_places.. \end{matrix}$$

3.1.6.3 No connection—no common categories

Sometimes, even when keywords are all within the same context, the potential synsets do not share any category in common. Consequently, the category connection fails to identify meaningful entries. Figure 3.7 shows an example where three related synsets from the medicinal field inherited from the keywords *Cancer*, *AIDS* and *Risk factor* do not have any common category whereas they might all in relation with health diseases. Therefore, respective matrices will be sparse with only scattered occurrences, such as the keywords/categories matrix:

$$\mathcal{M}_{kwd} = \begin{bmatrix} \text{Cancer} & \text{AIDS} & \text{Risk Factor} \\ 1 & - & - \\ 1 & - & - \\ \vdots & \vdots & \vdots \\ - & 1 & - \\ \vdots & \vdots & \vdots \\ - & - & 1 \end{bmatrix} \begin{array}{l} \text{Constellations} \\ \text{Oncology} \\ \vdots \\ \text{HIV/AIDS} \\ \vdots \\ \text{Finance} \end{array}$$

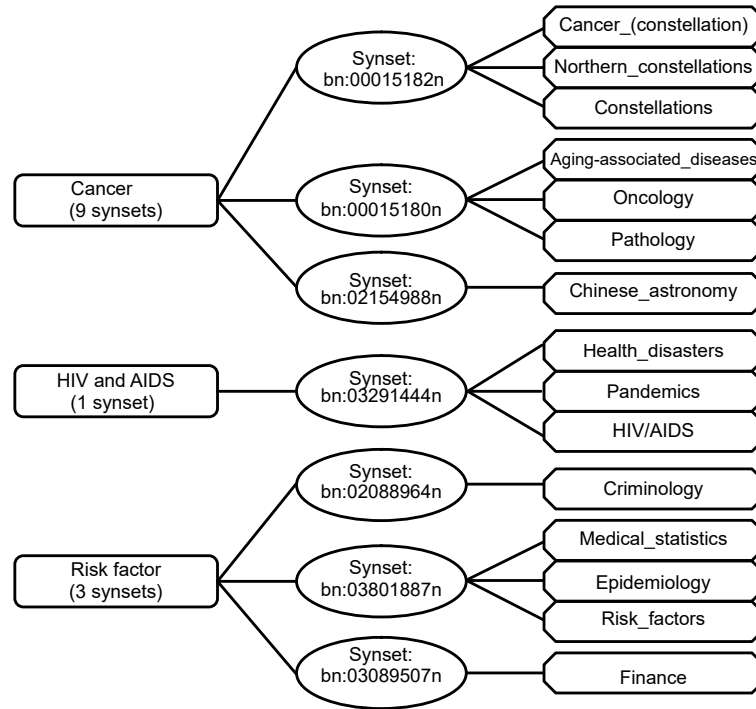


FIGURE 3.7: No connection—Categorization fails because related synsets do not share any category.

These situations might be the consequence of Wikipedia—*ingested by BabelNet*—where collaborative users may add categories without necessarily having advanced knowledge neither on Wikipedia nor in the domain of expertise. Indeed, an expert in plants might label an article treating about *Trifolieae*⁸ with its specific and narrowed category *Trifolieae*⁹ which is the most meaningful and precise category from his expertise, whereas a lambda user might have used the simple and broader category *Garden plants*¹⁰ which seems the best representative entry from his knowledge. Finally, all matrices are sparse and all categories occurrence matrices ($F(\mathcal{M})$) are zero matrices. Therefore, their Hadamard product is also a zero matrix:

$$\mathcal{A}_C = [0 \dots 0] \iff []$$

Even though it would have made sense if the medicinal synsets would have been connected, there is no confidence about which ones to select, given that keywords categories are not connected. Hence, this is preferable to miss some synsets than randomly proposing some, or proposing all possible ones. For example, in Figure 3.7,

⁸<https://en.wikipedia.org/wiki/Trifolieae>

⁹<https://en.wikipedia.org/wiki/Category:Trifolieae>
or <https://species.wikimedia.org/wiki/Trifolieae>

¹⁰https://en.wikipedia.org/wiki/Category:Garden_plants

there is no way to ensure that the synset about oncology is more accurate than the ones with the categories *Constellations* or *Chinese astronomy* for the keyword “Cancer”.

3.1.6.4 No connection—synsets from the same keyword

BabelNet may return, from the same search, several synsets belonging to the same categories. Grouping categories occurrences by their respective keywords is therefore a need to not select a category when it is over represented in one keyword synsets. Figure 3.8 is a good example where two synsets from the keyword *Gravity* belong to the category *Concepts in physics*. The top synset represents the physical attraction phenomenon whereas the bottom one is the entry for the law of gravitation.

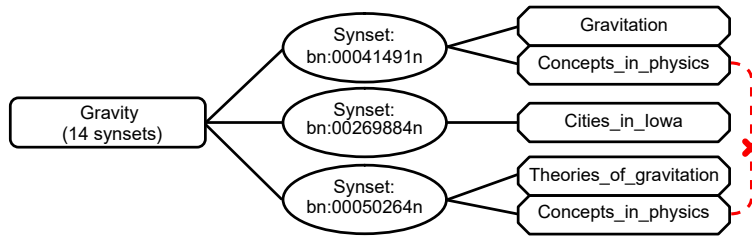


FIGURE 3.8: No connection—Filtering of common category shared by synsets from the same keyword.

Occurrence matrices are, for now, only used as presence flag representations and only the number of items (i.e., keywords, sub-keywords, synsets) is calculated for each category. However, the number of occurrences per categories could later give the possibility to weight the importance of a category, based on its frequency of occurrence. For the moment, boolean matrices are sufficient because the current goal is only to select categories with intersections, from both synsets, keywords and sub-keywords:

$$\mathcal{M}_{kwd} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix} \begin{matrix} Gravity \\ Gravitation \\ Concepts_in_physics \\ Cities_in_Iowa \\ Theories_of_gravitation \end{matrix} \begin{matrix} s_1 & s_2 & s_3 \\ \begin{bmatrix} 1 & - & - \\ \mathbf{1} & - & \mathbf{1} \\ - & 1 & - \\ - & - & 1 \end{bmatrix} \end{matrix} = \mathcal{M}_{syn}$$

Finally, even though the category *Concepts_in_physics* is coming from two synsets (s_1 and s_3 in \mathcal{M}_{syn}), it is only coming from the keyword *Gravity*. Therefore, no connection is established and no category is selected. The selection computation is detailed here-below (remind: $\mathcal{M}_{sub} = \mathcal{M}_{kwd}$ because no split is needed).

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} Gravitation \\ Concepts_in_physics \\ Cities_in_Iowa \\ Theories_of_gravitation \end{matrix}$$

Although there is a high probability that these synsets are the correct entry in a scientific context, there could be other connections within the other synsets of gravity as the ones from movies and other music albums. Therefore, again, it is better not to select synsets when no cross-keyword connection is identified.

3.1.6.5 No connection—synsets from the same sub-keywords

When original keywords have words in common, their splitting may lead to the creation of identical sub-keywords. The categories/sub-keywords matrix (\mathcal{M}_{sub}) helps to avoid the connection of categories inherited from these identical sub-keywords. The need of this third matrix becomes obvious with the example given in Figure 3.9.

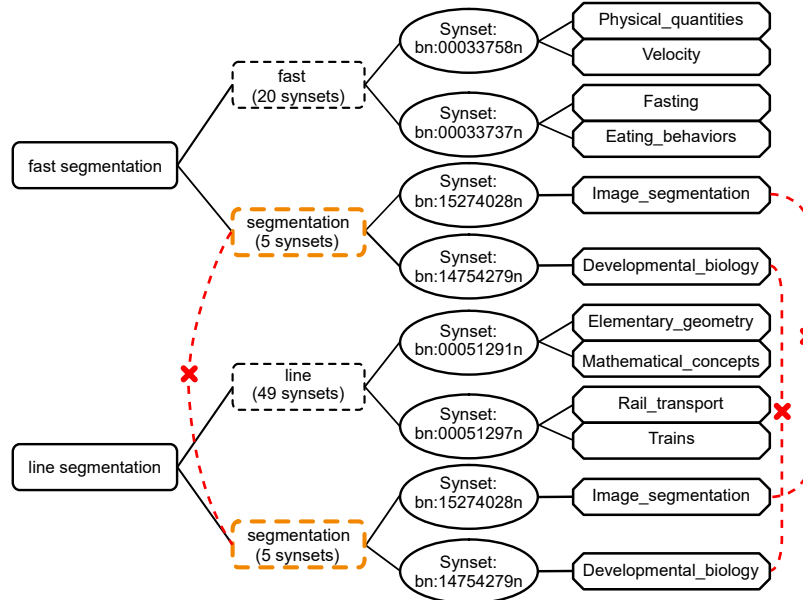


FIGURE 3.9: Sub-keywords—Filtering identical sub-keywords (*segmentation*). No connection, no category extracted.

Given that the five synsets resulting from *Segmentation* are coming from different keywords, the \mathcal{M}_{kwd} will have two entries for all of these synsets categories and the $F(\mathcal{M}_{kwd})$ will identify connections among keywords' categories. However, those will not be connected because they come from the same sub-keyword. Without this filtering, categories from all synsets will be activated and the keywords will remain ambiguous. The 3 matrices for this example are displayed here-below. The matrix \mathcal{M}_{syn} was contracted and condensed for a better visualization. In reality, it contains 74 synsets (i.e., columns), never more than 1 item per category (i.e., row), and each category for the *Segmentation's* synsets occurs twice because they are inherited from the same sub-keyword (i.e., they are in double).

$$\mathcal{M}_{kwd} = \begin{array}{cc|c} k_1 & k_2 & \\ \hline 1 & - & c_{sk_{11}} \\ \mathbf{1} & \mathbf{1} & c_{sk_{12},sk_{22}} \\ - & 1 & c_{sk_{21}} \end{array} \quad \mathcal{M}_{sub} = \begin{array}{ccc|c} sk_{11} & sk_{12,22} & sk_{21} & \\ \hline 1 & - & - & c_{sk_{11}} \\ - & 2 & - & c_{sk_{12},sk_{22}} \\ - & - & 1 & c_{sk_{21}} \end{array}$$

$$\mathcal{M}_{syn} = \begin{array}{ccc|c} s_{sk_{11}} & s_{sk_{12,22}} & s_{sk_{21}} & \\ \hline 20 & - & - & c_{sk_{11}} \\ - & 5 & - & c_{sk_{12},sk_{22}} \\ - & - & 49 & c_{sk_{21}} \end{array}$$

where :

- $k_1 = fast\ segmentation, k_2 = line\ segmentation$
- $sk_{11} = fast, sk_{12} = sk_{22} = segmentation, sk_{21} = line$
- c_{sk_x} : all categories from the sub-keyword sk_x
- s_{sk_x} : all synsets from the sub-keyword sk_x

Categories from the two *Segmentation* sub-keywords are not connected thanks to the filtering of the \mathcal{M}_{sub} matrix, but also from the \mathcal{M}_{syn} one. Indeed, given that the duplicated categories come from the same synsets (i.e., inherited from the same sub-keyword), the synset co-occurrence matrix (\mathcal{M}_{syn}) would have grouped them. Thus, they would have been neutralized. However, if the sub-keyword *Gravity—with two synsets sharing the same category* (see Figure 3.8)—would be in two different keywords, both synsets sharing the same category would be connected. Moreover, they would come from two different synsets from two different keywords. The usage of the \mathcal{M}_{sub} matrix is an extra filtering preventing these undesired connections, where two different synsets sharing the same category come from two identical sub-keywords of two different keywords. Hence, given that none of the categories is shared by two distinct sub-keywords in Figure 3.9, the categories occurrence matrix $F(\mathcal{M}_{sub})$ is a zero matrix. Consequently, the Hadamard product will also return a zero matrix, and none of the category will be selected (i.e., $\mathcal{A}_C = []$).

3.1.6.6 Connection—common categories from different sub-keywords

Even though connections from synsets coming from identical sub-keywords are filtered out, synsets are kept as potential candidates to connect with a category from another keyword, sub-keyword or even from sub-keywords coming from the same original keyword. To illustrate this, the keyword *Fast segmentation* (from Figure 3.9) is replaced by *Random Walker Segmentation*. Its combination with *Line segmentation* activates the category *Image_segmentation*, as shown in Figure 3.10.

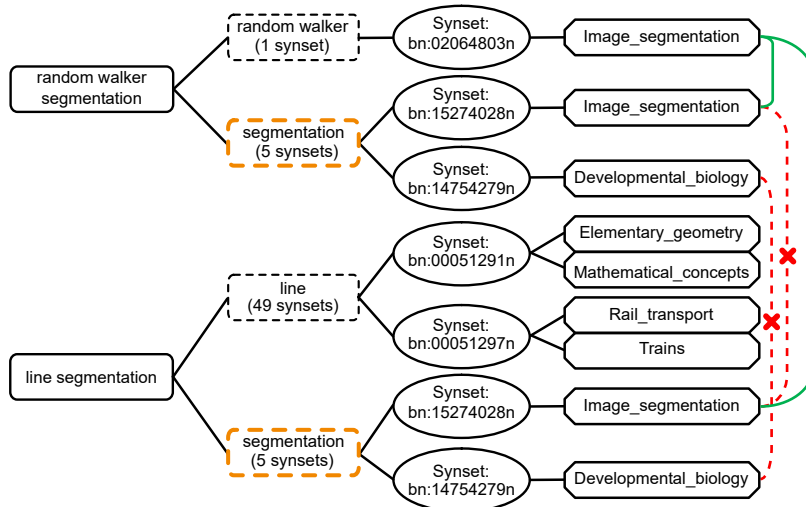


FIGURE 3.10: Sub-keywords—Filtering and connection between *segmentation* and another sub-keyword (*random walker*) from the same two keywords.

As stated above, synsets from sub-keywords remain potentially connectable to synsets from other keywords. Another illustration of this statement is when the keyword *Watershed* is added to the keywords *Fast segmentation* and *Line segmentation*, from which no connection was previously extracted (see Figure 3.9). The category *Image_segmentation* is now activated because one of the synsets from *Segmentation* share the same category as one of the *Watershed*'s synset (Figure 3.11). Therefore, the common category obtains a second synset from two different keywords, so the condition is verified and the category is considered as one of the article categories.

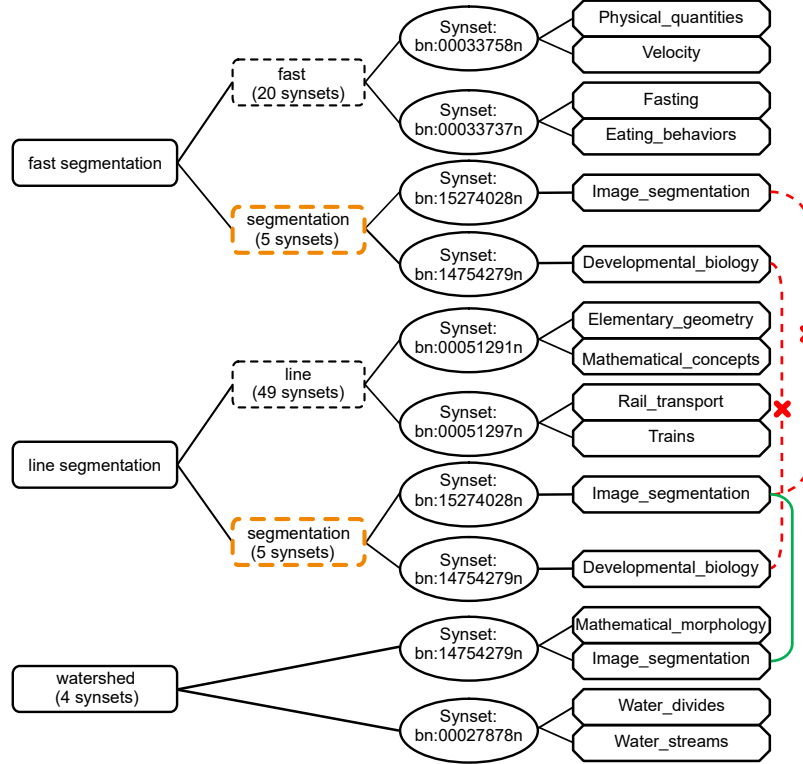


FIGURE 3.11: Sub-keywords—Filtering and connection between *segmentation* and another keyword *watershed*.

The role of the 3 matrices is represented in this example. We can observe that the category *Image segmentation* (C_{is}) is activated because it is coming from two different synsets, from two different keywords, and not from the same sub-keyword. Here again, matrices were compressed and categories were grouped for a better visualization.

$$\mathcal{M}_{kwd} = \begin{array}{ccccc} & k_1 & k_2 & k_3 & \\ \begin{bmatrix} - & 1 & - \\ \mathbf{1} & \mathbf{1} & - \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ - & 1 & - \\ - & - & 1 \end{bmatrix} & c_{fast} & c_{seg^*} & c_{is} & c_{line} & c_{wat^*} \end{array} \quad \mathcal{M}_{sub} = \begin{array}{ccccc} & sk_{11} & sk_{12,22^*} & sk_{21} & sk_{12,22^{is}} & k_3 & \\ \begin{bmatrix} 1 & - & - & - & - \\ - & 2 & - & - & - \\ - & - & - & \mathbf{2} & - \\ - & - & 1 & - & - \\ - & - & - & - & 1 \end{bmatrix} & c_{fast} & c_{seg^*} & c_{is} & c_{line} & c_{wat^*} \end{array}$$

$$\mathcal{M}_{syn} = \begin{array}{cccccc} & s_{11} & sk_{12,22^*} & sk_{21} & sk_{12,22^{is}} & k_{3^*} & k_{3^{is}} & \\ \begin{bmatrix} 1 & - & - & - & - & - \\ - & 2 & - & - & - & - \\ - & - & - & \mathbf{2} & - & \mathbf{1} \\ - & - & 1 & - & - & - \\ - & - & - & - & 1 & - \end{bmatrix} & c_{fast} & c_{seg^*} & c_{is} & c_{line} & c_{wat^*} \end{array}$$

where :

- $k_1 = fast\ segmentation$, $k_2 = line\ segmentation$, $k_3 = watershed$
- $sk_{11} = fast$, $sk_{12,22} = segmentation$, $sk_{21} = line$
- c_{xxx} : all categories from the keyword or sub-keyword xxx (e.g., c_{line} : all categories of *line*)
- s_{xxx} : all synsets from the keyword or sub-keyword xxx
- * = data minus entries from category *Image segmentation*
- is = only data with entries from category *Image segmentation*
- $c_{is} = Image\ segmentation$

The highlighted items from the matrices represent the categories that are selected in the categories occurrence matrices (i.e., $F(\mathcal{M})$). Finally, the Hadamard product returns *Image_segmentation* as the article’s category (\mathcal{A}_C).

$$\mathcal{A}_C = F(\mathcal{M}_{kwd}) \circ F(\mathcal{M}_{syn}) \circ F(\mathcal{M}_{sub}) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \begin{matrix} c_{fast} \\ c_{seg^*} \\ c_{is} \\ c_{line} \\ c_{wat^*} \end{matrix}$$

The last two examples proved the efficiency and the need of this extra filtering to avoid undesired connections among identical sub-keywords. However, it may sometimes lead to unexpected behaviour such as the example from Figure 3.12, where the category *Mathematical_concepts* coming from the sub-keywords *plane* and *line* is extracted. Even if legitimate—because the *plane segmentation of image is, at some degree, only a mathematical concept*—it is probably not the most representative one while the original keywords are about two different methods for the segmentation of images. It would probably make more sense if *Mathematical_concepts* was associated to *Image_segmentation*, however there is no confidence for the latest one. This behaviour is an undesired scenario directly originated from the split logic and from the connection logic, showing the importance of the keywords quality.

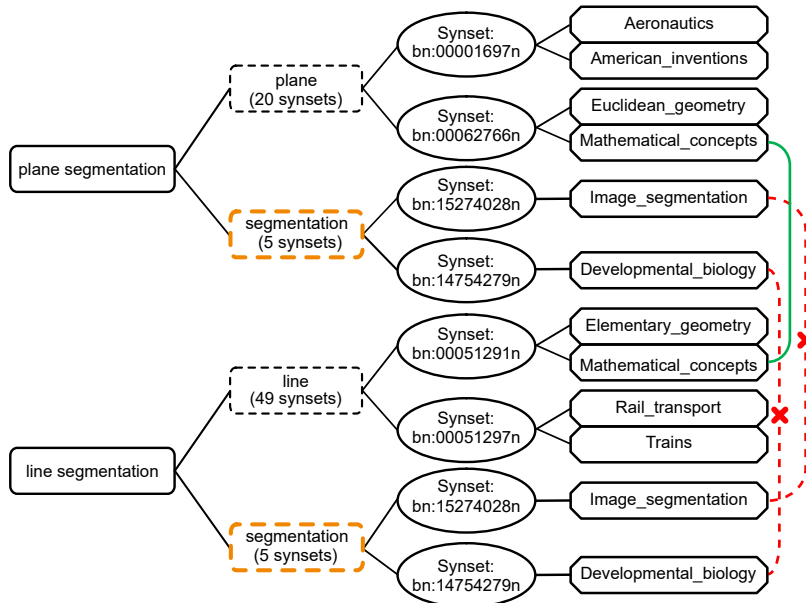


FIGURE 3.12: Sub-keywords—Filtering and connection among other sub-keywords (*plane* and *line*) from the same two keywords. The correct but not most representative category *Mathematical_concepts* is extracted.

3.1.6.7 No connection—same synset

The goal of the categories/synset matrix (\mathcal{M}_{syn}) is to not activate categories when the same synset comes from different keywords or sub-keywords. Indeed, BabelNet can return the same synset when requested with singular/plural variations (e.g., phosphate and phosphates), for an acronym and its original text (e.g, chronic obstructive pulmonary disease and COPD), or for words variants (e.g., radiation and

radioactivity) which are grouped within the same entry. The Figure 3.13 shows this filtering with an example in which the keywords *Chronic obstructive pulmonary disease* and *COPD* return the same synset (thick orange line in the Figure 3.13). Categories from this common synset are not linked together, and no category will be activated if there would not be another keyword (i.e., *Asthma*) sharing the category *Chronic_lower_respiratory_diseases* in common with the duplicated synset.

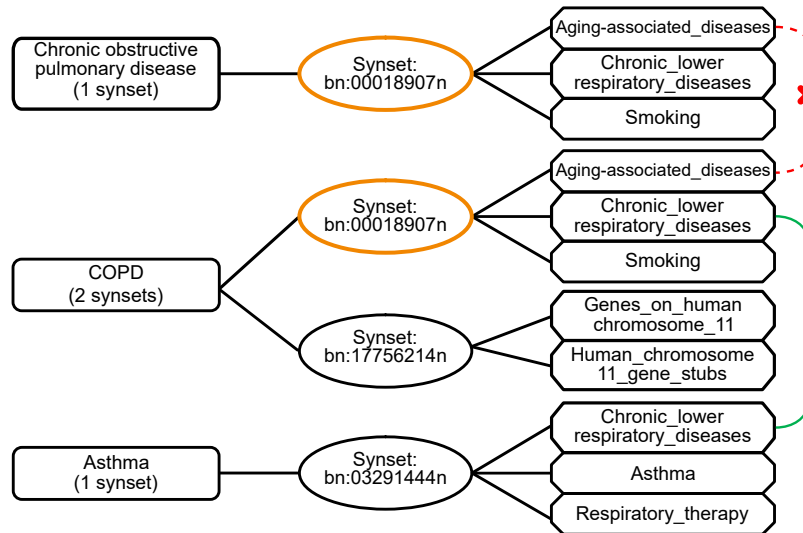


FIGURE 3.13: No connection—Filtering of identical synsets from two different keywords/sub-keywords, and connection with categories from another keyword.

All of these connections examples demonstrate that the proposed approach always ensures that a selected synset is cross-validated by two distinct synsets inherited from different keywords and sub-keywords. This connection logic avoids to select unrelated synsets, either not isolated (i.e., not connected) over represented (i.e., from the same keywords/sub-keywords).

3.1.7 Noise Filtering

As shown in the previous sections, unrelated synsets—*directly inherited from BabelNet richness while searching for keywords entries*—are naturally filtered out when disambiguated words share common categories. However, the richness and broadness of this knowledge database sometimes embraces too much data and brings too many potential synsets, from which several connections are found.

Figure 3.14 illustrates an example of three keywords (i.e., *Nonlocal gravity*, *Celestial mechanics* and *Dark matter*) from the domain of Physics. Unexpected connections from synsets coming from american films—*Gravity* (2013) and *Dark matter* (2007)— or from living people—*Gza* (a rapper with his studio album "*Dark Matter*") and *Danny_Boyle* (who seems to be wrongly attached to "*Gravity*" film¹¹)— are extracted. Also, the categories "*Celestial mechanics*, *American films*, *English-language films*, *Living people*" are connected from seven different synsets: One from "*celestial mechanics*", three from "*dark matter*", and three from "*gravity*".

Given that our approach is applied to the scientific literature, some undesired categories have been identified as constant noise. A parameter was therefore put in place to define static noise (e.g., "**_rock_groups*", "**_actors*", "**_rappers*", "**_films*", etc).

¹¹<https://tinyurl.com/babelBoyle>

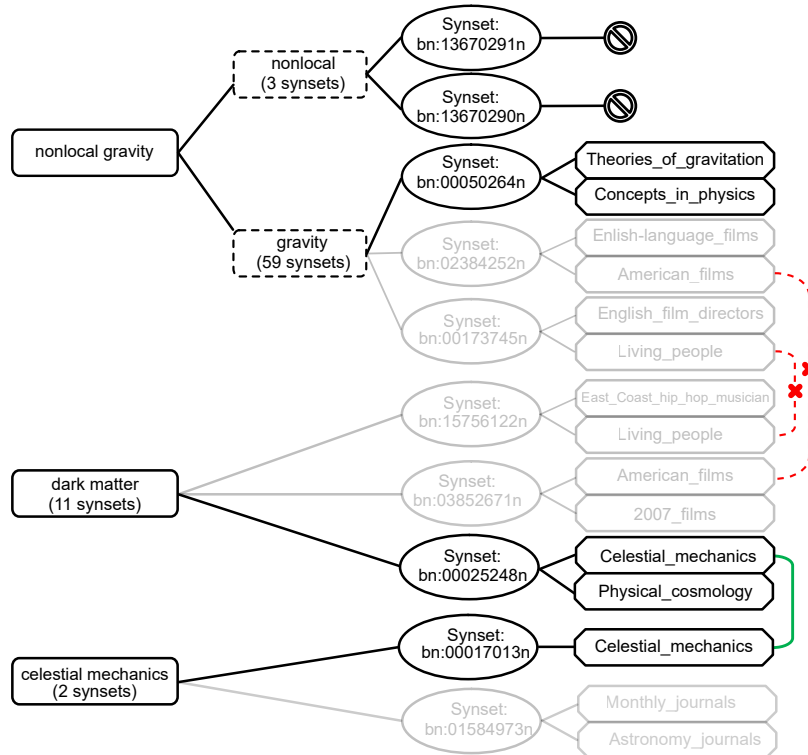


FIGURE 3.14: Noise filtering. Extra constant noisy categories (in gray) are de-activated.

Indeed, there is only a very small chance that scientific articles treat about specific singers, films or other unrelated topics as their main category. Hence these non scientific categories are considered as meaningless and synsets containing one of those are automatically neutralized before category connection. Finally, undesired and unrelated categories are filtered out and only the expected category "Celestial mechanics" is activated.

We plan to later use a dynamic filtering logic which could—for example—filter out categories which remain rare within a journal. This way, no customized filtering would be needed and our approach will remain 100% generic, potentially suitable to any domain. See Section 3.4.2.2 for more details.

3.2 Evaluation

In order to validate the efficiency of our approach, both offline and online evaluations were realized. Both datasets were specifically created for this purpose, and the offline one was embracing less articles than the online one.

3.2.1 Offline Evaluation

The offline evaluation includes a small number of papers and is mainly targeted to articles from physical sciences.

3.2.1.1 Dataset

The dataset is composed of 595 articles, from seven journals and two Open Access (OA) publishers. Six journals are in the field of physical sciences, whereas the last one is in a totally different field, namely Pediatrics. This isolated journal confirms that the approach is not dedicated to one single field and is enough general to be applied to other scientific domains. Figure 3.15 shows a distribution of the different articles in their respective journals.

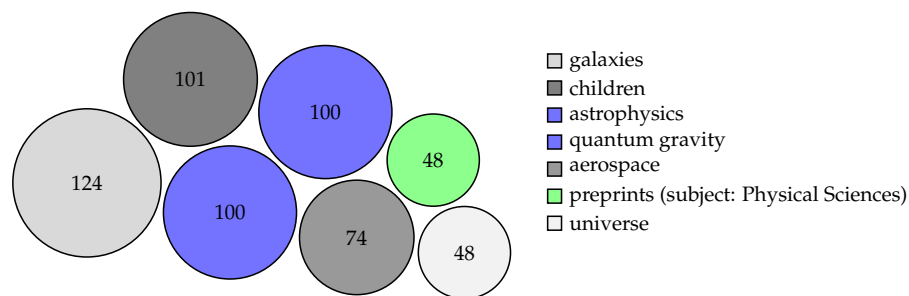


FIGURE 3.15: Offline evaluation – articles journal distribution

Articles represented by blue bubbles are from the IOP Publishing¹², the gray ones are for MDPI¹³ articles and the green bubble represents preprints from Preprints.org¹⁴ in the subject "Physical Sciences". This distribution also points that articles from *Children* are well represented (i.e., 1/6 of the dataset). All articles for the given journals were extracted in December 2016 and included into a file, except for the IOP journals which were too big—respectively 12,000 and 110,000 articles—and for which a limit of the 100 latest articles was defined.

DOIs, titles and keywords are given in Table 3.1 and only keywords are used by our approach for categorizing articles.

3.2.1.2 Categorization

Best connected categories returned by our approach were manually checked and annotated as correct or not in regards of the article's context. If the category was in relation with one of the article's topic, it was marked as valid. If the categories were somehow in relation but far—*or aside*—from the main context it was marked as incorrect. The analysis was as objective as possible, and standard metrics were computed from it. Precision P and Recall R are described as follow:

¹²<http://iopublishing.org/>

¹³<https://www.mdpi.com/>

¹⁴<https://www.preprints.org/> – MDPI preprints repository

TABLE 3.1: Sample of the dataset.

Title	Two-Body Orbit Expansion Due to Time-Dependent Relative Acceleration Rate of the Cosmological Scale Factor
Keywords	classical general relativity; cosmology
doi	10.3390/galaxies2010013
Title	Large Scale Cosmological Anomalies and Inhomogeneous Dark Energy
Keywords	dark energy; cosmological principle; inhomogeneous anisotropic universe
doi	10.3390/galaxies2010022
Title	Metamaterial Model of Tachyonic Dark Energy
Keywords	dark energy; analogue spacetime; hyperbolic metamaterial
doi	10.3390/galaxies2010072

$$P = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

where :

- *True positive* are articles with propositions containing only correct categories
- *False positive* are articles with propositions containing at least one wrong category

(3.7)

$$R = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

where :

- *False negative* are articles with propositions containing only correct categories but marked as incorrect.

(3.8)

In Equation 3.8, *false negatives* might be articles with more than ten correct categories or articles with no connection from categories (but from which only correct synsets were returned with corresponding categories). However in order to not annotate categories that will never be used—*reminder: unconnected categories are never proposed because there is no confidence degree*—, only connected categories (when less than ten) are returned. Hence, recall is always good given that the less restrictive threshold (see Section 3.3.1) will embrace all proposed ones.

F1 (or *F1 – score*) is another unique metric representing the harmonic mean between Precision and Recall:

$$F1 = 2 * \frac{P * R}{P + R} \quad (3.9)$$

Coverage *C* represents the ratio of articles exclusively correctly categorized over the entire dataset:

$$C = \frac{\text{true positive}}{\text{Total number of articles}} \quad (3.10)$$

Given that the desired approach must be precise, with an acceptable recall and

covers as many articles as possible, another metric was computed. It is the harmonic mean (H) of the three metrics and represents in a single metric the mode/threshold resulting the best compromise:

$$H = 3 * \frac{P * R * C}{P * R + P * C + R * C} \quad (3.11)$$

All of the above metrics are displayed in Table 3.2. We can observe that exact search provides a good precision and recall (0.95 and 0.93 respectively). Splitting keywords decreases the precision by two percents, but significantly improved the coverage. Indeed, there are almost twice more articles correctly categorized in further search (i.e., with split) compared to exact search (42% vs. 22%). Overall, $F1$ score is higher (i.e., better) for the further search, which also covers more articles. Consequently, a better harmonic mean (H) is observed.

TABLE 3.2: Categorization—metrics.

	Precision	Recall	$F1$	Coverage	H
Exact	0.95	1.0	0.97	0.22	0.46
Further	0.91	1.0	0.95	0.41	0.66

To summarize the results of the proposed categorization approach on the offline dataset, the further search—which splits unconnected keywords into sub-keywords—brings much more correct entries than the exact search (Coverage), even though the precision slightly decreases. The results for more variants and different options (as well as their descriptions) are given in Section 3.3.

3.2.2 Online Evaluation

After analyzing recommender systems for the scientific literature, Beel [11] concluded that user studies and online evaluations were the most adequate methods to evaluate a system. Therefore, a subset of Scilit data was created specifically for that purpose.

3.2.2.1 Dataset and extraction overview

A subset of Scilit was created and a new branch of the project was deployed onto a subdomain¹⁵ to host the user studies. Some 80,353 articles were fully randomly selected among Scilit OA articles, from a total of 5153 journals published by 886 publishers. Among these publishers, 504 (i.e., 57%) have less than 10 articles and 67 have more than 100 articles (i.e., 8%). Figure 3.16 illustrates the articles distribution for publishers with more than two thousands articles within the dataset. This distribution shows that the dataset is a representative subset of Scilit data—even though not fully proportionate—because the top publishers are all included into the top15 OA publishers¹⁶.

The categorization approach disambiguated articles' keywords by identifying connections among their potential synsets' categories. Domains, categories and corresponding synsets were saved into the database. We analyzed domains overlapping in order to see whether meaningful domains intersections were created or not—*categories overlap too much to represent them in a meaningful and comprehensive way.*

¹⁵<http://research.scilit.net/>

¹⁶<https://www.scilit.net/rankings>

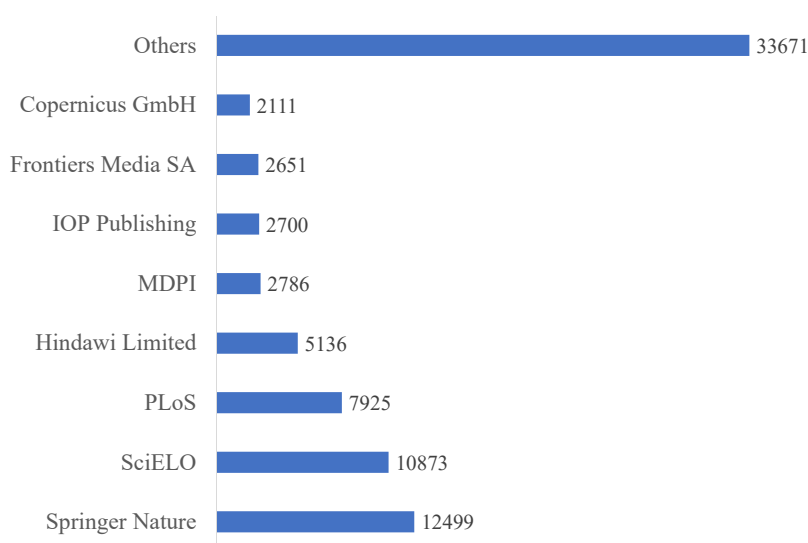


FIGURE 3.16: Publisher distribution.

Therefore, the pairwise domain analysis was conducted and the top 10 pairs are shown in Table 3.3. Only comprehensive combinations emerged on top of this table, which means that domains overlapping the more in our dataset are only plausible pairwise. To illustrate these meaningful combinations, we could showcase a paper¹⁷ which has the domain *Chemistry and mineralogy* co-occurring with *Engineering and technology* or another one¹⁸ belonging to the domains *Health and medicine* and *Biology*. Indeed, the legitimacy of some pairs is obvious.

However, some unexpected but rare domain combinations were also found for isolated cases such as a paper¹⁹ from which the three domains *Biology, Religion, mysticism and mythology* and *Philosophy and psychology* were activated. Thankfully, those remain marginal while only 35% of domain pairs occur in less than 5 different articles. This is the remaining noise inherited from unrelated synsets that our approach did not filter out. Some ideas to reduce their number are enumerated in Section 3.4.

Finally, the categorization process found connections for 27,880 articles (i.e., 34.7% of the articles in entry) from 41,072 distinct keywords. All of the 34 BabelNet domains were represented from the 5728 extracted categories inherited from 19,671 unique synsets. In other words, there is an average of 2.2 (± 1.5) categories and 1.25 (± 0.5) domains per article when some connections are identified. Moreover, categories coming from 3.6 (± 1.96) synsets per article in average, result from 3.2 (± 1.38) keywords. Having more synsets than keywords per article means that several synsets were returned per keyword. This is especially meaningful in the case of multi-words keywords, from which different sub-keywords were connected to synsets from other keywords/sub-keywords (Section 3.1.6.6).

3.2.2.2 Methodology

To perform online user studies, an evaluation form was added to the article's pages where authenticated users were able to rate selected categories. In order to make the evaluation as objective as possible, seven journals from different areas were selected

¹⁷<http://research.scilit.net/article/8717813319c652ea9afe62d9c5342e71>

¹⁸<http://research.scilit.net/article/81b8cc6e0611cb9ccda7da002ad03354>

¹⁹<http://research.scilit.net/article/fc5635a1ada1f1e3fc0143c08a8aeb0d>

TABLE 3.3: Domain overlapping.

Domain-1	Domain-2	Occurrences
Health and medicine	Biology	1782
Chemistry and mineralogy	Biology	674
Chemistry and mineralogy	Health and medicine	420
Chemistry and mineralogy	Physics and astronomy	417
Health and medicine	Philosophy and psychology	401
Physics and astronomy	Mathematics	355
Biology	Animals	229
Chemistry and mineralogy	Engineering and technology	143
Physics and astronomy	Engineering and technology	121
Mathematics	Computing	105

and 52 experts on corresponding domains were contacted. A link of ten-different-articles matching their field of expertise was sent to each editor, which was asked to rate at least three of them. After accessing the article's page containing metadata-i.e., title, abstract, keywords, journal information, etc-, their role was to evaluate from their knowledge, whether categories and domains resulting from our categorization approach were acceptable (i.e., correct) or not.

Figure 3.17 is the interface seen by the evaluators. On top of the abstract page, they see title, abstract and other metadata about the article (e.g., journal, authors, etc.). The evaluation section appears right below the keywords and starts displaying keywords for which synsets have been found. Below appear the two boxes for domains and categories rating from which evaluators have a yes/no radio button.

In total, 431 categories were rated by 24 evaluators, with an average of 18.5 categories (± 11.4) rated from 6.375 articles (± 4) per evaluator.

3.2.2.3 Results

Among these 431 rated categories, 71% (i.e., 307) of them were positively rated whereas 124 were estimated as wrong. A further analysis of the bad ratings was undertaken in order to better understand when the categorization fails. Figure 3.18 illustrates the distribution of categories ratings. Among the 124 negatively rated categories, 43 ratings (35%) were the consequences of bad keywords²⁰ in entry of our categorization process. The evaluation remains subjective and categories were estimated 52 times (i.e., 42% of the bad ratings) as invalid whereas connected synsets were correctly contextualized. Hence, these connections even though legitimate, do not seem to bring the most representative categories. Finally, 29 categories are rated as incorrect and are errors directly inherited from our approach. Those are either coming from the split logic of the further search (Section 3.1.5) or from BabelNet broadness/richness. Concrete examples of these identified problematic cases are given in Section 3.4.

Non-representative categories are from meaningful synsets' connections but are only *partially* correct. Their ratings are controversial because some experts might positively rate them (because they are representative of some keywords) whereas

²⁰Bad keywords are non representative keywords where even human would have difficulties to estimate the correct categories (see Section 3.4.1 for examples). In addition, three articles with more than ten keywords were also labeled as bad keywords because keywords are not representative of the article or too broad - involves 11 ratings representing 25% of bad keywords ratings.

Keywords: Dinoflagellates / Voltage-gated ion channels / ciguatera fish poisoning / neurotoxins / molecular action mechanism / paralytic shellfish poisoning / neurotoxic shellfish poisoning / azaspiracid poisoning / yessotoxin / palytoxin

Evaluation (article id: 67815)

This article:

Keywords:

8830 -> (Paralytic_shellfish_poisoning (paralytic shellfish poisoning))

17979 -> (Palytoxin (palytoxin))

16250 -> (neurotoxin (neurotoxins))

8828 -> (Ciguatera (ciguatera fish poisoning))

17978 -> (Azaspiracid (azaspiracid poisoning))

8829 -> (Neurotoxic_shellfish_poisoning (neurotoxic shellfish poisoning))

Show/hide synonyms and other related words

Is this domain acceptable?

Health and medicine

yes no

Are these categories acceptable?

Toxic_effect_of_noxious_substances_eaten_as_food

yes no

Seafood

yes no

Neurotoxins

yes no

Ion_channel_toxins

yes no

FIGURE 3.17: Evaluation—categories and domains.

others would consider them as wrong. Ratings from bad keywords are also meaningless for the categorization evaluation because it is often impossible (at least in the proposed approach) to properly categorize an article with off-topic keywords. Without those ratings—which are not directly in relation with the efficiency of our approach—91% of the 336 categories ratings are correct (i.e., 307 positive ratings) and only 29 categories (i.e., 9%) are false positive. Ideas about how to reduce the errors are given in Section 3.4.

More generally, 110 articles were evaluated by the 24 evaluators (Figure 3.19). Among those, 58 articles received exclusively positive ratings (green), involving 130 ratings with an average of 2.4 ratings per article (± 1.5). In contrary, 16 articles received exclusively negative ratings (red in the Figure 3.19) with a total of 27 ratings (1.68 ratings/article ± 2.17). The remaining third of the rated articles received both positive and negative ratings (orange), embracing a total of 290 ratings with 8.05 ratings per article in average and a standard deviation of 6.94. Over these articles that received hybrid ratings, 60% of the ratings were positive ones (green part on the right side).

Finally, the categorization of articles' keywords using BabelNet synsets obtains

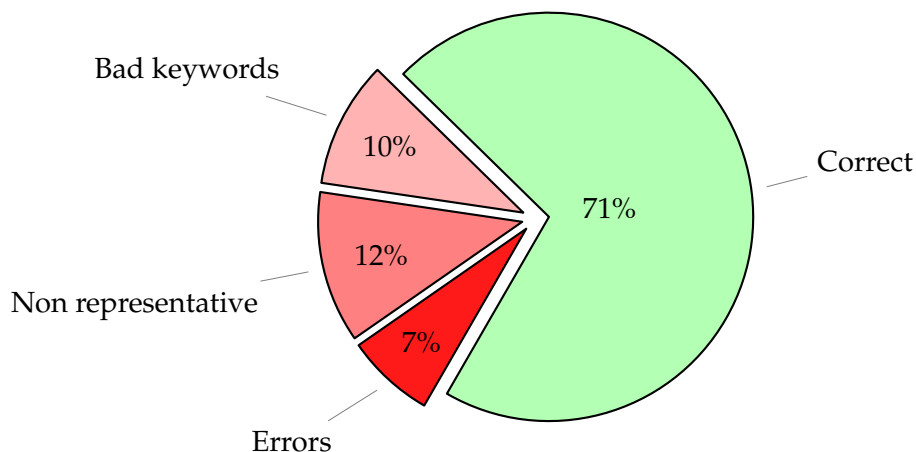


FIGURE 3.18: Categories ratings overview.

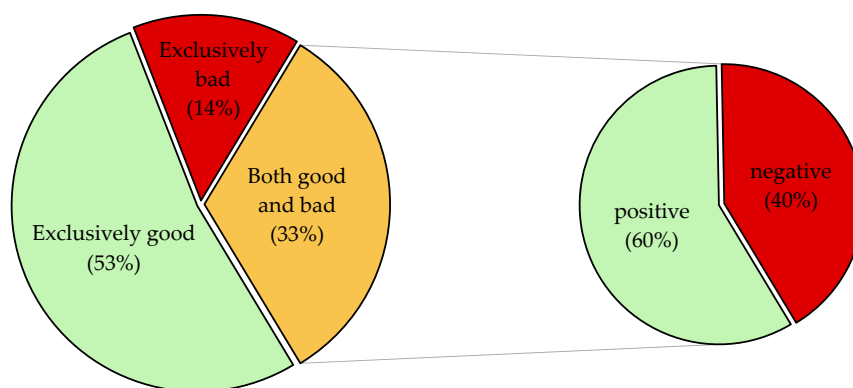


FIGURE 3.19: Distribution of ratings per articles.

promising results, but might be problematic when the category of a synset is too broad / narrowed. As an example, a synset with the broad category *Molecular biology*²¹ inherited from Wikipedia may be really specific or in contrary really general. Thus, connecting another synset with this category might bring dissatisfaction to experts for which the category would not necessarily be as specific as they would expect. In contrary, a synset with really narrowed category such as *Trifolieae*²² has only a very little probability to get connections because only a very few synsets belong to this category. Moreover, even though connections are found there is a high probability that this category is too specific.

²¹https://en.wikipedia.org/wiki/Category:Molecular_biology

²²<https://en.wikipedia.org/wiki/Category:Trifolieae>

3.3 Experiments

The development of the categorization part was an incremental process where the approach evolves within time and several different options were implemented and tested. This section gives more details about those. Some options were not finally activated because they did not improve the global precision / recall within our evaluation phases. However, they were kept into the code and can easily be activated to test them on other datasets for which they might perform better.

The first approach developed was the exact search. Keywords were searched against BabelNet without any pre-processing and returned synsets were analyzed. Several experiments were tested in order to identify the discriminant data for the disambiguation of synsets. Indeed, a huge number of parameters can be taken into account, such as the number of categories in common, the number of keywords sharing these categories, the ration between both and so on. Being more restrictive—*e.g., add a high threshold to select categories*—makes the system more precise—*because only highly connected categories are selected*—but with a really low recall. Results obtained at this step were relatively good and the intuition that BabelNet could be used for the disambiguation of keywords was confirmed.

3.3.1 Threshold Parameter

By analyzing synsets returned by the exact search, observations are converging to the conclusion that categories and domains are good unifying elements to disambiguate the sense of ambiguous keywords. But what are the conditions required to connect two synsets? What is the threshold when categories / domains are activated? To answer these questions, the threshold parameter α , which defines the applied selection criteria, is set up. Its value varies from 1 (most restrictive value) to 4 (more flexible value), as shown in Table 3.4.

TABLE 3.4: Threshold parameter (α) defines the restriction of the selection criteria.

Value	Constraint
1	minimum of three keywords share the item
2	two keywords share only one item
3	two keywords share one to three categories and domain validated (with $\alpha = 1$)
4	minimum of two keywords share one, two or three items
5	an item is shared by a minimum of two keywords

- item can be categories or domains
- lower α constraints are integrated into the higher values
i.e., 1 is effective in 1/2/3/4, 2 in 2/3/4, and so on

The analysis of results brought by each value of this parameter was realized on the (offline) dataset described in Section 3.2.1.1. Each category and domain selected by the connection—*based on the criteria imposed by the α parameter*—are manually annotated (i.e., correct/wrong) in a separated file, for reproducibility purpose. Indeed, a pair article-category (or article-domain) would only need to be annotated once. The best value for the α parameter is 5 for the categories, because it brings a precision of 0.91, a recall of 1.0 and covers 41% of the articles as shown in their respective tables ($H = 0.66$). The tuning of the threshold parameter depends on the domain

of application. Actually, if only highly precise connections are desired, α might be set to a lower value. However, recall would significantly decrease and the approach will cover much less entries. More results are given in Section 3.3.4, where metrics obtained by every value of α are compared.

3.3.2 Description and Effects of Used Options

3.3.2.1 Split – Classical/Modified n-grams

An n -gram is a sequence of n words. Classical n -grams are commonly used in text mining and other NLP (Natural Language Processing) approaches [24, 32, 85]. N -grams extraction is usually part of the pre-processing steps, and aims to discover frequent words combinations in order to use them rather than single words (see Section 2.1.3). Indeed, splitting a bi-gram or a tri-gram which has a specific sense might bring ambiguity because single words might have much more potential meanings. For that purpose, by default, synsets coming from single sub-keywords searches are only returned if they are connected to other sub-keywords from the same keyword in our approach.

Classical N-grams (*ngram*) – We started with classical n -gram extraction to search corresponding synsets against BabelNet. The same decreasing logic as the one described in Section 3.1.5 was used. Concretely, stopwords are removed and the largest possible n -gram is searched against BabelNet. If no synset returned, we reduce the window size (i.e., decrease n -gram size by 1) and request all possible n -grams. We repeat this until finding some synsets, or reaching uni-grams.

Modified N-grams (*ngram+*) – Modified n -gram extraction is the approach described in Section 3.1.5. This strictly follows the same logic of the classical n -grams extraction described here-above, but add extra n -grams inherited from linear onwards combinations. In other words, words order is not distorted but we may jump over some words to build new potential sub-keywords (i.e. skip-grams). This variation might slightly decrease the precision and improve the coverage, but it improves all metrics in our experiments. See Section 3.3.4 for a complete results overview for all options and variations.

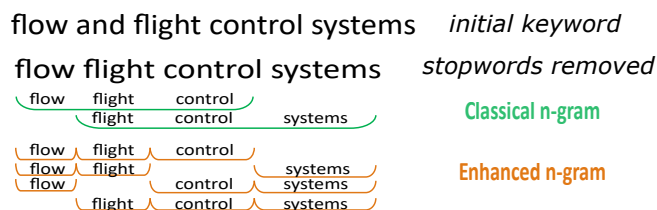


FIGURE 3.20: Split–Classical vs. Enhanced n-grams.

Figure 3.20 illustrates the difference between the classical version and our enhanced one. The basic n -gram simply applies a sliding window over the keyword to extract n -grams whereas the enhanced n -gram also allow gaps among words (without distorting words sequence). Consequently, the enhanced n -gram retrieves more combinations.

3.3.2.2 Two words exception (2W)

Given that splitting keywords can bring extra noise, the approach considers by default that only two cases are specific enough to keep synsets from sub-keywords as potentially representative of the original keywords. The first case is when the synsets are coming from multi-words, and the second one is when synsets inherited from single-words sub-keywords are inter-connected within their parent keyword. Indeed, we assumed that when synsets share categories they are safe to be used at the keyword level (i.e., the sub-keywords synsets connections are propagated to their parent keyword) in the connection process (Section 3.1.6).

However, the goal of the further search being to generate more potential meaningful sub-keywords to obtain and connect corresponding synsets, this 2W option brings more flexibility for keywords composed by two words (after stopwords removal). Synsets coming from single words from the same parent keyword are also kept as potential matching synsets for the keywords connection **even if no common category among those is identified**. Thus, they are considered almost as safe as their original keyword.

Effect – The two-words exception (2W) is an option which enhances the tolerance of our approach. When activated, unconnected synsets found from single-words sub-keywords are kept as potential synset of their keyword—i.e., *used for keywords connections*—for keywords composed of only two words after stopwords removal. This potentially brings more synsets (noisy or correct ones) to the keywords connection step. Hence more errors are made and precision decreases but the coverage increases (compared to other more restrictive options such as *NO*, *1L*, *NF* and *SF* – see Section 3.3.4.2) because correct categories are extracted for more articles.

3.3.2.3 Numbers filtering (NF)

Because numbers are mostly meaningless they are removed from keywords. Indeed, there might have some unexpected categories such as the number 10 which has different synsets with the categories *Integers*, *Tarot*, 10, *Turkish-language singers*, *Egyptian hieroglyphs*, *New York City Subway services* and so on. Numbers might however be meaningful when embedded into words, such as in the molecule H_2O . Therefore only fully numerical words are removed.

Effect – The numbers filter (*NF*) removes digital words from keywords. This does not impact too much the results because digital (sub-)keywords were not more present than single letters within the test dataset. This option, however, improves a bit the precision compared to *NO* mode (see Section 3.3.4.2) while three wrong categories (two *Integers* and *2010_debut_albums*) are removed whereas all correct ones are conserved. In addition, given that numbers will never help for the disambiguation of keywords, this option was not activated.

3.3.2.4 Soft filter (SF)

As discussed in Section 3.1.6, categorization might be realized by finding connections from synsets' categories, but also from domains. Table 3.5 gives the precision, recall, F1, harmonic mean and coverage of the synsets connections by their domains.

The analysis of returned domains provides metrics similar to the analysis of categories connections (P : 0.90 and R : 1.0 vs. 0.91 and 1.0). However, the coverage is much higher (0.81 vs. 0.41). This is expectable given the fact that domains are much more general than categories (see Section 3.1.1). Hence, domains have more chance to overlap than specific categories. SF is used in the split function to reduce the list of potential synsets by filtering unconnected synsets as soon as other synsets have any item in common. This creates a smaller and safer list of non-connected synsets²³.

Effect – The soft filter (SF) is an option to filter unconnected synsets with connected information from the further search. For example, if two synsets are connected by their domains from two different single-words sub-keywords but the searches brought many synsets without category connection, the domain connection will be used to filter out unconnected synsets. It does not seem to have an important contribution in regards of our dataset, even though the idea seems good.

3.3.3 Description and Effects of Unused Options

TABLE 3.5: Categorization—metrics (domains).

	Precision	Recall	F1	Coverage	H
Exact	0.95	1.0	0.97	0.50	0.74
Further	0.90	1.0	0.95	0.81	0.9

Even though results seem to be better with domains, finding / suggesting related articles is the next step after the categorization. For that purpose, domains are much too large to propose accurately articles to a user reading a specific article. Therefore, categories connection is the selected approach.

3.3.3.1 One letter filtering ($1L$)

For the same reason as the numbers filtering (Section 3.3.2.3), a single letter does not necessary bring lots of information. In contrary, they mostly bring more noisy synsets such as the ones with categories *Units of temperature*, *Potassium*, *1000 (number)*, and *Chemical elements* for the letter K . However, several of these categories might be legitimate in scientific literature, such as molecule names in *Biology* which would be retrieved from single-letters. Hence this filter was not activated because we might remove meaningful categories and synsets from single letters²⁴.

Effect – The $1L$ filter ignores 1-letter-words. This option does not improve results within this dataset because there was not many single letters keywords represented, and also because synsets were not used for keywords connection while there was no sub-keywords connections. Moreover, single-letters words could provide meaningful synsets in specific

²³The filtering of unconnected synsets removes unrelated categories from synsets belonging to unrelated domains. Therefore, even though synsets do not share any category, all of their categories are correct in regards to the article sense. Consequently, it becomes a false negative (i.e., relevant entry considered as wrong). However, given that no confidence can be allocated to those, we decided to not include them into the results.

²⁴In online dataset, connections from 26 1-letter sub-keywords (i.e., the whole alphabet) were found for 162 articles. This embraces 185 unique keywords and 120 distinct synsets.

fields (e.g., molecular sciences), hence this filter is not used but extra noise—*i.e.*, *non-scientific categories*—was identified (e.g., `_Subway_services`).

3.3.3.2 Single words (SW)

This option is similar as the two-words exception (Section 3.3.2.2), except that there is no restriction to return synsets inherited from single sub-keywords searches. This option brings a lot of extra potential synsets—*unrelated (i.e., noisy) or related*—to be compared by our keywords connection logic. This option was considered as too dangerous at the time when development and investigation was done, and probably too costly to run in such a big amount of data. However, its evaluation provides good results and it even performs better than two-words exception.

Effect – Using the single-words exception (SW) reflects to the same behavior as the 2W usage in general. It obtains one of the best coverage by comparing the options with single option activated (see Section 3.3.4.2).

3.3.4 Variants Results

As presented here-above, several different options were implemented and tested. This sections aims to further compare their contributions and efficiency.

3.3.4.1 Basic modes

Firstly, we start the comparison with our three main modes, to compare the effects of each of them onto our evaluation metrics. In this comparison, the n-gram modes are using only the two safest and more basic options, namely static noise and numbers filtering. This section compares the efficiency of the three following modes:

- Exact search (ES) – Section 3.1.4
- Classical n-gram – Section 3.3.2.1
- Enhanced ngram (ngram+) – Section 3.1.5

The exact search is considered as the baseline because it is the naive approach. The second mode is the further search with the classical n-gram approach. Then, our enhanced n-gram approach is used in comparison with others. Precision, Recall, Coverage and Harmonic mean are compared respectively in Table 3.6, Table 3.7, Table 3.8 and Table 3.9. The most precise approach is the exact search, which is logical because this is also the safest way to search for words. Given that keywords are not pre-processed, the searches returning synsets are consequently the most accurate because no variation with the original keywords is realized.

TABLE 3.6: Categories - Precision – 3 modes.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Exact	1.00	0.96	0.95	0.95	0.95
Ngram	1.00	0.96	0.94	0.94	0.93
Ngram+	1.00	0.96	0.94	0.94	0.94

TABLE 3.7: Categories - Recall – 3 modes.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Exact	0.18	0.79	0.92	0.96	1.00
Ngram	0.22	0.77	0.94	0.96	1.00
Ngram+	0.21	0.75	0.93	0.96	1.00

TABLE 3.8: Categories - Coverage – 3 modes.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Exact	0.04	0.18	0.21	0.22	0.23
Ngram	0.08	0.27	0.33	0.34	0.35
Ngram+	0.07	0.27	0.33	0.34	0.36

The classical n-gram approach activates the further search (i.e., split option) and keywords are split on spaces and punctuation. This mode slightly decreases the precision—because *splitting keywords add both more potential correct synsets to connect and more noise*—but significantly increase the coverage—because *it helps to find new correct connections*—, which is an expected behavior. Finally, the enhanced n-gram mode (n-gram+) add extra linear combinations compared to the classical n-gram one. Surprisingly, it slightly improves both precision and coverage while we were expecting the precision to decrease²⁵. Recall is for all modes equal to 1, because unconnected categories are not kept and hence not counted as correct/wrong. Table 3.9 shows that n-gram+ outperforms the other modes, while considering these 3 metrics.

3.3.4.2 Further options effect analysis

This section further analyzes and compares the effects of options that were presented in Section 3.3. General options' behaviors and effects have already been described in their respective sections, but specificities and combinations of modes are given in this section. We applied those to the n-gram+ approach because this was the mode which provided the best compromise for precision, recall and coverage, as shown in the Harmonic mean (Table 3.9).

²⁵N-gram+ actually provides, in this usage, better precision than classical n-gram because only multi-words keywords or single-words sub-keywords with connections are returned.

TABLE 3.9: Categories - Harmonic mean – 3 modes.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Exact	0.10	0.38	0.43	0.44	0.46
Ngram	0.16	0.50	0.58	0.59	0.61
Ngram+	0.16	0.49	0.58	0.59	0.61

The following options are discussed and compared with the metric tables:

- N-gram+ – *the baseline approach, with SN and NF options*
- No option (NO) – *N-gram+, without any option, not even static noise removal*
- Numbers filtering (NF) – *Remove isolated numbers (Section 3.3.2.3)*
- Static noise (SN) – *Removal of identified noise (Section 3.1.7)*
- One letter (1L) – *Removal of single letter words (Section 3.3.3.1)*
- Two words exception (2W) – *Unconnected sub-keywords searches are only kept for 2-words keywords (Section 3.3.2.2)*
- Single word (SW) – *The most basic ngram approach: split returns synsets even for unconnected sub-keywords (Section 3.3.2)*
- Soft filtering (SF) – *Known connections (from domains or categories) are used to filter unconnected entries (Section 3.3.2.4)*
- All – *All the above options activated*
- Combination of options
 - Ngram+ with SN, NF, SF (SN_NF_SF)
 - Ngram+ with SN, NF, SW (SN_NF_SW)
 - Ngram+ with SN, NF, SW, SF (SN_NF_SW_SF)
 - Ngram+ with SN, NF, 2W, SF (SN_NF_2W_SF)

TABLE 3.10: Categories - Precision – all options effects.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Ngram+	1.00	0.96	0.94	0.94	0.94
ng_NO	0.81	0.81	0.76	0.76	0.71
ng_1L	0.81	0.81	0.76	0.76	0.71
ng_NF	0.85	0.82	0.77	0.77	0.72
ng_2W	0.73	0.76	0.71	0.71	0.67
ng_SF	0.81	0.81	0.76	0.76	0.72
ng_SN	0.96	0.95	0.93	0.93	0.93
ng_SW	0.72	0.75	0.71	0.70	0.66
ng_all	0.97	0.95	0.90	0.89	0.90
SN_NF_SW	0.99	0.95	0.91	0.91	0.91
SN_NF_SF	1.00	0.96	0.94	0.94	0.94
SN_NF_SW_SF	0.99	0.95	0.91	0.90	0.90
SN_NF_2W_SF	0.99	0.95	0.91	0.90	0.91

All of these variations use the same disambiguation logic (i.e., keywords connection) which is the Hadamard product of the three categories matrices (per keywords, sub-keywords and synsets) presented in Section 3.1.6. In addition, from an article level, only categories from connected synsets are annotated—i.e., *categories from isolated synsets (i.e., synsets without category connection) are not used for the evaluation.* Moreover, in the further search, synsets coming from single-words sub-keywords searches are only conserved when there is a connection with other sub-keywords (or

TABLE 3.11: Categories - Recall – all options effects.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Ngram+	0.21	0.75	0.93	0.96	1.00
ng_NO	0.27	0.78	0.93	0.97	1.00
ng_1L	0.27	0.79	0.93	0.97	1.00
ng_NF	0.27	0.78	0.93	0.97	1.00
ng_2W	0.32	0.79	0.94	0.97	1.00
ng_SF	0.27	0.78	0.93	0.97	1.00
ng_SN	0.21	0.75	0.93	0.96	1.00
ng_SW	0.33	0.79	0.94	0.97	1.00
ng_all	0.27	0.75	0.95	0.97	1.00
SN_NF_SW	0.27	0.75	0.95	0.97	1.00
SN_NF_SF	0.21	0.75	0.93	0.96	1.00
SN_NF_SW_SF	0.27	0.75	0.95	0.97	1.00
SN_NF_2W_SF	0.27	0.75	0.95	0.97	1.00

TABLE 3.12: Categories - Coverage – all options effects.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Ngram+	0.07	0.27	0.33	0.34	0.36
ng_NO	0.09	0.25	0.30	0.31	0.32
ng_1L	0.08	0.25	0.29	0.31	0.32
ng_NF	0.09	0.25	0.30	0.31	0.32
ng_2W	0.11	0.28	0.33	0.34	0.35
ng_SF	0.08	0.24	0.29	0.30	0.31
ng_SN	0.07	0.27	0.33	0.34	0.36
ng_SW	0.11	0.28	0.33	0.34	0.35
ng_all	0.11	0.31	0.39	0.40	0.41
SN_NF_SW	0.11	0.31	0.39	0.40	0.41
SN_NF_SF	0.07	0.26	0.33	0.34	0.35
SN_NF_SW_SF	0.11	0.31	0.39	0.40	0.41
SN_NF_2W_SF	0.11	0.31	0.39	0.40	0.41

when SW/2W options are activated). Generally, the observed tendency is that the more restrictive we are (i.e., small alpha, more filters) the more precise results we get to the detriment of recall and coverage which decrease. Given that we matter about all of these three metrics—*precision* (Table 3.10), *recall* (Table 3.11) and *coverage* (Table 3.12)—, the higher α value (i.e., 5) brings the best results in every mode (see Harmonic mean table for more details – Table 3.13). Hence, results will be discussed mainly for this threshold.

The mode without noise filtering has no option (NO) activated—*not even the static noise removal*—and uses default behavior of the further search (i.e., only connected synsets are retained). It performs relatively well because it achieves a precision bigger than 0.71 and covers almost 32% of all articles in entry. It even outperforms some options for specific values of α . The explanation is that as long as only synsets from multi-words sub-keywords (or connected single-words sub-keywords) are kept, the adding of extra noise—*inherited from the splitting logic and single-words searches*—is minimized.

The single-words exception (SW) is the less restrictive mode and uses the enhanced n-gram approach without any option and special case. Indeed, no static

TABLE 3.13: Categories - Harmonic mean – all options effects.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Ngram+	0.16	0.49	0.58	0.59	0.61
ng_NO	0.18	0.46	0.52	0.54	0.54
ng_1L	0.18	0.46	0.52	0.54	0.54
ng_NF	0.18	0.46	0.52	0.54	0.54
ng_2W	0.22	0.48	0.54	0.56	0.56
ng_SF	0.18	0.45	0.51	0.53	0.54
ng_SN	0.16	0.49	0.58	0.59	0.61
ng_SW	0.23	0.48	0.54	0.56	0.56
ng_all	0.22	0.53	0.63	0.64	0.66
SN_NF_SW	0.22	0.53	0.64	0.65	0.66
SN_NF_SF	0.15	0.49	0.58	0.59	0.61
SN_NF_SW_SF	0.22	0.53	0.64	0.65	0.66
SN_NF_2W_SF	0.22	0.53	0.63	0.64	0.66

noise is removed and all unconnected synsets from sub-keywords are used to identify keywords connections (even if they come from single-words). This is also the less precise one, but not necessary the one which bring the worse results by looking at the harmonic mean of our three metrics.

The filtering of static noise (*SN*) is very useful in our dataset. Its contribution is not obvious while comparing single option usage, because the noise inherited from further search is limited—*because unconnected single-words are ignored*. However, when the *SW* option is activated, the noise filtering is really efficient to filter out non-scientific categories. This is especially useful for articles in physics and astronomy—*because lot of songs/albums/books cover topics like gravity or dark matter*—which are massively represented in our dataset.

The safest and also the most precise mode is the *Ngram+*, which combines *NF* and *SN* filtering. If the precision is the most important metric to be considered, this default mode should be used. The drawback of this mode is that it covers only 35% of the articles in entry, but this is not too far from the most precise options bringing correct categories for 41% of the articles.

NF is definitely a safe option for the keywords sense disambiguation. *SN* is obviously an efficient options that must be applied for the categorization of scientific articles—*if only the options SW and SF are activated, results obtained are not satisfactory (P: 0.67, R:1.0, C: 0.35, H: 0.56)*. The *1L* is controversial because it could remove legitimate synsets. Therefore, some options were added to the proposed *Ngram+* baseline which is the combination of *SN* and *NF* options.

Bringing more synsets with the *SW* option (*SN_NF_SW*) decreases the precision but increases coverage, as expected. Finally, the harmonic mean for this combination is 5% higher than the basic *N-gram+*. Adding the *SF* does not change significantly the results (*SN_NF_SF* and *SN_NF_SW_SF*), but there is no risk to use it. The last combination tried (*SN_NF_2W_SF*) is the *2W* together with *SN*, *NF* and *SF*, which brings similar results as its single-words variant (*SN_NF_SW_SF*).

Activating all of the options (*all*) available brings acceptable results on this dataset. It is not the most precise option (i.e., 0.90) but covers 41% of the articles. At the end, it achieves one of the best Harmonic mean (0.66). It might be an interesting default mode to test on new datasets, but we would still suggest to think about each option depending on the type of data.

3.4 Future work

This section gives an overview about the problematic cases that remain challenging to resolve from the categorization part, and the solutions / perspectives to improve its results.

3.4.1 Problematic Cases

Even though both offline and online evaluations bring satisfactory results—*respectively 0.93 and 0.91 precision*—a few problematic cases were identified. Unfortunately, some of them are hard—if not impossible—to avoid.

3.4.1.1 Valid but non-representative connections

The first source of the 124 negatively rated categories from the online evaluation is actually controversial. Indeed, 42% of them (i.e., 52) are from correct but non-representative categories. These partially (in-)correct categories are often the consequence of broad (i.e., general) / vague (i.e., not precise enough) / non-representative keywords. Synsets inherited from those may be representative to only a part of the keywords, and hence categories will not be the most representative of the article.

As an example, an article²⁶ describing a new algorithm realizing the segmentation and the classification of LiDAR²⁷ point cloud has the following keywords: *cross-line elements, plane segmentation, airborne LiDAR point cloud, line segmentation, and fast segmentation*. Three keywords over five include the term "segmentation" associated with one of its characteristics, another one—*cross-line elements*—does not bring a lot of information about the content of the article (i.e., is non-representative), and the last one is about the source of the data (i.e., the LiDAR). Even though humans might guess that the article is about some image segmentation problems, keywords are not really specific and explicit about the article's context. Finally, because none of the keywords/sub-keywords share any category with *segmentation*, only the non-representative category *Mathematical concepts* is activated from the connection of *Plane* and *Line* synsets (see Figure 3.12 for more details about the connection).

3.4.1.2 Poor keywords

The second source of negative category ratings is related to the previous one because it is the consequence of bad keywords quality. Actually, 35% of them (i.e., 43) are coming from non-representative or sometimes even off-topic keywords. Authors may choose keywords representing the field of applications they are working on, the name of developed project / algorithm, experiments conditions or other meaningless / off-topic keywords.

The following article's keywords *MODIS, fire events, ignition, extinction, Alaska, Portugal, Greece, California, Australia, uncertainty* are a perfect example of these low quality keywords. The first keyword (*MODIS – MODerate resolution Imaging Spectroradiometer*) is the material providing images to analyze and determine fires start / end conditions. The three next keywords are more specific to the topic, even though they describe the purpose of the method presented in the article²⁸ instead of the way

²⁶<http://research.scilit.net/article/4816fd81061b57628af0358ff3c4131c>

²⁷A LiDAR comes from the combination of "Light" and "raDAR". It is now used as the acronym of "Light Imaging, Detection, And Ranging"

²⁸<http://research.scilit.net/article/73ba2217be9a7f554f40c389454b0179>

to reach their goal. Finally, the five next keywords are country names from which the satellite images were from, which is not representative of the article and its proposed algorithm at all. Consequently, unrelated categories are found from the connections of synsets inherited from these bad keywords, such as *European Parliament constituencies* found in synsets from *Greece* and *Portugal*.

3.4.1.3 BabelNet limits

The last source of bad category ratings, in contrary to the first two reasons, is directly inherited from BabelNet architecture and probably from the way our approach uses it. This involves 29 bad ratings (i.e., 23%) from our experts evaluators. Keywords from the example given here-above may also highlight BabelNet's limits. Indeed, the synsets from the keywords *ignition*²⁹ and *extinction*³⁰ might have been sharing common categories since they are both about fire events. But these synsets do not have any categories attached, because none of these terms have a corresponding entry in Wikipedia—*Reminder: categories are mainly inherited from Wikipedia*. Hence, even though the correct meanings are attached to these synsets, they are useless for our disambiguation approach.

Consequently, our approach did find a common category from the keywords *Fire* and *Extension*, but from two novel books wearing these keywords as title, resulting into the category *American fantasy novels*. Moreover, other unrelated categories (*Schooners* and *Individual sailing vessels*) were found from ships named *California* and *Australia*. From this section, we conclude that both the richness and the information lack from BabelNet can be discriminant for the categorization.

3.4.2 Solutions and Perspectives

In this section, we bring ideas and perspectives to improve our categorization approach and resolve the remaining problematic cases enumerated in Section 3.4.1.

3.4.2.1 Extend to text

The first idea to avoid providing bad categories from poor keywords might be to extend our approach to abstracts and / or titles. Of course, this will involve an extra pre-processing step such as the part-of-speech (POS) tagging from a syntactical analyzer. For this purpose, good candidates might be SyntaxNet [7], CoreNLP [75] or Spacy [46]. Then, more precise requests—including POS—could be launched against BabelNet. Franco-Salvador et al. [35] implemented a classifier based on a word sense disambiguation relying on BabelNet data. Of course, taking sentences in entry would imply a modification of the disambiguation techniques given that many more potential connections would be added. Moreover the concept would not necessary be constant over different sentences of an abstract, but the idea is attractive.

3.4.2.2 Dynamic filtering per journal

Given that scientific articles are published in journals / conferences usually dedicated to a specific field, they should—in theory—mainly embrace articles somehow in relation to each other. Therefore, a pruning step might be put in place to cut (i.e., remove noise) or mark isolated categories per journal. Indeed, if a category is only

²⁹<https://babelnet.org/synset?word=bn:00045883n>

³⁰<https://babelnet.org/synset?word=bn:00032456n>

represented into one single article of a journal containing several thousands of papers, it might be an indicator about the quality of the category (i.e., wrong, too specific or innovative). Further analysis of this theory might be considered in a future work.

3.4.2.3 Lemmatization

Given that searching for plural and singular words in BabelNet does not necessarily bring the same results, an idea would be to lemmatize keywords. Either only multi-words keywords or all keywords, this lemmatization may result in more connections. However, it might also distort the meaning of words and bring more noise, so more wrong categories. Though, this idea still remains a good track to improve the recall and coverage of our approach.

3.4.2.4 Further use of domains

Section 3.3.2.4 showed that filtering by domains are providing much more results than categories' filtering. In future works, we will try to define what should be the threshold to use synsets and attached categories from domains connections. In other words, because domains are correct, categories might also be correct. The potential next problematics might be: Are all categories automatically correct whenever the synsets are sharing domains? From which number / proportion of keywords sharing common domains is it safe to keep even unconnected categories? If there is a rule to extract correct categories from domains connections, this will help to significantly covers many more articles.

3.4.2.5 Lesk algorithm

To overcome the limits of related words not sharing any category (see Section 3.4.1.3), the Lesk algorithm [69] might be implemented. This approach finds all senses of a given list of words—*usually within a sentence*—and identifies the best topic overlapping among their different potential senses (i.e., definitions). Then, the disambiguation is realized by selecting senses of these words overlapping the most. Indeed, it relies on the assumption that senses tend to overlap within a related word association. This approach might be applied to articles keywords in order to counter the leak of category overlapping.

3.5 Conclusion

This chapter described the usage of keywords to categorize scientific articles. Both the categorization of an article and the disambiguation of its keywords is achieved in the same time. Firstly, an exact search from original keywords is realized and potential matching synsets are returned by BabelNet. Then, after the stopwords removal, a modified decreasing n-gram logic is applied to keywords from which no synsets were obtained. Finally, a novel way of exploiting BabelNet is proposed where common categories are searched among synsets returned by all keywords. In other words, we use synsets categories intersections to disambiguate their corresponding keywords senses.

The proposed approach brings satisfying results, both from offline (Section 3.2.1) and online evaluations (Section 3.2.2). Indeed, the offline evaluation achieves a precision of 0.93 and a recall of 1.0 and correct categories are found in 42% of the 595

articles in entry. Respectively, connections are found in 35% of the 80,353 articles (i.e., 27,880). The ratings from domains experts end up with a precision of 91% over the 336 categories rated³¹.

The categorization opens doors to a further exploitation of BabelNet, especially with its ontology graph which gives the possibility, from a given synset, to get its related neighbors. This is the purpose of our second contribution, discussed in Chapter 4.

³¹When we exclude errors from poor and illegitimate (i.e., non-representative categories) keywords. More details are given in Section 3.2.2.3

Chapter 4

Information Retrieval

4.1 Our Approach

The second contribution is the exploitation of the minimized and contextualized data returned by the categorization step, which realizes the keyword sense disambiguation by identifying shared categories among keywords of the same article. Synsets bringing category connections are therefore validated (i.e., disambiguated) and their linked neighbors—*edges within the BabelNet ontology graph*—are further exploited. Indeed, articles similarity is measured from the keywords and neighbors intersections, thus the most related articles can be retrieved.

4.1.1 General Overview

The categorization step and the information retrieval are interconnected as illustrated in Figure 4.1. The first step of the retrieval is the data augmentation. For that purpose, neighbors words—*i.e., edges retrieved from a given synset¹ within BabelNet’s ontology graph*—are extracted. There are 39 different types of neighbors’ associative relations, mainly inherited from WordNet, Wikidata and Wikipedia Bitaxonomy, such as hyponym, hypernym, meronym, semantically inherited, similar to, etc². Then, neighbors synsets (called neighbors in the rest of the thesis) are exploited to compute similarity among articles from which synsets have been returned.

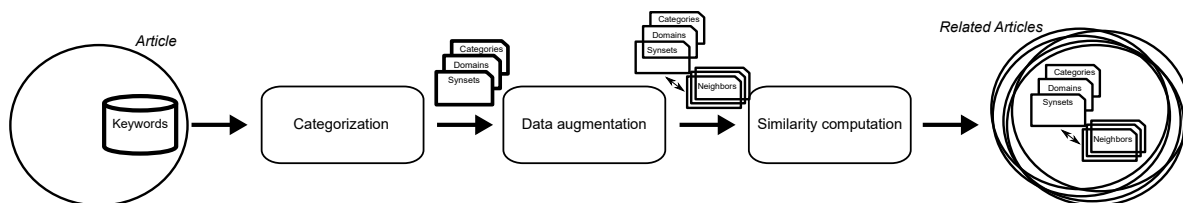


FIGURE 4.1: General workflow of our complete approach.

Finally, after disambiguating keywords, expanding data and computing similarity, the general approach tends to propose semantically related articles.

4.1.2 Data Augmentation

The previous categorization step reduced the metadata to a higher semantic level (i.e, categories). In this step, contextualized synsets are exploited in order to augment data. This is an important point because synsets may be really specific—*sometimes*

¹In BabelNet graph, vertices (i.e., the synsets) are connected by semantic relationships.

²All pointer types available here:

<https://babelnet.org/4.0/javadoc/it/uniroma1/lcl/babelnet/data/BabelPointer.html>

even rare—within the corpus of data. This rarity may lead to isolated articles from which no other paper can be suggested. Moreover, finding similar articles only with these disambiguated synsets will not bring more results than a basic full-text search approach—*even though it might remove false positives*.

4.1.2.1 Generalities

Our original idea was to add semantics into scientific search engines in order to link articles sharing the same idea even if they use trans-disciplinary terminology variation. As an example, we would imagine an article with the keywords *Categorization*, *Word Sense Disambiguation* and *Text Mining* being close to another one with keywords *Natural Language Processing*, *Semantic Similarity* and *Information Retrieval* because they are about the same topic. Hence, synsets' neighborhood will be checked, and related neighbors will be used. These neighbors might offer the possibility to recommend articles sharing related keywords.

4.1.2.2 Neighbors selection – Use case of artificial intelligence

Given that generic / common synsets have a lot of neighbors, only neighbors sharing at least one category in common with the article of origin are selected. An article with keywords "*Artificial intelligence, Evolutionary algorithms, Genetic algorithms, Optimization*" would be a representative example of this problematic amount. From this example, each keyword returned one synset matching at least with one category among *Cybernetics, Mathematical optimization, Optimization algorithms and methods*. The synset coming from *Artificial intelligence*³ has 2212 neighbors containing some unrelated synsets, such as *1997*⁴, *Chess*⁵, *Military*⁶, *Edward Fredkin*⁷ or some which might be slightly related in a specific context, such as *Go*⁸, *Video game*⁹, *The Terminator*¹⁰ or *Robot*¹¹.

Moreover, 1002 are duplicated neighbors which are inherited from different relation types (i.e., several edges pointing to the same vertex). Over the 1210 distinct neighbors, 5483 categories are found and the following list contains the most represented ones, in terms of number of occurrence per synset:

- Living people (71)
- Artificial intelligence (48)
- History of artificial intelligence (31)
- Artificial intelligence researchers (24)
- Cognitive science (22)
- Philosophy of artificial intelligence (19)
- 20th-century philosophers (19)
- American inventions (18)
- English-language films (18)
- Cybernetics (17)

³<https://babelnet.org/synset?word=bn:00002150n>

⁴<https://babelnet.org/synset?word=bn:02818166n>

⁵<https://babelnet.org/synset?word=bn:00018197n>

⁶<https://babelnet.org/synset?word=bn:00005732n>

⁷<https://babelnet.org/synset?word=bn:03135067n>

⁸<https://babelnet.org/synset?word=bn:00040833n>

⁹<https://babelnet.org/synset?word=bn:00021477n>

¹⁰<https://babelnet.org/synset?word=bn:02387171n>

¹¹<https://babelnet.org/synset?word=bn:00007371n>

From this list, some categories are obviously not matching with the disambiguated synset and the article’s context, such as *20th-century philosophers*, *Living people*, *Artificial intelligence researchers* and so on. Consequently, BabelNet can be seen as a dense graph connecting synsets to their neighbors. These connected synsets can be inherited from various fields, relation types (i.e., hyponyms, similar to, etc) and can be more or less related. Figure 4.2 illustrates this concept. The synset *Artificial intelligence* (in the middle) is connected to all of his neighbors (in gray)—*only a very small portion of the neighborhood is represented here*. The most expected neighbors for this synset are the synsets belonging to one of the article’s category.

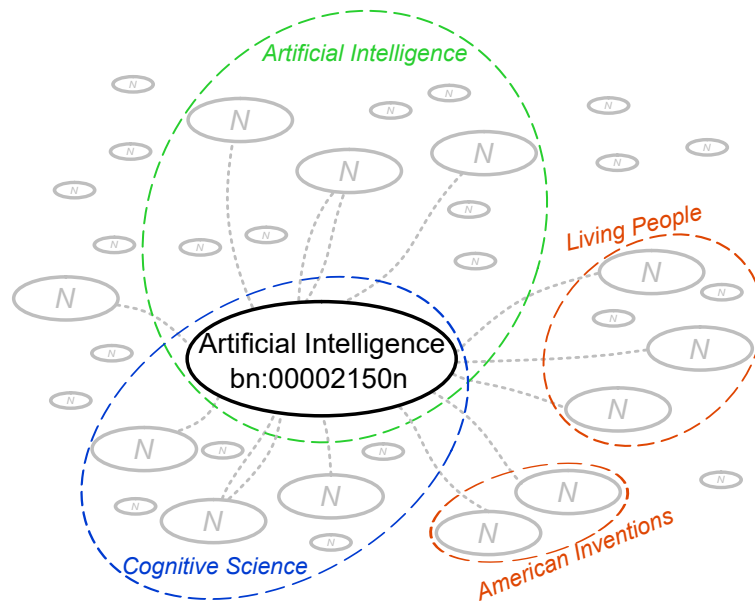


FIGURE 4.2: Neighborhood of Artificial intelligence synset.

Therefore, filtering unexpected neighbors is a need in order to avoid selecting unrelated ones and to reduce their numbers. Given that the article context has already been identified in the categorization step, only neighbors matching the expected categories are kept. Finally, over 2212 neighbors, only 30 neighbors—*belonging to 95 categories*—are kept because they match with the article context, which is much more acceptable. The most represented categories are the following:

- Cybernetics (17)
- Optimization algorithms and methods (10)
- Mathematical optimization (7)
- Systems theory (4)
- Formal sciences (3)
- Evolutionary algorithms (3)
- Search algorithms (3)
- Operations research (3)
- History of artificial intelligence (3)
- Control theory (2)

At the end, retained neighbors and their respective categories are much closer

to the articles original context. Indeed, selected neighbors such as *Machine learning*¹², *Genetic algorithm*¹³, *Mathematical optimization*¹⁴ or *Hill climbing*¹⁵ are much more related to the origin article.

Connecting articles from their neighbors is by definition less precise / safe than connecting them with common keywords, because keywords are in theory more representative of the article context. A weight has therefore been introduced in the similarity equation to make these connections more or less important (see Section 4.1.3).

4.1.3 Similarity Computation

As discussed in previous sections, disambiguated synsets and their related neighbors are used to compute similarity between two different articles. The proposed similarity score aims to quantify how much two articles are related, from a semantical point of view. Hence, a weighted sum of Jaccard indexes¹⁶ estimates the similitude between keywords and neighbors sets from an article pair.

$$sim(A_i, A_j) = \frac{1}{\alpha + \beta + \gamma} * \left(\alpha jac(K_i, K_j) + \frac{\beta}{2} jacKN(K_i, N_j, K_j) + \frac{\beta}{2} jacKN(K_j, N_i, K_i) + \gamma jacNN(N_i, N_j, K_i, K_j) \right)$$

where:

- K_x is the set of keywords' synsets of the article A_x
- N_x is the set of neighbors' synsets of the article A_x
- $jac()$, $jacKN()$ and $jacNN()$ are three jaccard index variants, respectively defined in Section 4.1.3.1, Section 4.1.3.2 and Section 4.1.3.3

(4.1)

Coefficients are set up to reflect confidence degree because α —which is the safest possible intersection while it connects papers from their common keywords' synsets—is set to 4, twice more than β which has a weight of 2 and which is twice bigger than γ and its weight of 1. With this weight distribution, keywords intersections (i.e., K_1K_2) are more important than other connections—*four seventh of the metric*—because they will probably be more obvious to users, thus more legitimate. However, other classical approaches—*keywords matching or probabilistic ones*—would also be able to recommend these articles and scholars potentially already found these articles from other scientific platforms. If we want to privilege novelty and propose to the users unknown but related papers, β and γ might be increased to the detriment of α . This weighted sum is divided by the sum of each weight in order to normalize the metric between 0 and 1. However, a balance must be found in order to not only propose farthest articles. The keyword-neighbor intersection ($jacKN()$) might be favored compared to neighbor-keyword one with the insertion of two dedicated weights (e.g., β_1 and β_2), but the metric would not be symmetric anymore.

Figure 4.3 is a theoretical illustration of the articles intersections involved in Equation 4.1. Keywords from articles A and B—*respectively on the left and right side*—are categorized and their corresponding synsets (S on the figure) are retrieved. Data is augmented and synsets' neighbors (N on the figure) are exploited—*synset-neighbor relationships are represented with dashed lines*. Finally, keywords intersection are synsets

¹²<https://babelnet.org/synset?word=bn:01647033n>

¹³<https://babelnet.org/synset?word=bn:03130158n>

¹⁴<https://babelnet.org/synset?word=bn:03309733n>

¹⁵<https://babelnet.org/synset?word=bn:02996399n>

¹⁶Jaccard index gives a similarity indicating degree of similitude among two sets. This is the division of these two sets' intersection size by their union size.

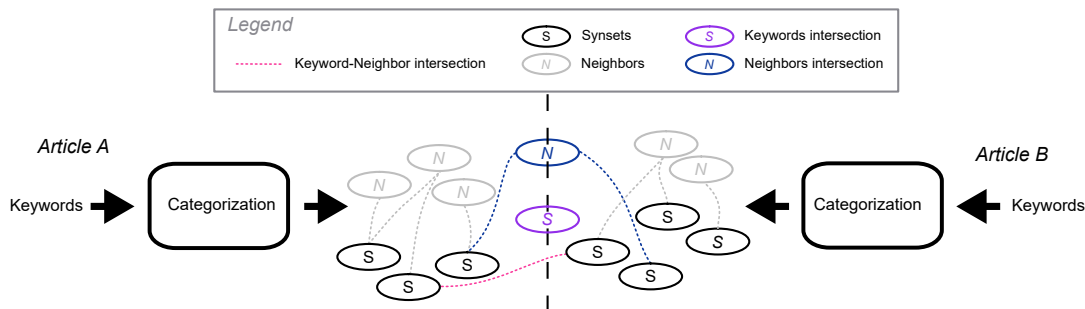


FIGURE 4.3: Theoretical similarity intersections.

sitting astride both sides (i.e., in purple), keywords-neighbors are red dashed lines crossing the symbolic middle lines, and neighbors intersections are neighbors shared by both articles (i.e., on symbolic line connected with blue dashed lines).

4.1.3.1 Keywords intersection

The safest way to identify related articles is when they share keywords in common. Actually, our approach will search for intersections among disambiguated keywords (i.e., synsets). In most of the cases, these connections could also have been proposed by traditional probabilistic approaches that any other basic search engine could have retrieved, even simplest systems using full text matches. However, it is important that our approach also propose closest articles (i.e., most related) to interested users. Indeed, readers might expect suggestions to be as related as possible especially when they use those to evaluate the accuracy of our data / approach. Suggesting the most related articles is therefore mandatory because scientists might expect suggestions to be highly related to the article they are reading, otherwise they will not look at them anymore.

$$jac(K_i, K_j) = \frac{|K_i \cap K_j|}{|K_i \cup K_j|} \quad (4.2)$$

Equation 4.2 is the Jaccard index between keywords sets of articles A_i and A_j . This reflects the similarity coefficient of these two sets and is contributing to the similarity equation (Equation 4.1). The added value of articles brought by these connections—in contrary to the ones coming from full-text exact matching methods—are that articles share contextualized keywords. This means that false positives inherited from homonyms¹⁷ are filtered out.

4.1.3.2 Keyword–Neighbor intersection

The second way to connect articles together uses neighbors of disambiguated synsets from the categorization step. Articles are connected when a neighbor of an article's keyword synset is the keyword synset from another article. Hence, these connections are already considered as semantic relationships that classical approaches will not necessarily retrieve. Keywords from which intersections with other keywords were already found (i.e. keywords intersections) are not compared with neighbors of the other article, in order to not increase the similarity score from different places.

¹⁷Homonyms are words with different meanings for exactly the same spelling

$$jacKN(K_i, N_j, K_j) = \frac{|K_i \cap N_j|}{|K_i \cup N_j| - |K_i \cap K_j|} \quad (4.3)$$

Equation 4.3 represents the similarity coefficients between keywords and neighbors sets from both articles. Their weights remain equal in order to keep the metric symmetric. This means that articles retrieved from keyword-neighbor intersections ($jacKN(K_i, N_j, K_j)$) are as important as the ones found from neighbor-keyword intersections ($jacKN(K_j, N_i, K_i)$). However, if keyword-neighbor intersections need to be favored compared to neighbor-keyword ones, simply adjust their respective weights and make the first β bigger than the second one.

4.1.3.3 Neighbors intersection

The last and farthest way to connect articles is when keywords' synsets share neighbors in common. Keywords and neighbors from which keywords and keyword-neighbor intersections were found are excluded from these sets to not boost the score with multiple connections from the same synset. When two articles share the same synsets (i.e., keywords intersection), they will obviously share neighbors in common and the neighbors Jaccard index will be higher. Suggestions coming from these intersections are the direct added value of our approach, which takes advantage from BabelNet architecture and obtains the possibility to recommend more—*previously unobtainable*—related papers. This is especially useful for narrowed field, where experts already know most of the related papers (from keywords intersections).

$$jacNN(N_i, N_j, K_i, K_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j| - (|K_i \cap N_j| + |N_i \cap K_j|)} \quad (4.4)$$

Hence, these intersections might be helpful to bring novelty to experts and help to explore broader part of the literature. Indeed, bringing novel articles treating about semantically related topic might be valuable for a researcher during the bibliographic phase.

4.1.3.4 Intersection example

The similarity equation (Equation 4.1) takes into account the three different types of intersections described above to compute the similarity score of an articles pair. Articles might share more than one type of intersections, and they all contribute to the similarity computation with different degrees of importance.

Figure 4.4 illustrates the three different ways to connect articles together from our approach. Keywords for the first article (*HIV* and *HIV testing and counseling*) and keywords from the second one (*AIDS*, *HIV* and *Cervical cancer*) are respectively on the left and right side. The categorization steps are displayed with transparency, and only selected synsets are fully printed. Keywords from the first article share the category *HIV / AIDS*. The keyword *HIV* of the second article (right side) is the unifying element because it shares the category *Sexually transmitted diseases and infections* with *Cervical cancer* and the category *HIV / AIDS* with the keyword *AIDS*. After categorizing both of these articles, 30 and 11 neighbors were extracted, respectively for *HIV* and *HIV testing and counseling* for the first article, while 38 (*HIV*¹⁸), 10 (*cervical cancer*) and 49 (*AIDS*) neighbors were matching categories of the second article.

¹⁸More neighbors are retrieved for the keyword *HIV* of the second article because it shares one extra category than the first article. Therefore, neighbors matching *Sexually transmitted diseases and infections* are also added as neighbors of this synset.

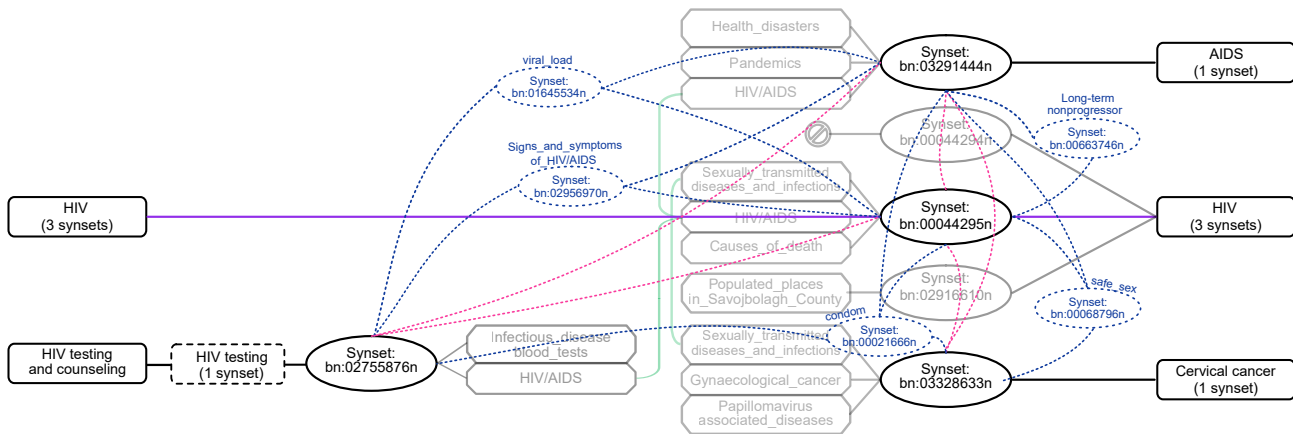


FIGURE 4.4: Articles connection from keywords' (purple), neighbors' (blue) and keyword-neighbor (pink) intersections.

Keywords intersections are represented by purple lines. From this example, both articles were sharing the keyword *HIV* (i.e., one keywords intersection). As said in previous sections, recommendations inherited from keywords intersections—even though they are the safer connections—might be partially retrieved from exact match among ambiguous keywords. However, because they might bring the most obviously related articles, it remains the highest weight of the general similarity equation in order to not only propose farthest related articles to a user and give him a bad impression about the suggestions.

Two Keyword-Neighbor intersections are displayed in pink in Figure 4.4. These relationships are well represented in this example because keywords from these articles are highly related—especially keywords from the second article (i.e., right) which are all neighbors between them. In this example, the sub-keyword *HIV testing* returned a synset—*Diagnosis of HIV/AIDS*—which is a neighbor of both *HIV* and *AIDS*, keywords of the second article. These connections are the second safest way to connect articles and are therefore the second highest weight in the similarity equation (Equation 4.1).

Given that all disambiguated synsets are highly related among these sets of keywords / neighbors, they share a lot of connections. Actually, only the synset inherited from the keyword *Cervical cancer* is not connected with keywords and keyword-neighbor intersections. However, it also shares neighbors with the sub-keyword *HIV testing*. This new neighbor intersection links the remaining unconnected synset.

Finally, the similarity score for this pair is equal to 0.21. The score is not too high because of the weights of the similarity equation which favor keywords rather than other intersections. This makes the score discriminant in regards to keyword-neighbor and keywords intersections. If equivalent weights would be allocated to α , β and γ (e.g., 1) the similarity score would be 0.25. However, if only neighbors connections are considered ($\alpha = 0$, $\beta = 0$ and $\gamma = 1$), similarity will be 0.44. Different weightings were tested and results are given in Section 4.3.1.

4.1.3.5 Summary and big data perspectives

Several pieces of program—called *processes*—were implemented in order to realize the linking of scientific articles. Figure 4.5 illustrates their scheduling. The first one categorizes articles (1). It goes through all articles, makes BabelNet's requests for each keyword, connect synsets from their categories and save connection information. Of course, the whole mapping is saved such as keywords-synsets, synsets-categories,

articles-categories relationships, in order to easily find connection details. Then the second process retrieving synsets' neighbors (2) is executed. It searches, for every saved synset, its corresponding neighbors and loop through those, filtering out the ones which do not match with any of the article's categories involving these synsets. *When a specific synset is used in several articles, it might be connected to different categories, neighbors sharing any category involving the given synset are kept as neighbors.* This process is the slowest part of our approach because it needs to retrieve synsets information (especially categories) for a very big amount of neighbors. Then keywords, synsets and neighbors identifiers (i.e., IDs) are partitioned for scalability purpose (3). Indeed, our approach was implemented with the perspective of a big data usage. This partitioning provides the flexibility to parallelize the distance calculation onto several different machines which might load a predefined articles range from files and compute the similarity score for all pairs.

The fourth and last process (4) loads this partitioned data in memory and may be used in two slightly different modes. The first one computes the top X closest articles—*i.e., the most similar ones, having the higher similarity score*—for every article and adds them to partitioned data. The other mode is saving the whole similarity matrix. This second mode is only for lab experiments from which comparisons and analysis may be realized. A few analysis results are given in Section 4.2.1.

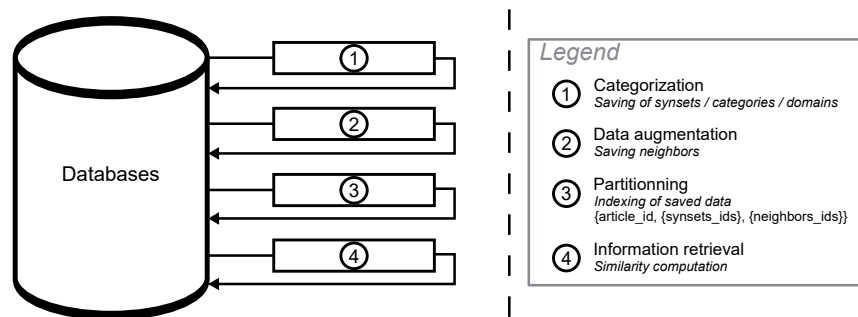


FIGURE 4.5: Global scheduling of our processes.

4.2 Evaluation

4.2.1 Offline Evaluation

Given that the offline dataset described in Section 3.2.1.1 contains 6 journals in the same field (*physical sciences*), all of their articles are potentially related. Therefore, a new lab dataset is created for a more comprehensible visual analysis. It contains three fully distinct journals, from which we expect to find similarity only for articles within the same journal. Then another journal, related to one of the three others is inserted and the goal is to find extra connection among the two similar journals.

4.2.1.1 Dataset

The three original unrelated journals are *Symmetry*, *Religions* and *Viruses*. The fourth one (*Toxins*)—close to *Viruses* scope—completes our journals selection. The dataset finally contains 3112 articles from these four journals, almost equally represented, as shown in Figure 4.6. The expected behavior is that the similarity score will be high for articles within the same journal or for articles coming from similar journals. In contrary, the similarity score across unrelated journals should be as small as possible.

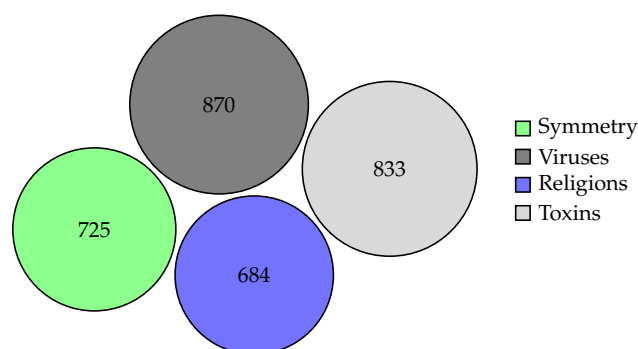


FIGURE 4.6: Offline evaluation – articles journal distribution

Before being able to compute similarity score for articles pairs, the categorization step was launched for all articles in the dataset. From this step, 1068 categories were extracted from 1389 articles (i.e., 45%)—2.09 in average (± 1.65). We obtained the mean of 3.81 (± 2.59) connected synsets per article with categories, for a total of 2668 unique synset IDs. Then, from the connected synsets, neighbors sharing one of the article's categories are extracted. This added an average of 20.4 neighbors per synsets (± 28.08) for a total of 14149 unique extra neighbors added in total.

4.2.1.2 Visualization of synsets and neighbors

The Figure 4.7 is a tag cloud of the synsets names from which categories connections were extracted. It also contains related neighbors of the connected synsets (with the prefix "n:" and in a brighter color). To make this tag cloud understandable, journal titles are plotted within the cloud. Size of the names are proportional to number of occurrences among the corpus.

Within this tag cloud, meaningful and relevant synsets (e.g., *Faith*, *Evangelicalism* or *Islam*) and neighbors (e.g., *Christ*, *Protestantism*, *Orthodoxy*) are represented for *Religions* and other journals.

Table 4.1 displays the top 2 synsets and neighbors per journal, and their number of occurrence. We may observe that some journals have higher occurrence number

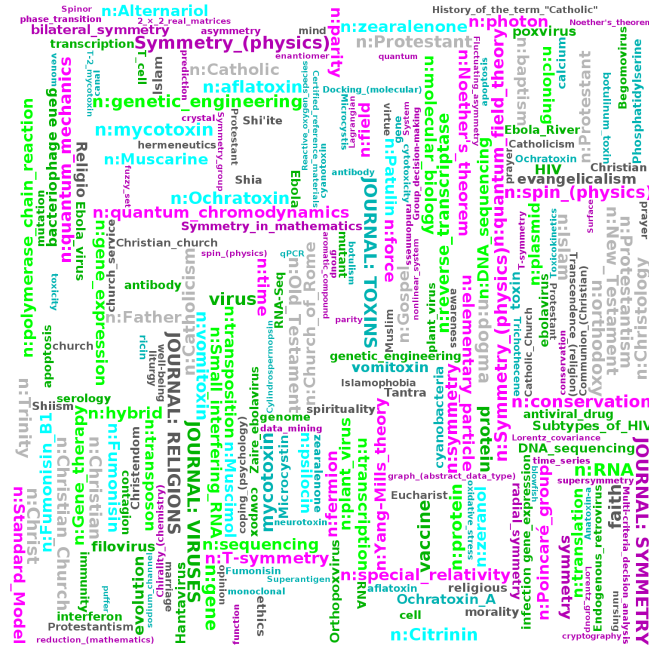


FIGURE 4.7: A tag cloud of journals' synsets and neighbors. Gray: *Religions* / Purple: *Symmetry* / Green: *Viruses* / Blue: *Toxins*

of their top synsets. For example, *Religions*—which is the smallest journal within our dataset, in terms of number of articles—has 33 articles with the synset *Faith*. In contrary, the top synset of *Viruses*—the bigger journal—only has 18 occurrences.

TABLE 4.1: Top 2 synsets and neighbors per journal

Journal	Type	Word	Occurrences
<i>Viruses</i>	synset	Vaccine	18
<i>Viruses</i>	synset	Virus	18
<i>Viruses</i>	neighbor	RNA	85
<i>Viruses</i>	neighbor	Genetic engineering	83
<i>Toxins</i>	synset	Mycotoxin	44
<i>Toxins</i>	synset	Vomitoxin	23
<i>Toxins</i>	neighbor	Citrinin	57
<i>Toxins</i>	neighbor	Fumonisin B1	52
<i>Symmetry</i>	synset	Symmetry (physics)	44
<i>Symmetry</i>	synset	Symmetry	30
<i>Symmetry</i>	neighbor	Quantum field theory	83
<i>Symmetry</i>	neighbor	Time	80
<i>Religions</i>	synset	Faith	33
<i>Religions</i>	synset	Religio	22
<i>Religions</i>	neighbor	Trinity	119
<i>Religions</i>	neighbor	Dogma	112

4.2.1.3 Distance matrix

To evaluate the similarity score from our approach and its capability to connect similar articles, we computed the distance matrix (i.e. $d(A_1, A_2) = 1 - sim(A_1, A_2)$). The same weights as the ones announced in Section 4.1.3 are used for the similarity equation (Equation 4.1). From this distance matrix, a threshold is defined to select rather or not a pair of articles is considered as valid. If the distance is lower than the threshold for articles from the same or similar journals, the prediction is considered as a true positive (TP). However, if it is above the selected threshold, this is a

false negative (FN). In contrary, when two articles from distinct journals have a low distance—*i.e.*, *below the threshold*—the pair is a false positive (FP), because they should in theory have high distance and be above the threshold in order to become a true negative (TN). Table 4.2 recapitulates the predictions classification in regards to the expected class, considered as the ground truth.

TABLE 4.2: Predictions naming. Y: Yes, N: No, TP: True Pos., FP: False Pos., TN: True Neg., FN: False Neg.

		Expected	
		Y	N
Predicted	Y	TP	FP
	N	FN	TN

Articles were inserted grouped by journal. Distance matrix for the three different journals is shown in Figure 4.8a. Colors represent the journal affiliations of the article in abscissa. The perfect prediction would only contain 3 filled squares along the diagonal and no other point elsewhere. In this example, the threshold is set to 0.98, which means that pairs with a distance bigger than this threshold are not printed in the graph. A smaller threshold will remove FP but also TP (*i.e.*, squares will be more localized but less dense). In opposite, a higher one will insert more pairs (FP and TP) and squares will be denser (previously FN will become TP) but more FP will be inserted (*i.e.*, pair for distinct journals will be displayed). This behavior is illustrated in Figure 4.8b where a threshold of 0.995 is defined.

A fourth journal (*Toxins*)—*related to Viruses*— is added to these three journals. The expected behavior is that distances for articles from these two related journals are lower than the threshold and are valid predictions. Figure 4.8c shows that this behavior is verified. Indeed, the squares located in top-left (green) and bottom-right (orange) are representing pairs of articles from these two journals which are higher than the threshold value. This validates the capability of our approach to connect related articles by their semantic relatedness. The parallel is also made with the threshold 0.995 in Figure 4.8d.

If the threshold is too small (*i.e.*, more restrictive), neighbors' and keyword-neighbor intersections will be removed, because of the α , β and γ weights from Equation 4.1. These weights make the metric brutal and discriminant when no keywords intersection is found. Therefore, it is expected that reducing the threshold will reduce FP because most keywords intersections will remain, which are the safest way to connect articles.

4.2.1.4 Metrics analysis

This approach obtains good results in terms of precision and recall for big threshold values, as shown in their respective figures (Figure 4.9a and Figure 4.9b). Precision goes from 0.994 for the smallest values of the threshold—*which means that only keywords intersections are retained*—to 0.82 for the highest threshold value—*i.e.*, *less restrictive including even neighbors intersections* with a peak at 0.997 for the threshold 0.58. Given that keywords intersections are favored in our similarity metric (*i.e.*, much higher weight than other types of intersections), pairs with higher similarity score (or lower distance value) tend to be more precise than other less secure articles pairs sharing exclusively neighbors. However, the number of retrieved articles are much lower while considering only safest connections, as shown in the recall (Figure 4.9b) and the true positives (Figure 4.9d) curves. The true positive curve may

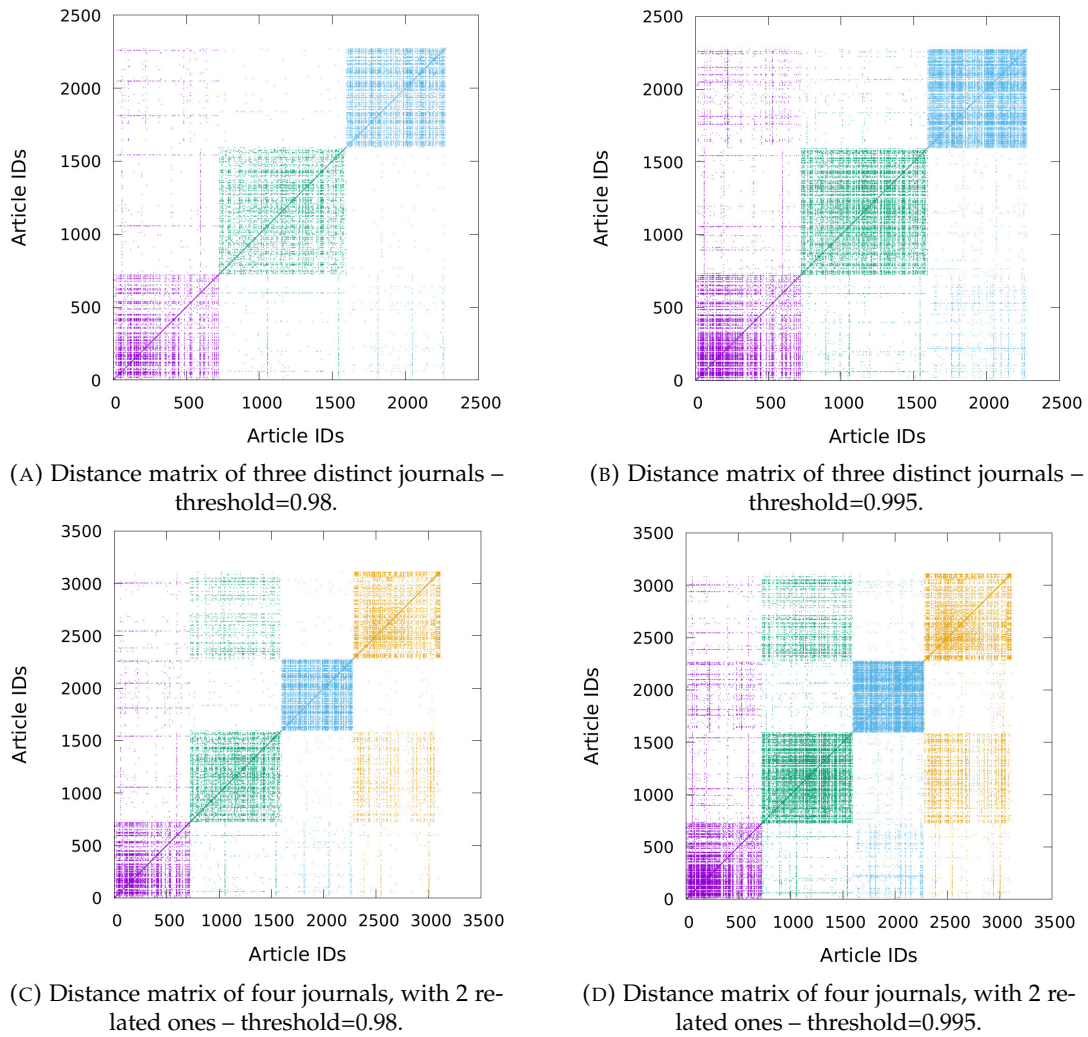


FIGURE 4.8: Distance matrices with (A) / (C) three distinct journals and (B) / (D) four journals including two related ones, for thresholds: 0.98 and 0.995. The purple, green, blue, and orange points represent articles in abscissa respectively from the journal *Symmetry*, *Viruses*, *Religions*, and *Toxins*.

also represent the coverage of this approach, from which only 0.2% (i.e., 182) of the potentially positive pairs (i.e., TP + FN) are retrieved for the first threshold values (e.g., 0.02). From this threshold, only perfect intersections pairwise are kept. Indeed, such a close distance (lower than 0.02) means that the similarity score was close to 1. In other words, articles were sharing exactly the same synsets, hence also the same neighbors.

For approaches where precision matters the most (Table 4.3), a threshold of 0.58 can be selected, which leads to a precision of 0.997 but with a recall of 0.004 (i.e., F1-score of 0.009). However, only 364 correct pairs are retrieved with this most precise threshold. The second entry of the table (i.e., threshold: 0.94) is an alternative achieving a precision of 0.98 and returns 8148 true positives.

However, when selecting the best point of the Precision/Recall curve (Figure 4.9c)

TABLE 4.3: Predictions with best precision

Threshold	Precision	Recall	F1	True positive
0.58	0.997	0.004	0.008	364
<i>best acceptable 0.94</i>	<i>0.98</i>	<i>0.092</i>	<i>0.168</i>	<i>8148</i>

which is the point with the smallest euclidean distance to the [1, 1] point, the highest threshold (i.e., 1¹⁹) is selected because it brings all potential correct articles pairs, while maintaining an acceptable precision value. In other words, given that the variation of the precision is not significant enough whereas the recall variation is exponential, selecting the threshold providing the best recall provides in this case the best results. Therefore the best compromise between precision and recall is the higher threshold values because the precision will only decrease by 17%—compared to the average precision of 0.987 for all other thresholds—whereas the recall and TP are multiplied by a factor 24—compared to average TP of 3596. At the end, the best compromise found from this curve is for a threshold equals to 1. The recall is 1 because all and all the 88'588 potentially correct pairs are retrieved, with a precision of 0.819 (see Table 4.4).

TABLE 4.4: Predictions with best precision/recall compromise

Threshold	Precision	Recall	F1	True positive
1	0.819	1	0.901	88588

Finally, the ROC (receiver operating characteristic) curve (Figure 4.9e) is also generated. This curve shows the true positive rate (TPR—Equation 4.5) by the false positive rate (FPR—Equation 4.6).

$$TPR = Recall = \frac{TP}{TP + FN} \quad (4.5)$$

$$FPR = \frac{FP}{FP + TN} \quad (4.6)$$

The threshold providing the best results is estimated from the ROC curve by finding the closest point to the [0,1] point. From our experiments, its threshold tends to be slightly smaller than the one chosen from the precision / recall curve. Table 4.5 shows the statistics of this point.

TABLE 4.5: Predictions with best ROC curve

Threshold	Precision	Recall	F1	True positive
0.995	0.918	0.44	0.595	38974

¹⁹A distance of 1.0 may occur either when no data is available from one of the article—i.e., when no data is found from the categorization step—or when articles are fully disconnected. Hence, pairs with a distance of 1.0 are not included in this analysis because all potentially correct pairs (but irretrievable) may wrongly / incomprehensibly modify the normal behavior.

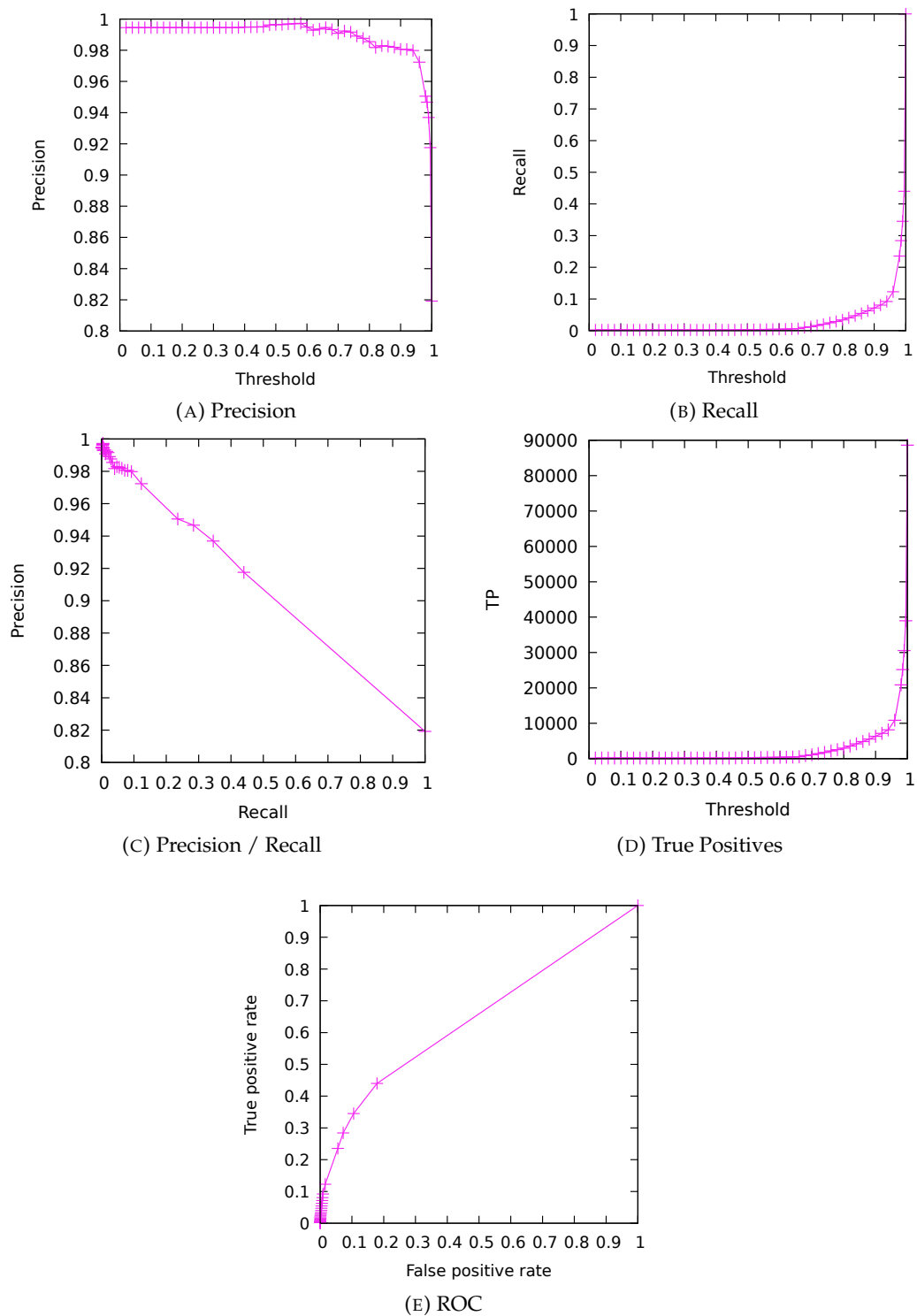


FIGURE 4.9: Metrics of the 4 journals predictions
 $(\alpha = 4, \beta = 2, \gamma = 1)$

4.2.2 Online Evaluation

Online evaluation of our similarity metric was realized on the dataset described in Section 3.2.2, from which categories and domains were evaluated by 52 experts. Out of the 27,880 categorized articles from which 19,671 unique synsets were extracted,

53,632 related words (i.e., neighbors) were added and used for the similarity computation. More details about the extraction are given in Section 3.2.2.1. Evaluators—while accessing the abstract page of an article—were asked to evaluate the quality of the related articles proposed by our approach. Actually, in order to estimate the efficiency of each part of the similarity score (Equation 4.1), the 2 closest suggestions (i.e., top2 higher ratio) of each contribution were evaluated. Indeed, the top2 articles from the intersections of keywords, keyword-neighbors and neighbors were proposed and evaluators were asked to rate those. An example of the interface for the keywords evaluation is shown in Figure 4.10.

Keywords-keywords

0.375 – Evaluation of Harmful Algal Bloom Outreach Activities
Marine Drugs, 2007

Relevancy: highly relevant relevant marginally not relevant
[Show/hide connection details](#)
 Keyword-keyword (3): [8828, 8830, 8829]

FIGURE 4.10: Evaluation–Keywords intersection , or another one with neighbors-neighbors...

Please note that non-representative categories—*counted as wrong in the categorization evaluation*—positively contribute to the related articles’ retrieval, because they affect the similarity score from Equation 4.1. Even though these categories are not the best representatives, their synsets are in relation with the article context and suggestions rank (i.e., order) might be altered when synsets from these additional categories match other articles. Therefore, suggestions matching both categories (representative and non-representative) will be ranked higher in the list and more accurate suggestions will be proposed. On the other hand, they can also bring non-representative related articles if suggestions only match with synsets from these categories. Then, articles matching by only one category will be equally ranked for relevant and non-representative connections.

4.2.2.1 Overall results

The ratings of articles recommendations are unfortunately not very comprehensible. The first reason is that evaluators have different ways to evaluate recommended articles. Most of them only read the title of the recommendations in order to estimate their relevancy and only 6% of the suggested articles were rated after accessing the abstract. An evolving evaluation protocol might be put in place (e.g., larger scale, several phases of ratings, etc – see Section 4.4.6) in order to better understand the users behaviors. Figure 4.11 shows a distribution of suggestions ratings grouped by type of suggestion (4: highly relevant, 3: relevant, 2: marginally relevant or 1: not relevant). The analysis of the results revealed that the 4-options scale implies brutal behaviors because evaluators tend to provide harder marks on small scales. This assumption may be corroborated with the example of a computer scientist reading a paper about the categorization of documents using text proximity. This expert may consider an article describing the categorization of movies from their descriptions as "marginally related" to the original paper, whereas he would rate it as 50 or 60%

similar on a 0-100 scale (with 0: not related and 100: highly related) given that linking methods are similar. For this reason, the three upper values—*marginally related*, *related* and *highly related*—are considered as positive ratings (49%). Examples of different ratings are given in the following section with the explanation about the recommendations together with a discussion whether they are legitimate or not.

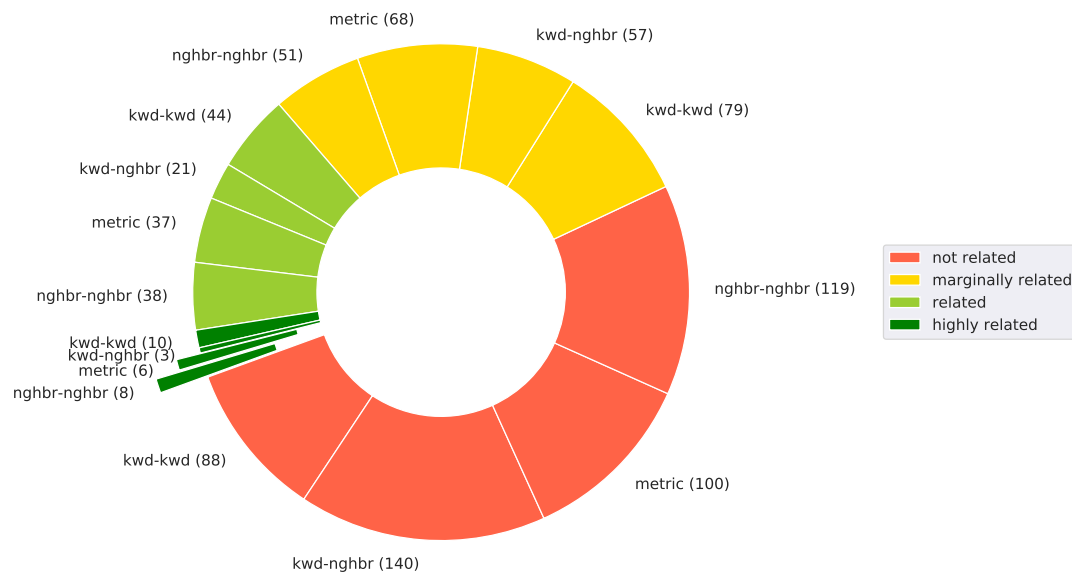


FIGURE 4.11: Evaluation-Ratings distribution

The incomprehensibility of ratings results may also be inherited from similar pairs receiving dissimilar ratings from two different evaluators, sometimes even both extremes. A representative example of both of these reasons is an article evaluating the harmful algal bloom²⁰ which is recommended for an article about dinoflagellates producing natural toxins and causing harmful algal blooms²¹. Given that they both treat *harmful algal bloom*, they share common synsets (*ciguatera fish poisoning*, *paralytic shellfish poisoning* and *neurotoxic shellfish poisoning*). Consequently, our approach considers that they are related and on the same topic. However, two evaluators rated this as *marginally related* while two other ones estimated it as *highly related* and another one rated it as *related*. In total, there were 62 pairs evaluated by at least 2 different evaluators but only 25 of those (i.e., 40%) found a perfect agreement across all ratings evaluators, and 12 (i.e., 19%) had partial agreement (only a part of the evaluators rating the same article agreed).

Another variable making these ratings incomprehensible relied on the way to extract the top2 suggestions per type of intersection. Given that the most similar articles are extracted both for similarity metric and for each of its intersection type, only the highest pairs were kept. Moreover, when a common keyword is identified, it is not compared with neighbors of the other article. Thus, the best keyword-neighbor intersection pairs have been computed independently and it represents the Jaccard distance (i.e., a ratio) of the remaining sets—for entries that did not match in above semantically intersection. However, articles pairs coming from keyword-neighbor intersections might share other keywords in common, and suggestions inherited from neighbors intersections may also share other type of intersections. Therefore, ratings may be biased in one direction or the other one. In order to make the evaluation more robust and comprehensible, suggestions should be added into the list

²⁰<http://research.scilit.net/article/9f6d463f72ba46556d0ca8fa8eb562a8>

²¹<http://research.scilit.net/article/1e4b03773f0f006e25591d2693d6ab2d>

only when there is no higher semantical intersection. In other words, if two articles share some synsets in common (i.e., keywords intersection), they should not be added into the list of articles shared in lower intersections—i.e., *keyword-neighbor*, *neighbor-keyword* and *neighbors intersections*.

4.2.2.2 Analysis

Even though the understanding of results from the evaluators ratings is not linear, four different cases have been identified. They are described here-below in their respective sections. By analogy with Table 4.2, we may consider an expected positive item as a legitimate or meaningful article suggestion and the prediction will be the rating value.

True Positives (TP) – Legitimate suggestions, positively rated.

The first class of ratings are the ones from which our approach recommended meaningful and legitimate articles to evaluators regarding the article of reference, and those rated them accordingly. In other words, a positive rating given to an expected positive suggestion. A representative example of these TPs is the suggestion proposed for articles about *harmful alga bloom* (described in Section 4.2.2.1). Articles share three synsets in common and this pair finally received 100% positive ratings (2 highly related, 2 related and 1 marginally related). Even though these cases are not necessarily the most represented in ratings, they prove the feasibility of recommending relevant articles from each type of intersections.

True Negatives (TN) – Illegitimate suggestions, negatively rated.

Some negative ratings may be negatively rated simply because they are inherited from errors / limits of our approach. Most of those are the consequence of bad / non-representative categories from the categorization step (see Section 3.4.1 for more details). This is obvious that when selected categories are wrong, their related synsets sharing them are also wrong. The same for non-representative categories; there are big risks that non-representative categories imply non-representative synsets, hence meaningless suggestions. The dynamic filtering of categories (e.g., per journal) is therefore a lead that might reduce these wrong or non-representative categories (see Section 3.4.2.2). A TN example is an article from molecular sciences describing the synthesis of a new scaffold²² and having the following (really specific) keywords: *αArylα(pyridazin3yl)acetoneitrile*; *C-arylation*; *αarylα(pyridazin3yl)acetamides*; *7H*; *8Hpyrimido[1,6b]pyridazin6*; *8diones*. Hence, only synsets from chemical elements are extracted and an article aiming to strengthen carbon foam characteristics²³ is suggested because they both share the synset of carbon chemical element²⁴. This suggestion, even though chemical elements were legitimate, deserved its four bad ratings (*not related*). The article retrieved was actually the closest within our dataset.

²²<http://research.scilit.net/article/cc4564cc9c317995bcca3dd973e273ae>

²³<http://research.scilit.net/article/eb4ed2f5640dbf4ca0a69be3e4e34b0b>

²⁴<https://babelnet.org/synset?word=bn:00006890n>

False Positives (FP) – Legitimate suggestions, negatively rated.

Recommendations might also be negatively rated whereas they are in relation with the article of reference, thus legitimate. Those might be inherited by the evaluator's rating methodology / criteria. Indeed, an expert in a specific field might be more selective than occasional reader with only basic knowledge, because experts might expect more precise suggestions, or would even think about specific articles that could be suggested. Evaluators also often did not access recommendations before rating those and estimated their relevancy by their titles. Hence, they may miss the main topic of the article or the reason why it was proposed.

The other potential false positive example might be non-representative articles. When recommendations are about the same topic but treat about different methodology or application, an evaluator might see it as not related or marginally related. A representative example may be an article evaluating the consequences of environmental noise on mental health²⁵. Our approach suggested an article discussing about the way to measure and collect noise pollution to evaluate its effect on the quality of life within smart cities²⁶. Even though these articles are legitimately related because they both treat about the same topic (i.e., noise), the evaluators ratings were heterogeneous (1 rated the recommendation as marginally related, and another one as highly related). To counter this undesired effect inherited from non-representative categories, a confidence indicator—i.e., *the number of keywords sharing those*—might be used to decrease the score of suggestions coming from these *unsafe* categories.

False Negative (FN) – Illegitimate suggestions, positively rated.

Suggestions may be positively rated by an evaluator whereas it does not match with the article.

In other words, the recommended article is not legitimate because it comes from the limits of our approach, but gets a positive rating. The article about dinoflagellates producing natural neurotoxins (presented in Section 4.2.2.1) get a recommendation for an article using botulinum neurotoxins to treat chronic migraine. Even though both articles are somehow related to neurotoxins, they are not related. However, evaluators again did not agree because two of them rated it as *not related* (i.e., TN), two ratings said *marginally related* and one evaluator found it *related*. The three positive ratings may be here considered as FN.

All cases have been identified, but the labelling of each rating will be too subjective. Moreover, given that this evaluation will probably not be as comprehensive as expected and because we have known rooms for improvements (see Section 4.4)), spending significant amount of time on this tedious manual task will not bring any benefit.

4.2.2.3 Summary

The analysis of evaluators ratings is controversial because no comprehensible conclusion can be drawn, neither positively nor negatively. However, ratings from this

²⁵<http://research.scilit.net/article/d34866e009ea3df2a14eb592eb977a6d>

²⁶<http://research.scilit.net/article/6655fe80ca045b76e2345ed736fed80b>

online evaluation pointed the capability of our approach to recommend articles from all types of intersections considered in the similarity metric. Indeed, even legitimate recommendations from neighbors intersections were confirmed by experts, which is the added value of our approach compared to classical non-semantic approach. More variables might be involved in the similarity equation such as the representativeness of the connected neighbors, or original-*ambiguous*-keywords in order to make it more general (see Section 4.4 for more details).

A new evaluation stage might be run in order to better dissociate the different types of intersection. For example, suggestions inherited from neighbors intersections must exclusively include neighbors relationships, and no higher semantical ones such as keywords or keyword-neighbor intersections. Another good way to evaluate the efficiency of recommendations from this approach would be to embed suggestions in the final website with A-B testing. Then, recommendations would be proposed alternatively from the classical approach (exact keywords matching) and from different variants of our approach (see Section 4.3.1 for variants description). Then, these approaches (and optionally other ones) could be compared via click-through rate. Another potential cause of the ratings quality is that best suggestions are only computed within a subset of 80,353 articles. This means that the closest articles are actually the less far within our dataset, thus not necessarily the best recommendations over the scientific literature.

4.3 Experiments

The development of the similarity score, in contrary to the categorization development, was more straight forward. Indeed, the similarity matrix provided satisfactory and promising results, which validated that our approach was able to link articles from the same journal. Therefore, given that the offline analysis was showing relevant connections, online evaluation was put in place because no benchmark is available for evaluating the quality of recommended scientific articles, as of our knowledge. However, comparisons were done—*or at least tested*—with different variants of our approach and also another famous probabilistic approach.

4.3.1 Variant Results

From the similarity equation (Equation 4.1), three weights (α , β and γ) might be changed in order to favor one type of intersections (respectively keywords, keyword-neighbor or neighbors). Finally, all of the following modes were tested:

TABLE 4.6: Weight variants and their legend

α	β	γ	
0	0	1	
0	1	0	
1	0	0	
1	1	1	
1	2	4	
1	4	2	
2	1	4	
2	4	1	
4	1	2	
4	2	1	

4.3.1.1 Overall analysis

In order to analyze how the different weights impact the results of our approach, the *all or nothing* logic is applied. From these variants, single parts of the equation are activated separately and the threshold is only based on the value of this part (i.e., the three first rows of the Table 4.6). Then, a variant with equal weights is tested (4th row). Finally, all combinations with a factor 2 between each weight were created (i.e., the last six rows). The similarity matrices have been created for all of these variants, and predictions statistics (from Table 4.2) are computed from incremental thresholds.

These statistics are directly used to compute the precision (Figure 4.12a), recall (Figure 4.12b) and their respective Precision/Recall curve (Figure 4.12c). The curve representing the number of TP for a given threshold (Figure 4.12d) is also displayed in order to illustrate the coverage of our approach. The F1 curve (Figure 4.13a) represent the harmonic mean of precision and recall, and is following the same tendency as the recall because it varies brutally, in contrary to the precision which only slightly decreases. Finally, the ROC curve (Figure 4.13b) is also displayed.

While focusing on the precision curve (Figure 4.12a), we may see that the most precise variant is the one considering exclusively the keywords (1-0-0 – the gray

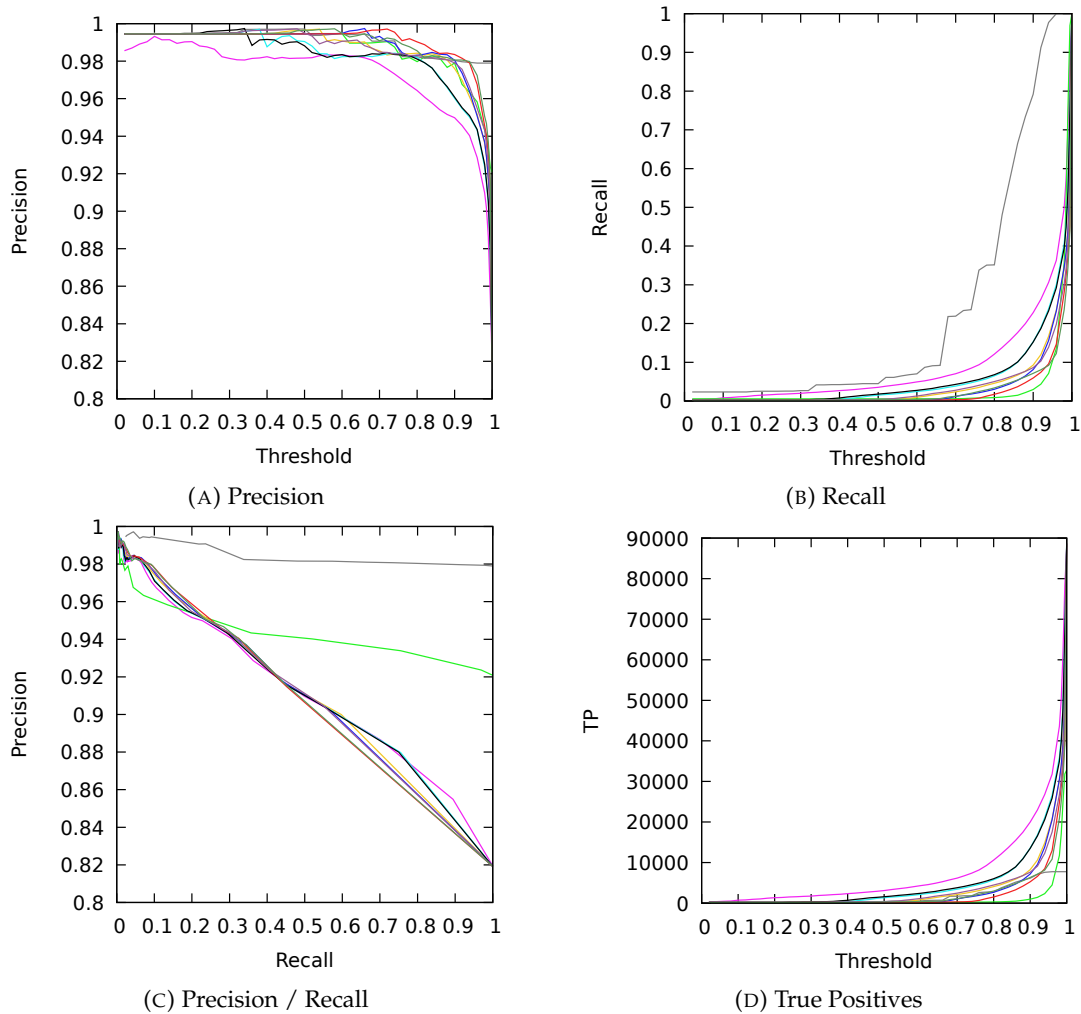


FIGURE 4.12: Metrics / Curves of the 4 journals predictions (weights variations)

line). In contrary the less precise one (pink line) is the variant where only neighbors are taken into account. These observations are totally expectable and the assumption that keywords' intersections are safer than neighbors' intersections is therefore verified. Selecting the approach bringing the best results depends on the needs and requirements of the system. If the precision of recommendations is the only criterion that matters, the highest point of the precision curves can be selected. Statistics about the most precise threshold points are displayed in Table 4.7. All variants return a best precision score between 0.993 and 0.997. In other words, all variants have a similar best precision peak. The recall column is not necessarily very relevant because recall depends on articles' presence after applying the threshold filter. Hence, the best recall for these most precise points is obtained by the variant 1-0-0—i.e., only keywords intersections considered—but this only returns 348 pairs. The most interesting information inherited from this table is that, from the most precise threshold, favoring neighbors intersections (i.e., increasing γ) seems to make the approach returning more results while being as precise as other variants.

TABLE 4.7: Most precise threshold selection.

α	β	γ	Threshold	Precision	Recall	F1	True positive
0	0	1	0.1	0.993	0.008	0.017	730
0	1	0	0.58	0.995	0.006	0.011	188
1	0	0	0.46	0.997	0.045	0.086	348
1	1	1	0.54	0.997	0.004	0.008	341
1	2	4	0.38	0.997	0.004	0.008	335
1	4	2	0.66	0.997	0.004	0.008	378
2	1	4	0.34	0.997	0.004	0.008	378
2	4	1	0.72	0.997	0.004	0.008	349
4	1	2	0.48	0.997	0.004	0.008	364
4	2	1	0.58	0.997	0.004	0.008	364

4.3.1.2 Precision / recall exploitation

If both precision and recall matter, the precision / recall curve (Figure 4.12c) might be exploited. Indeed, the point on the curve with the smallest euclidean distance to the $[1, 1]$ point is the best compromise between these two metrics. Precision curves of all variants tend to oscillate between 0.98 and 0.99 for small threshold values and decrease up to 0.82 in worse cases for the highest thresholds whereas recall stagnates below the 0.1 until the threshold 0.9 and exponentially grows until 1. Therefore, the best compromises between precision and recall are always with the higher threshold values. Selected thresholds and their statistics are given in Table 4.8. This way of selecting the best compromise is biased by the fact that a good recall does not take into consideration the coverage of the approach. Hence, the mode $1-0-0$ (exclusive keywords intersections) obtains the best precision (0.979), consequently the best F1 score too. However, it only returns 7768 correct pairs, in opposite to the exclusive neighbors intersection mode ($0-0-1$) which returns 78310 correct pairs but with a lower precision (0.855). This explains the appearance of the mode $1-0-0$ in recall curve (Figure 4.12b), where its recall is much better than other curves because only a few articles are retrieved. Consequently, the precision / recall curve (Figure 4.12c) gives the feeling that this mode provides the best results, but it returns approximately a tenth of TP compared to other approaches (see Figure (Figure 4.12d)). The mode $0-1-0$ (involving exclusively keyword-neighbor intersections) represents the balance of both other exclusive modes ($0-0-1$ and $1-0-0$) and tends to position itself in the middle of the gap of both variants, as observable in Figure 4.12c. All variants involving the three types of intersections have the same characteristics because, for the threshold 1, all article pairs will be returned and the ratio of the weights does not matter anymore. This is expectable when looking at the similarity equation (Equation 4.1), given that all intersections will affect the distance, it will always be below 1 as far as a non-null coefficient multiplies them.

4.3.1.3 ROC exploitation

Selecting the best point of each variant from ROC curves leads to similar conclusion to selecting the best point from precision / recall curves for the three exclusive variants ($0-0-1$, $0-1-0$, $1-0-0$). Indeed, considering exclusively the keywords intersections is really precise (0.981) but returns only a few correct pairs (4453)—*this is the most*

TABLE 4.8: Best point from the precision / recall curve

α	β	γ	Threshold	Precision	Recall	F1	True positive
0	0	1	0.995	0.855	0.895	0.874	78310
0	1	0	1	0.921	1	0.959	32836
1	0	0	1	0.979	1	0.989	7768
1	1	1	1	0.819	1	0.901	88588
1	2	4	1	0.819	1	0.901	88588
1	4	2	1	0.819	1	0.901	88588
2	1	4	1	0.819	1	0.901	88588
2	4	1	1	0.819	1	0.901	88588
4	1	2	1	0.819	1	0.901	88588
4	2	1	1	0.819	1	0.901	88588

precise variant. The exclusive keyword-neighbor variant loses 4% of precision but returns 17'089 TP. And the variant involving only neighbors intersections is less precise (0.886) but returns many more TP (61'489). However, selected thresholds tend to be slightly slower and therefore a little bit more restrictive. Hence, all variants return different data which are more representative of their efficiency. The best F1 score is this time achieved by the 0-0-1 variant—i.e., the exclusive neighbors intersections variant—which is the more inclusive mode. The variants 1-4-2 and 1-2-4 are interesting compromises because they respectively achieve a precision of 0.903 and 0.905 and return near 50'000 correct pairs each. The statistics of selected best points of each variant are included in Table 4.9.

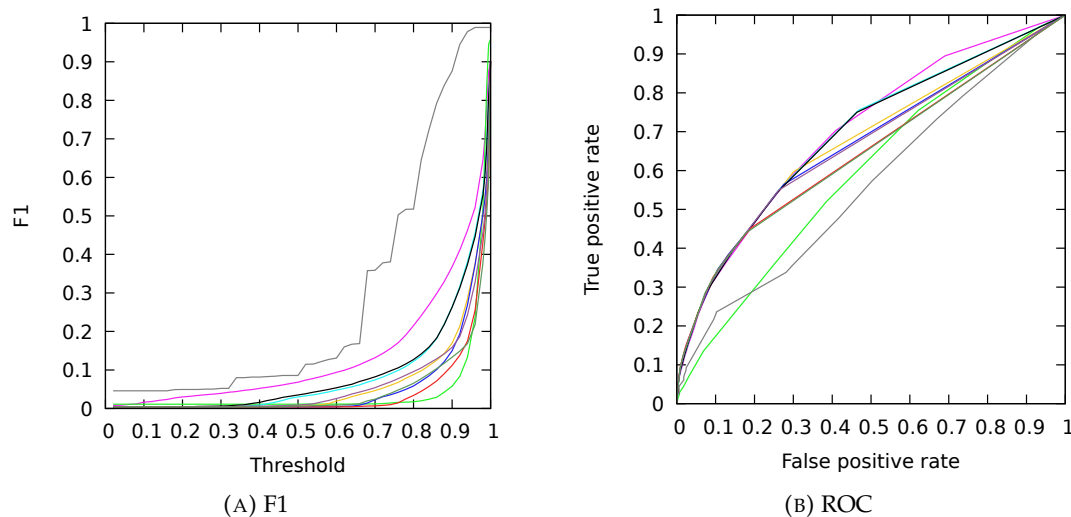


FIGURE 4.13: ROC and F1 curves of the 4 journals predictions (weights variations)

The similarity matrices of the three exclusive variants 1-0-0 (*keywords*), 0-1-0 (*keyword-neighbor / neighbor-keyword*) and 0-0-1 (*neighbors*) for the best point selected from the ROC curves are displayed respectively in Figure 4.14a, Figure 4.14b and Figure 4.14c. They corroborate the fact that articles pairs inherited from keywords intersections

TABLE 4.9: Best point from the ROC curve.

α	β	γ	Threshold	Precision	Recall	F1	True positive
0	0	1	0.99	0.886	0.702	0.784	61489
0	1	0	0.985	0.94	0.52	0.67	17089
1	0	0	0.84	0.981	0.573	0.724	4453
1	1	1	0.995	0.9	0.597	0.718	52914
1	2	4	0.99	0.905	0.549	0.684	48663
1	4	2	0.995	0.903	0.564	0.695	50001
2	1	4	0.99	0.905	0.541	0.677	47948
2	4	1	0.995	0.915	0.452	0.605	40019
4	1	2	0.995	0.905	0.549	0.684	48664
4	2	1	0.995	0.918	0.44	0.595	38974

(1-0-0) are more precise than the ones from keyword-neighbors intersections (0-1-0), which are more precise than articles pairs from neighbors intersections (0-0-1). Therefore, we define that keywords intersections are semantically higher than keyword-neighbor ones, and so on. This looks obvious because we may consider the neighbors at the bottom of the ontology tree, below the synsets. On top of the synsets sit the categories, and right above the domains. The mode embracing highest semantical intersections (i.e., keywords intersections) is more precise but only returns a few TPs, as can be seen in its sparse matrix (Figure 4.14a). By analogy, the two other modes (0-1-0 and 0-0-1)—bringing semantically lower intersections—are less precise but return much more articles (respectively 3.8 and 13.8 times more).

Finally, these three tables highlight the fact that when the precision matters the most, a more restrictive variant may be chosen—i.e., a variant with higher α value—to the detriment of the number of true positives discovered. Retrieving more true positives costs in precision but all variants embracing all types of intersections remain above a precision of 0.9, while keeping a distance below 0.995. The chosen variant for the online evaluation is the variant 4-2-1 given that it will favor safer pairs—based on keywords intersections—and obtains the best precision (0.918 at a threshold of 0.995) among all variants including all weights, and returns 38'974 correct pairs.

4.3.2 Neural Network

The online results analysis (Section 4.2.2) may point that our similarity equation (Equation 4.1) is not necessarily optimized. Indeed, taking into account many more parameters and variables could make its computation more robust. For example, the representativeness of keywords from which categories are connected, the number of keywords sharing a given category, or the number of identical keywords—among other different variables which might play a role—might be taken into account in the similarity computation. The relationship between variables, the coefficients and other equation's parameters might also be differently combined for a better representativeness. For that purpose, all potential connection variables are computed and used to feed a neural network. The goal is to train a model classifying whether two articles are related or not, based on the pre-computed connections variables. In this section, we distinguish K_x , the set of ambiguous keywords (i.e., original keywords before categorization) of an article A_x , with the set of BabelWords BW_x (i.e., the set of disambiguated synsets) and N_x , the set of neighbors.

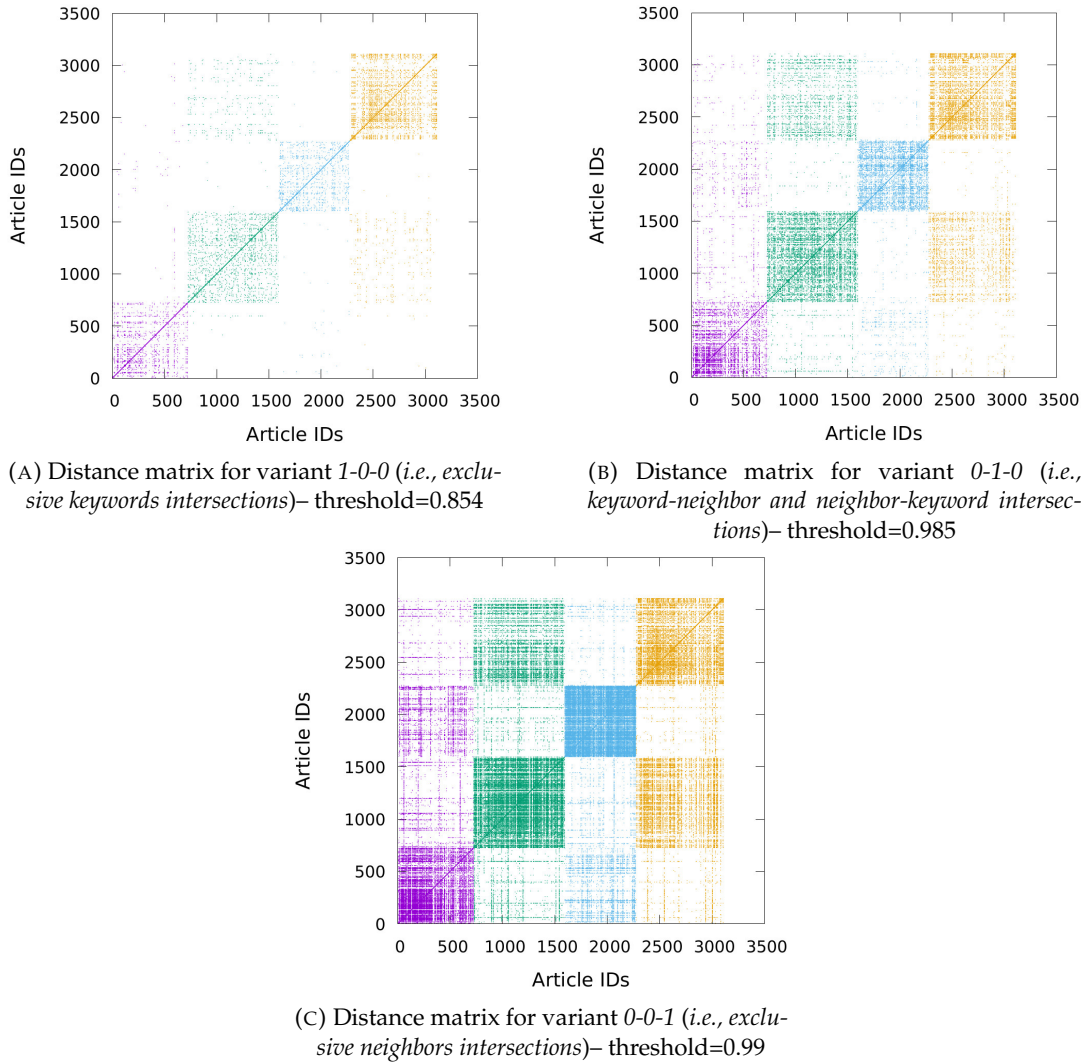


FIGURE 4.14: Distance matrices for best point from the ROC curves

Figure 4.15 illustrates the usage of the neural network with our approach. The categorization and data augmentation are the pre-treatment steps providing features as input of the network. The network learns to predict similarity regarding the journal similarity among articles pairs. In other words, if two articles belong to the same or similar journals, we assume that they are similar.

4.3.2.1 Features

Finally, the cardinality of pairwise connections (intersections and unions) of all pre-computed variables are treated and passed to the neural network. The following variables are the features given to the neural network:

- BabelWords (i.e. synsets) relationships – intersection and union of the keywords synset (i.e., what we called keywords' intersection in Equation 4.2)
 - intersection $BW_1 \cap BW_2$
 - union $BW_1 \cup BW_2$
 where BW is the set of babel words in article x (A_x).

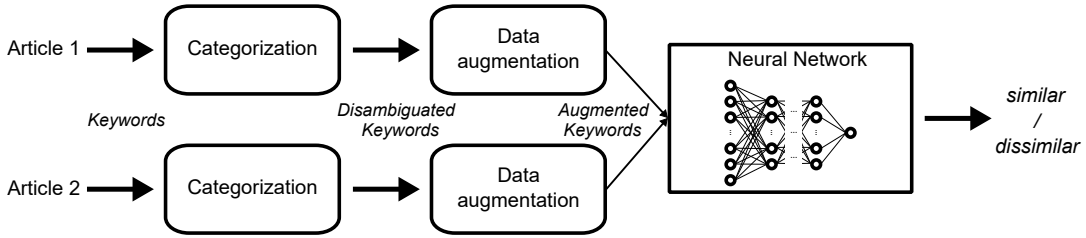


FIGURE 4.15: General workflow of our approach using neural network. Keywords are first categorized and augmented. Then, a neural network learns to predict similarity, based on their journal belonging.

- Original keywords relationships²⁷ – number of intersections and unions of original keywords from both articles.
 - intersection $K_1 \cap K_2$
 - union $K_1 \cup K_2$
- Synsets intersection confidence – the confidence / representativeness ratio of the connected synsets in regards to their best category occurrence.

$$\text{conf}(BW_i, BW_j) = \sum_{s \in (BW_i \cap BW_j)} \frac{1}{2} * \left(\frac{\text{bestCat}(A_i, s)}{|BW_i|} + \frac{\text{bestCat}(A_j, s)}{|BW_j|} \right), \quad (4.7)$$

where $\text{bestCat}(A_x, s)$ returns the highest category weight for s in (A_x)

The category weight is the number times a category occurs in different synsets. This ratio may represent a mean weighted category value of connected synsets. This ratio will be higher if the connecting synsets are from a highly represented category than if they are from a category shared by only 2 items.

- BabelWords-Neighbors relationships – intersection and union between synsets from the first article and neighbors from the second article (Equation 4.3)
 - intersection $BW_1 \cap N_2$
 - union $BW_1 \cup N_2$
- Neighbors-BabelWords relationships – intersection and union between neighbors from the first article and synsets from the second article (Equation 4.3)
 - intersection $N_1 \cap BW_2$
 - union $N_1 \cup BW_2$
- Neighbors relationships – intersection and union between neighbors from both articles (Equation 4.4)
 - intersection $N_1 \cap N_2$
 - union $N_1 \cup N_2$
- Number of original keywords – the number of **ambiguous** keywords (i.e., authors' or computed keywords) for each article. This data might be used to determine whether the intersection per union ratio is representative of the overall number of keywords.
 - Number of \mathcal{A}_{K_1}
 - Number of \mathcal{A}_{K_2}

²⁷Original keywords are the ambiguous article's keywords. Even keywords without synsets are taken into account. The number of keywords of both articles in the pair and the number of their intersections will be used to train the network in order to increase the recall and its capabilities to compute similarity even for articles without BabelNet entries.

- Number of keywords with synsets – *the number of disambiguated keywords for each article. This represents the number of keywords from which synsets sharing common categories have been found in the categorization step.*
 - Number of BW_1
 - Number of BW_2

4.3.2.2 Dataset statistics

In total, the dataset embraces 141'920 pairwise connections from the offline dataset (Section 4.2.1.1). Connections cardinalities (intersections and unions) are computed and used to train the neural network. This dataset represents all pairs of articles sharing some keywords, BabelWords (i.e., BabelNet synsets) or neighbors in common. In this analysis, articles without synsets are also considered because the goal of this experiment is to obtain either an equation or the binary classifier model recommending related articles. Table 4.10 provides the dataset pairwise statistics. The first line displays the number of articles with at least one intersection type, the second line gives the number of pairs from which no higher types of intersection is found. Keywords intersections ($K_1 \cap K_2$) are considered as higher than BabelWords (i.e., synsets) ones ($BW_1 \cap BW_2$), which are higher than BabelWord-Neighbor intersections ($BW_1 \cap N_2 / N_1 \cap BW_2$), and finally neighbors intersections ($N_1 \cap N_2$) are the lowest ones. $BW_1 \cap N_2$ and $N_1 \cap BW_2$ are grouped because there is a huge overlap, hence it does not make sense to differentiate them.

TABLE 4.10: Distribution of article pairs according to their type of intersection within the entire dataset

Number of pairs / type	Total	K_1-K_2	S_1-S_2	S_1-N_2 / N_1-S_2	N_1-N_2
Count	141,920	26,291	7461	39,964	118,194
Grouped by higher intersections	141,920	26291	5381	32,750	77,498

To train the neural network, the expected predictions are set to 1 for pairs of articles from similar and related journals. For any other pair, the expected value is set to 0. Given the fact that these journals should be really distinct, we assume that none of their articles should be linked. Finally, the model will be a simple binary classifier predicting rather two articles are similar or not.

The split of the dataset into train and test subsets slightly affects the performance of the neural network. Therefore results may be different with a different distribution of train/test sets. For this reason, the splitting of sets is realized once and the same sets are re-used in every experiment in order to find an architecture performing well with our data, independently to the split.

4.3.2.3 Perceptron

The first model to be tested is the most basic neural network: the perceptron [105]. This binary classifier uses only one neuron to learn the linear relationships (i.e., weights) among all features and adjust them at each iteration, in order to reduce the loss of the model and increase its accuracy. The train set represents 70% of our data while the test set is the remaining 30%.

The perceptron accuracy oscillates between 0.70 and 0.83, depending on the training hyperparameters (such as early stopping, number of epochs) and also the split

of the data. The perceptron achieves an average accuracy of 0.7781 (± 0.0481) on different split of our data, with its best accuracy of 0.83%. The early stopping option is activated and the learning stops when the tolerance (i.e., stopping criterion) does not exceed 0.01 for the last 10 iterations ($loss > previous_loss - tol$). The perceptron learning phase converges after 15 epochs in average. Table 4.11 gives statistics about the 12,619 wrongly predicted pairs, regarding the type of intersection found. Articles were grouped by their higher type of intersection.

TABLE 4.11: Distribution of article pairs (and accuracy) according to their higher type of intersection within the test set and the set of bad/good predictions

Number of pairs / type	Total	K_1-K_2	S_1-S_2	S_1-N_2 / N_1-S_2	N_1-N_2
Test set (TS)	42,576	7920	1623	9760	23,273
Bad predictions	12,619	321	189	1525	10,584
Correct predictions	29,957	7599	1434	8235	12,689
Accuracy	0.7036	0.9595	0.8835	0.8437	0.5452

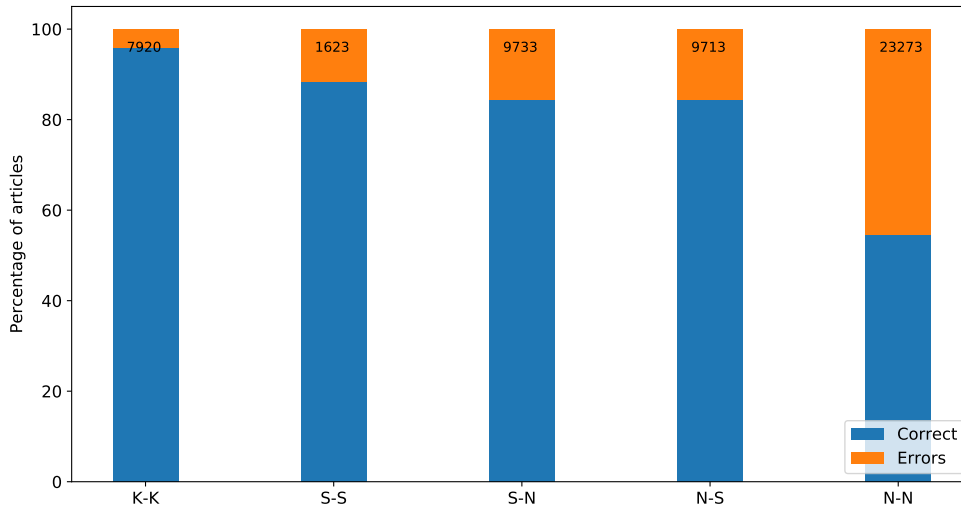


FIGURE 4.16: Perceptron predictions accurateness percentage.

Figure 4.16 shows the percentage of correct / wrong predictions on the test set. Number of total predictions proposed per type by the network are included on top of the bars. The figure highlights that the error rate increases when predictions are less safe. This is expected given that suggestions based on neighbors intersections are less safe than the ones relying on synset-neighbor intersections, which are less safe than synsets-synsets, which are less safe than keywords intersections. Here, keywords intersections are exact matches from articles pair. Those were not previously exploited, but given that they may play a role in the articles connection, they are included to the dataset. Finally, our assumption that the more confident the connection, the safer the prediction is verified.

In order to further understand when the network provides correct or wrong predictions, its similarity matrix is displayed in Figure 4.17b. This provides insights about the expected and predicted similarity pairwise correlation. Figure 4.17a. is the similarity matrix of the training set, showing similar or dissimilar pairs, based on

similar journals belonging (articles within the same journal or related one are considered as similar). TP, TN, FP and FN are described in Table 4.2 from Section 4.2.1.3. Please, note that the articles are in different order here and similar journals are at the extremity of the matrix.

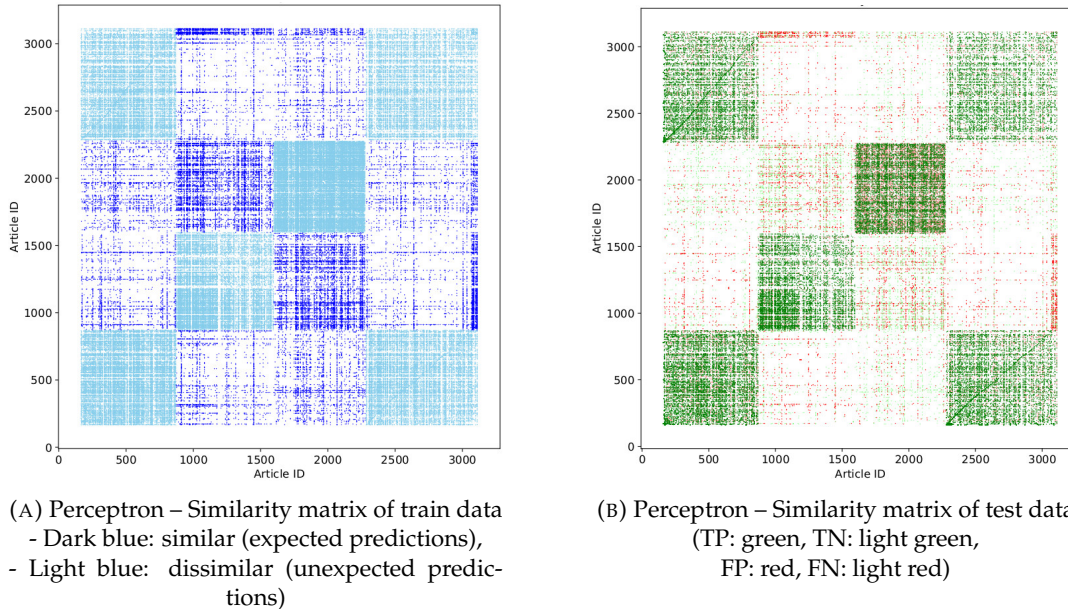


FIGURE 4.17: Perceptron – Similarity matrices of train and test data sets.

These images show that the set used to trained the model provides a significant number of dissimilar articles, sharing some intersections. Indeed, 23% of the pairs from the training set are dissimilar pairs sharing semantic intersections. Also, there are 9652 FN (22.5%), 2967 FP (7%), 25,785 TP (60.5%) and 4172 TN (10%), for a global accuracy of 0.7036.

However, while using the same learning attributes the accuracy significantly varies over different split of our dataset. This highlights the fact that our hyperparameters are only tuned to bring the best results possible for a dedicated set. Hence, the perceptron seems to show its limit to be general enough to properly provide predictions above 80% accuracy. Therefore, there is probably some non-linear relationships among variables.

4.3.2.4 Multi-Layer Perceptron

In order to improve the accuracy of our model, more *non-linear* relationships among features in input must be computed. For that, several different multi-layer perceptrons (MLP) architectures are implemented and compared in order to identify MLP which brings the best results. Playing with the hyperparameters such as the number of neurons, layers or iterations is a needed part in order to find the optimized model for our classification problem.

Given that the network takes in entry the connection details of our approach, no optimal architecture is available in the literature and an investigation is needed to identify the best one. For that, different MLPs from sklearn²⁸ are trained, and several network architectures are tested. All combinations inherited from the following network parameters are tested, which leads to 225 different networks:

²⁸https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

- number of layers: 1-16
- number of neurons: 1:6:1, 10:101:10
- number of epoch: 200
- early stopping: activated

By taking the default MLP settings offered by the library (i.e., activation="relu", solver="adam", batch_size="auto", max_iter=200, etc), the architecture of 7 layers of 90 neurons performs the best, in equal measures of 10 layers of 80 neurons and 9 layers of 100 neurons. They all achieved an accuracy of 0.882. Below 5 neurons per layer, the number of layers does not significantly affect the performances of the models and their accuracies oscillate between 0.836 and 0.848. An MLP of one layer of 10 neurons achieves an accuracy of 0.85, and has a much easier architecture than the network achieving the highest accuracy (7 layers, 90 neurons). Also, a 2-layer network of 90 neurons seems to be one of the best trade-off between accuracy and network depth, because it provides an accuracy of 0.876 and only 35 networks perform slightly better. Adding one more layer of 90 neurons increases the accuracy to 0.88 and only 7 architectures provide tiny better results (maximum, +0.002).

Given that different hyperparameters might also affect the accuracy of our networks, a random search strategy [15] based on a 5-fold cross-validation estimates the best combinations of the following hyperparameters ranges:

- solver: lbfgs, sgd, adam
- alpha: 0.0001, 0.001, 0.01, 0.1
- batch size: auto, 200, 500, 1000
- learning rate init: 0.001, 0.01, 0.1

These hyperparameters are tested on 1-, 2- and 3-layer networks of 8, 16, 32, 64, 128 neurons per layer. The best results are obtained by the network with three layers of 128 neurons with an accuracy of 0.8794 (± 0.0031) with the 5-fold cross-validation [19] on the training set and an accuracy of 0.8841 on the test set. All best results use Adam [60] solver—the library recommends the usage of adam for datasets with more than thousands samples—and a learning rate of 0.001—i.e., the stepping of weights update variation.

Finally, this selected architecture achieves a constant accuracy of 0.8837 (± 0.002) on different train/test splits, which confirms the ability of our method to generalize well. Table 4.12 provides statistics about good and wrong predictions of this selected network (on the same test set as the one used by the perceptron in Section 4.3.2.3).

TABLE 4.12: Distribution of bad/good predictions and accuracy of the MLP results

Number of pairs / type	Total	K_1-K_2	S_1-S_2	S_1-N_2 / N_1-S_2	N_1-N_2
Bad predictions	5043	267	109	820	3847
Correct predictions	37,533	7657	1514	8940	19,426
Accuracy	0.8816	0.9663	0.9451	0.9121	0.8356

Figure 4.18 shows the predictions accurateness, where 88.16% of the predictions are correct. The same tendency as the perceptron's behavior is observed, where safest intersection types bring more accurate predictions.

Given that the training set is the same as the one used for the perceptron, the similarity matrix is already displayed in Figure 4.17a. Figure 4.19 displays the accuracy of the network in regard to the pairwise classes expectation. We may observe that

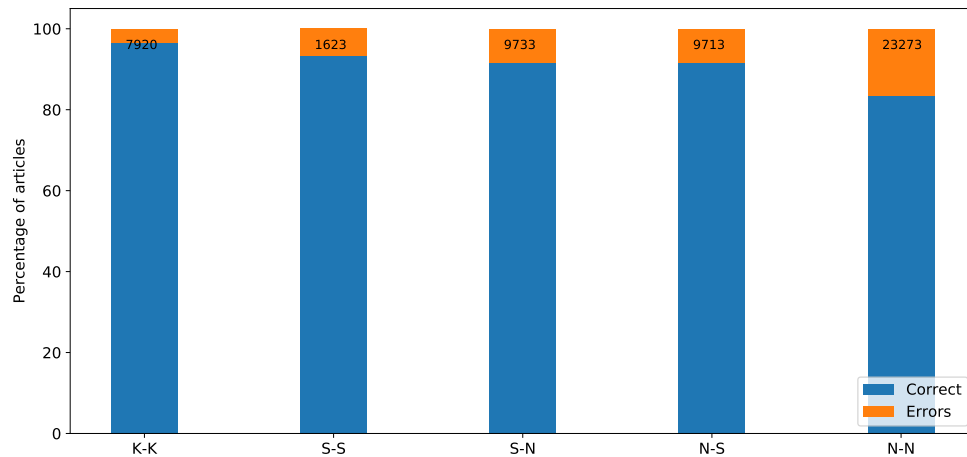


FIGURE 4.18: MLP predictions accurateness percentage.

more correct predictions because there are 1372 FN (3%), 3671 FP (9%), 34,065 TP (80%) and 3468 TN (8%), for a global accuracy of 0.8116 (vs. 0.7036 for perceptron).

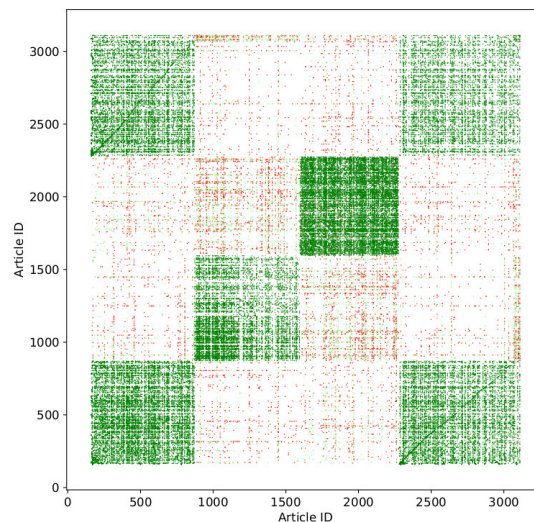


FIGURE 4.19: MLP – Similarity matrix of test data
(TP: green, TN: light green,
FP: red, FN: light red)

4.3.3 Probabilistic Methods – Word2Vec

Our favorite potential competitor would be an approach exploiting word embeddings from Word2Vec (see Section 2.1.4) because it could be used in a very close way as how our approach uses BabelNet²⁹. A search could be done for every splitted keyword and each of those will receive a 300D vector in return. Then the next step would be to compute similarity among vectors coming from different articles' keywords. Our approach uses the model trained on Google News Dataset³⁰–100 billion words–containing vectors for 3 millions words, as well as its minified version³¹ of 300,000 words.

²⁹The default mode 4-2-1 (i.e., favoring keywords intersections) is used as a baseline

³⁰<https://code.google.com/archive/p/word2vec/>

³¹<https://github.com/eyaler/word2vec-slim>

4.3.3.1 Word Mover Distance

For the similarity computation, the combination of Word2Vec together with the word mover distance (WMD – see Section 2.2.5) has been implemented and tested on the offline dataset described in Section 4.2.1.1. Keywords are exploited in order to compare approaches while dealing with the same data. For that, stopwords are removed, keywords are splitted on spaces and vectors from splitted words (i.e., sub-keywords) are retrieved from Word2Vec. Finally, the WMD is computed with resulting keywords embeddings vectors of each articles pair.

4.3.3.2 Comparison

The first difference between WMD approach and ours is that our approach is normalized and has an upper bound of 1, whereas WMD does not guarantee any limit. Its highest score was around 5.8 (and 1.5 for the slim version).

Overall.

The WMD approach, based on Word2Vec vectors, is an interesting way to retrieve related articles given that it also finds connections among articles within the same/similar journals.

However, curves analysis is not practical because WMD is able to find most of the expected pairs for high threshold values—*every articles pair, even unrelated, is connected with WMD*—but reach a non acceptable precision (i.e., 1.94 millions of TP for a $T=5.8$, but with a precision of 0.40). Hence, when an acceptable precision is obtained (i.e., above 0.80), the recall is too low (0.06 at a threshold of 3.48) because of the number of potential TPs brought by these far and unsafe connections. For this reason, WMD similarity matrix was filtered and only pairs with a distance lower than 3.36 were kept—*this threshold was determined by the latest moment WMD approach went below the minimum acceptable precision of 0.80*—(1.27 for slim version). In other words, we consider pairs with a higher distance than these specific thresholds as always negative, hence they are not taken into consideration anymore (this filtering logic is illustrated in Figure 4.20a and Figure 4.20b). This significantly increased the recall for low threshold values, and numbers are more comparable. But because the recall of this mode is relative to the determined cut-off threshold, comparing precision and number of TP (rather than recall) makes more sense.

Precision curves (Figure 4.20a) have the same tendency for both approaches. Although thresholds are not really comparable³² given that they do not represent similar distance, respective distance values are kept.

Table 4.13 shows the evaluation metrics for the WMD approaches achieving the most precise results, together with our proposed approach (described in Section 4.1.3 – (4-2-1)). The best precision values are comparable, but with an advantage for WMD with slim model in terms of number of TP retrieved. Recall is slightly better for the proposed approach (i.e., BabelNet) given that there is much less potential TPs in this mode.

³²A distance of 0.8 from BabelNet does not have a unique value on WMD, given that approaches work with different data to compute similarity. Therefore, when our approach returns $sim(A_1, A_2) = 0.8$, WMD might return $WMD(A_1, A_2) = 0.6$. However, it might also return $WMD(A_1, A_3) = 1.2$ when $sim(A_1, A_3) = 0.8$.

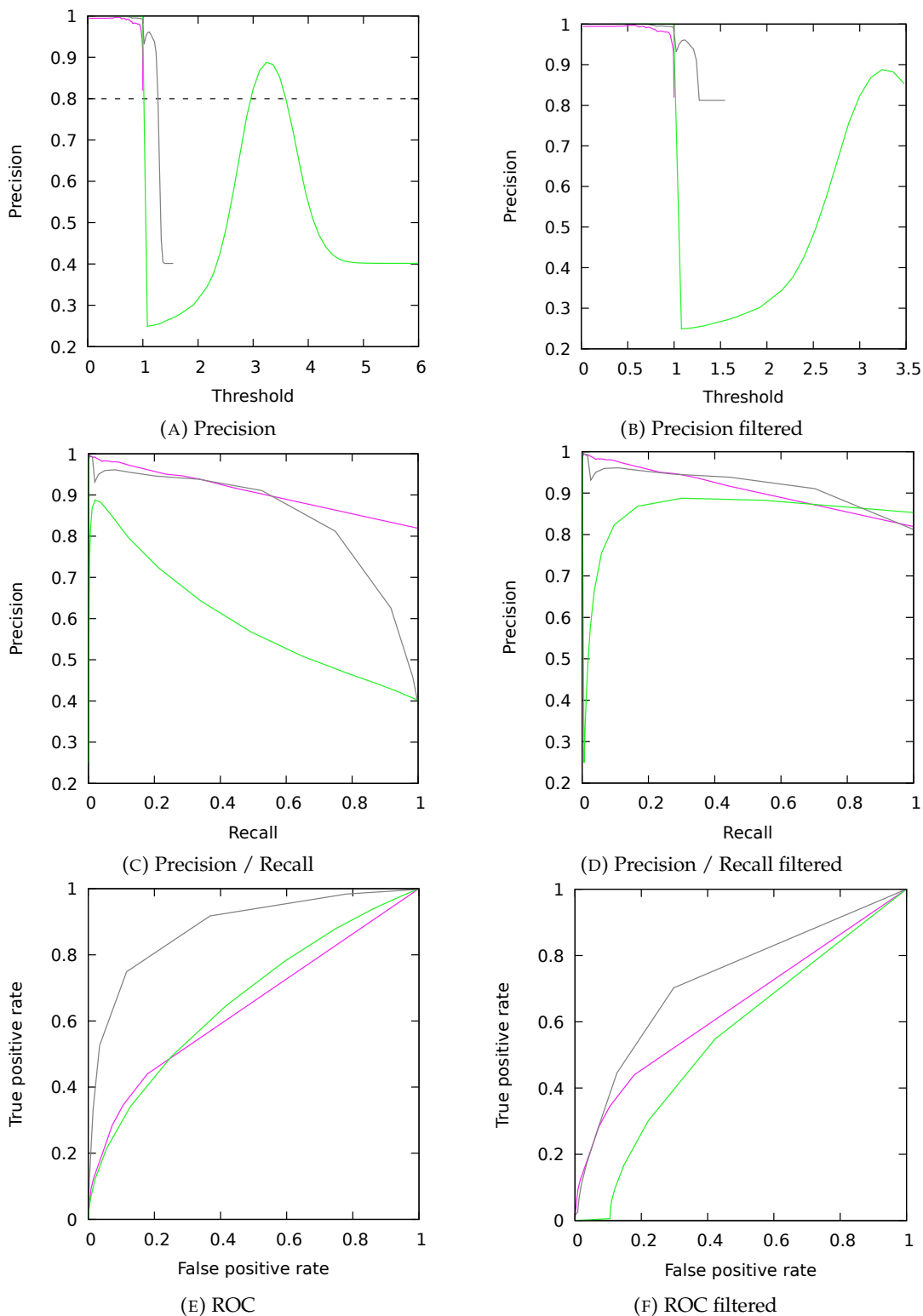


FIGURE 4.20: Curves for our approach (pink line) and WMD ones. WMD approaches are based on Google News Word2Vec model (gray line) and slim one (green line)

While selecting the best results from the precision / recall curve, WMD is more precise than our approach— 0.853 vs. 0.819 —and WMD with slim model (i.e., WMD-slim) is slightly less precise. Indeed, it brings 130,931

TABLE 4.13: Word2Vec-WMD vs. BabelNet – Predictions with best precision.

Mode	Threshold	Precision	Recall	F1	True positive
Proposed approach (4-2-1)	0.58	0.997	0.004	0.008	364
W2V-WMD	0.995	1	0	0	53
W2V-WMD (slim)	0.713	1	0	0	8001

TABLE 4.14: Word2Vec-WMD vs. BabelNet – Best point from the precision / recall curve.

Mode	Threshold	Precision	Recall	F1	True positive
Proposed approach (4-2-1)	1	0.819	1	0.901	88588
W2V-WMD	3.48	0.853	1	0.921	130931
W2V-WMD (slim)	1.271	0.812	1	0.896	1448793

correct pairs, representing +48% of the 88,588 correct pairs returned by BabelNet approach, from its threshold providing the best compromise. However, the WMD-slim version also retrieves 335'235 TPs, which represents +1535% compared to BabelNet approach (i.e., 19,550), and +1006% compared to WMD (i.e.,). Of course, given that precisions of approaches are in the same range, the ratio of false positives also scales in the same proportions and WMD-slim proposes 335,235 wrong pairs, respectively +1615% of the 19,550 BabelNet FPs and +1286 of the 22,561 WMD ones. The WMD similarity matrix (Figure 4.21) illustrates the predicted pairs for a threshold of 1.271. The number of wrong predicted values are clearly represented in this matrix, even though the expected squares are filled. Hence, we can conclude that bringing more TPs does not necessarily mean that suggestions are better.

TABLE 4.15: Word2Vec-WMD vs. BabelNet – Best point from the ROC curve.

Mode	Threshold	Precision	Recall	F1	True positive
Proposed approach (4-2-1)	0.995	0.918	0.44	0.595	38974
W2V-WMD	3.36	0.883	0.548	0.676	71742
W2V-WMD (slim)	1.24	0.911	0.702	0.793	1017504

When thresholds providing the best points from the ROC curves (Figure 4.20e) are selected, our approach achieves a slightly better precision (0.918 vs. 0.883) but again WMD brings more TP (+84%). WMD-slim approach is again close to our approach in terms of precision—*slightly lower actually*—but retrieves many more TPs. Metrics of the three modes are given in Table 4.15. However, looking at WMD-slim distance matrix at this level of threshold reveals that the noise (i.e., bad predictions) is omnipresent (see Figure 4.21). Given that our approach only categorizes 45% of the articles (see Section 3.2.1.2), predictions can only be done from these categorized articles. Therefore, improving the coverage of the categorization step might help to reduce the difference of TPs from both approaches and make BabelNet one perform better than WMD approaches.

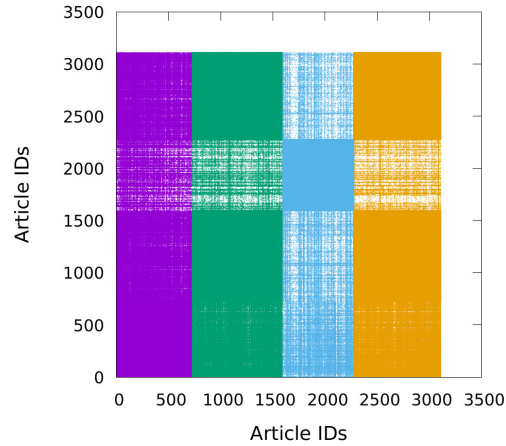


FIGURE 4.21: Distance matrix for WMD using Word2Vec slim model
– threshold: 1.271 (best point from ROC curve)

Computation time.

The WMD computation is the slowest part given that the cosine distance needs to be computed for each vectors combination of every articles pair. Consequently, its computation time is exponentially proportional to the number of articles—*i.e.*, *half of the number of articles squared (because only one half of the symmetric matrix needs to be computed)*. The computer time announced in this section is estimated from experiments run on a laptop with 8 Go or RAM and 4 CPUs. In contrary, the slowest part of our approach is the neighbors extraction—*i.e.*, *when neighbors are retrieved for every synset from the categorization step*—and its time is proportional to the number of articles to process. It takes in average 1.45 seconds per article, so the extraction of X articles takes $\left[1.45 * X\right]$ seconds (*i.e.*, 1680 days for 100 millions articles). The distance computation is very fast because it only compares integers³³, hence its computation time is negligible. Figure 4.22 shows the scalability of both approaches. It can be observed that WMD approaches do not scale properly. The total time has been extrapolated from the empirical testings. The time taken for the WMD of a pair of articles (t_{WMD}) is $8.55 * 10^{-4}$ second (or $1.149 * 10^{-3}$ second for the slim version³⁴). Therefore we estimate that the function computing all WMDs for a distance matrix of X articles (*i.e.*, a $X * X$ matrix) takes $\left[8.55 * 10^{-4} * \frac{X^2}{2}\right]$ seconds (*i.e.*, more than 100,000 **years** for 100 millions articles). Hence, the computation of the distance matrix will be faster than WMD for our approach after 2524 articles (and after 3392 articles when WMD relies on slim model) from the theoretical curve intersection calculation. For 10,000 articles, our approach will be approximately 4 times faster (3 times for slim model). This difference grows proportionally to the number of articles and our approach would finally be around 40,000 times faster (30,000 times for slim model) for 100 millions of articles.

³³The similarity (Section 4.1.3) is computed with IDs of the partitioned files (see Section 4.1.3.5)

³⁴The WMD takes less time when it relies on slim model because it tends to retrieve less vectors, thus there is less potential combinations of vectors.

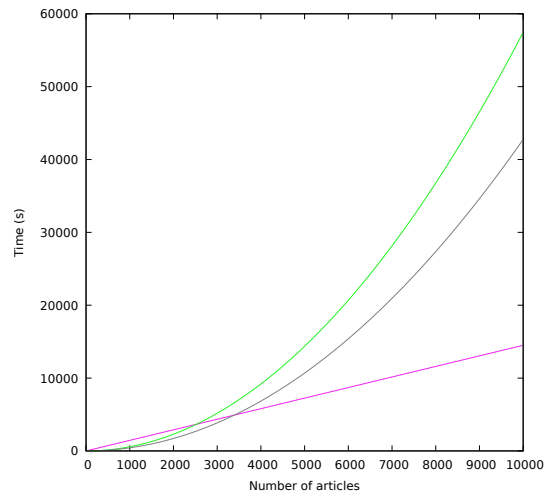


FIGURE 4.22: Extrapolated time for approach scalability for WMD-W2V (gray), WMD-W2V (slim – green) and our approach (pink)

Strength.

Word2Vec finds vectors for 79% of the single-word keywords in entry. Consequently, at least 1 vector is obtained for 99.94% of the articles in our dataset (i.e., only 2 did not have any). Therefore, WMD is computed for all pairs embracing two articles with vectors, which leads to a dense matrix, compared to BabelNet approach which can only compute similarity for categorized articles (i.e., 45%). This makes this approach retrieve many more correct articles pairs (TP) while being as precise as our approach.

Drawback.

However, its main drawback is the time it takes and the storing space it needs. In order to save some computation time, keyword vectors are pre-fetched and partitioned so that words' vectors do not need to be requested each time an article is checked. This is the same logic as our approach, from which synsets IDs are saved (see Section 4.1.3.5). The average space used to serialize 1000 articles is 295Kb for our approach against 11Mb for Word2Vec vectors (i.e. +3620%). To serialize the 115 millions of articles from Scilit, our approach will use 33.9G (i.e., loadable on one server) when Word2Vec would need 1.3T. Therefore, the WMD would be even slower because the program would need file accesses, or it would need lots of servers to compute.

Moreover, WMD is slow even though vectors are pre-fetched from the word embeddings model and partitioned. Therefore, even without taking into account the time that would take the word2Vec vectors extraction—the same requests need to be done for one article's keywords each time it's compared with others—the WMD by itself is already too slow. Finally, this implementation is not scalable, as shown in Figure 4.22.

Summary.

Even though WMD approach follows ours in terms of precision and number of correct pairs (TP) for small threshold values, it may be much

more general and find connections from any article pairs. Figure 4.21 shows that when the threshold exceeds a certain level, WMD finds relationships among all articles. The perfect threshold is not obvious to define given that it will depend on our expectations and on the data in entry. Finally, there is probably a little advantage for the WMD thanks to its better coverage of the articles landscape, but to the detriment of the processing time. Therefore, our approach has the ambition to improve this covering part (see Section 4.4) in order to better compete, even outperform WMD approach based on Word2Vec.

4.3.4 Further Investigation of BabelNet Tree

The coverage is the main drawback of the categorization step. Given that it depends on the capability to connect keywords' synsets by their categories, the level of the ontology leaf from which synsets are retrieved may affect their connection. Hence, using synsets hyponyms/hypernyms to identify connections is questionable, and this lead was explored. The conclusion of the hypernyms tree discovering is that an infinite loop appears, and the same constant 7 generic synsets (*Psychological feature*³⁵, *Abstraction*³⁶, *Entity*³⁷, *Philosophy*³⁸, *Humanities*³⁹, *Discipline*⁴⁰, *Knowledge*⁴¹) repeat again and again through this tree. Therefore, the decision to use neither hypernyms nor hyponyms is taken as we consider them as not usable. Hypernyms are too dangerous to exploit because there is no indication about the level in the ontology tree from which a synset is inherited—and *generic synsets might be used*—and hyponyms will lead to too many potential candidates, which should be even more specific than synsets.

4.3.5 Summary

The comparison of our approach, its different variants and one reputed probabilistic approach (i.e., Word Mover Distance, based on Word2Vec embeddings) was discussed in this section. This comparison highlighted that—even though there are still rooms for improvements—our approach manages to compete with the WMD (coupled with Word2Vec), in terms of suggestions precision. WMD has a better coverage because it finds vectors for every article, hence every articles pair has a distance. However, the scalability of this approach will be really costly both in terms of computation resources and in terms of storage of (key-)words vectors. Therefore it would need either a huge bench of powerful servers or the cloud usage. The limit of our approach is that, being based on the categorization step, it only covers between 40% and 50% of the articles in entry. Hence, only pairs involving these categorized articles may be recommended. Nevertheless, the distance computation is really fast (up to 40,000 times faster than WMD – see Section 4.3.3.2) because it only manipulates identifiers (i.e., IDs), and the storage of these IDs (to be re-used at each distance computation) uses much less resources (37 times less). Both approaches have their own strengths and drawbacks, but the weaknesses of our approach are known and there are leads to reduce their impact to clearly outperform Word2Vec.

³⁵<https://babelnet.org/synset?word=bn:00020452n>

³⁶<https://babelnet.org/synset?word=bn:00065023n>

³⁷<https://babelnet.org/synset?word=bn:00000492n>

³⁸<https://babelnet.org/synset?word=bn:00031027n>

³⁹<https://babelnet.org/synset?word=bn:00061984n>

⁴⁰<https://babelnet.org/synset?word=bn:00006195n>

⁴¹<https://babelnet.org/synset?word=bn:00007985n>

Moreover, neural networks have been trained and provided promising results while involving many more parameters. The idea is that embracing every parameter which might play a role in the distance computation should make the similarity metric much more representative and accurate. Finally, we have the intuition that investigating on our planned future works both from the categorization part (see Section 3.4) and from the similarity part (see Section 4.4) may lead to a clear and significant outperforming compared to WMD. Currently, this idea remains at the stage of assumption and WMD still has a little advantage thanks to its capability to recommend articles for every article accessible, but does not scale properly.

4.4 Future work

This section describes the perspectives of future work. Those are either some solutions to encountered problems, or some interesting leads that might enhance the approach.

4.4.1 More Data Connected

As mentioned several times within this thesis, involving more synsets and neighbors might push further the performances of the information retrieval. The following two parts are the main ideas that we plan to work on in a near future.

Retrieve more neighbors.

Extracting *all* categories from validated and disambiguated synsets would increase the number of correct categories linked to articles and hence bring more related neighbors to connect articles together. As an example, in Figure 3.6 the categories *Health disasters*, *Pandemics*, *Sexually transmitted diseases and infections* and *Causes of death* from the validated keywords *AIDS* and *HIV* can be kept and would be as accurate as the common shared category *HIV/AIDS*. This idea should increase the number of neighbors retrieved and hence find more connections among categorized articles.

Retrieve more synsets.

This proposition shares the same idea as the one proposed in Section 3.4.2.4 because it proposes to use unconnected categories when there is a *safe*⁴² domain connection. Connecting synsets by their domains even when no category connection is found should help to significantly improve the recall, especially when domain coverage—the percentage of articles with exclusively correct domains extracted from synsets connections—is above 80% (see Table 3.5 from Section 3.3.3). Then, our approach might exploit categories from the validated synsets as it currently does and therefore bring many more potential true positives. The hardest part is now to define the threshold when domain connection is legitimate or not (e.g., synsets are connected only when not more than 1 synset by keyword or sub-keyword), and evaluate the quality of the recommended articles based on this approach.

Both of these ideas are not yet added to our approach and only the specific connected categories are kept because several general ones—*sometimes even off-topic*—may be included into synset’s categories. A perfect example is the synset for *Artificial intelligence*⁴³. It contains correct categories such as *Artificial intelligence* or *Computational neuroscience* but also very generic ones such as *Emerging technologies* or *Formal sciences*. Consequently, lot of potentially related neighbors are missed. Further investigations could be undertaken in order to identify ways to automatically filter out bad synsets and related ones from these unsafe categories.

⁴²This safety should be defined. Currently, we did not find a rule defining the threshold when categories are good enough based on domains connections.

⁴³<https://babelnet.org/synset?word=bn:00002150n>

Clean neighbor selection.

Identified noise (i.e., non scientific categories) are removed in the synset connection (Section 3.1.7), but not in the neighbors retrieval (Section 4.1.2) given that only neighbors matching validated categories are selected. However, some neighbors may belong to the correct category but also to undesired ones at the same time (e.g., an english movie speaking about artificial intelligence might potentially be embedded into list of neighbors for the category *Artificial intelligence*). Hence, excluding identified noisy categories might be a good option to automatic filtering out a quantity of bad neighbors.

4.4.2 Similarity

In order to improve both the quantity and the quality of predictions, the similarity equation may involve many more variables.

Consider original keywords.

Taking into account the original keywords of the articles before any pre-treatment and disambiguation may also increase the number of possible recommendations. Indeed, they could contribute to the linking of articles when exactly similar keywords are shared between two articles. This will help to improve recall when the categorization step fails, or when only some keywords are categorized (i.e. only a part of the keywords' synsets share common categories). More investigation should be realized in order to determine conditions when to use those and the effect on predictions inherited from these keywords.

Use the ontology tree.

The current implementation of the metric only exploits synsets inherited from categorized keywords and their related neighbors, in two different pools. Then, Jaccard indexes provide a ratio between intersection and union, for all different types of intersections. The limit of this approach is that the ratio does not take into account the number of keywords linked by the neighbors. Let us take the example where two articles have three keywords each, and only share one synset, from which 10 neighbors are associated. The four other synsets only have one isolated neighbor each. The Jaccard index of neighbors sets will be 10/14. Let us take the same example but this time, there is no common synset and all synsets only share one neighbors with a synset from the other article. The Jaccard index will be 3/21⁴⁴, whereas all synsets are connected. The pair of article from the second example might be much closer than the one from the first example.

4.4.3 Benchmarks

Ideally, comparing our approach against standard datasets/benchmarks will be really interesting. Unfortunately we did not find any dataset fitting our expectation. There are however a couple of interesting ones that we plan to further investigate.

⁴⁴There are 12 neighbors from each article (10+1+1), so 24 for the pair, from which 3 are in common. Therefore, there are 21 unique neighbors

RARD.

The datasets RARD [14] and RARD II [12] provide logs of recommended article pairs from Mr. DLib⁴⁵. This recommender system realizes an on-line evaluation by embedding different recommendation algorithm and rotating among those to suggest related articles to its users (i.e., A-B testing). The click-through rate (CTR) is provided in the logs. This is an indicator of the overall suggestions relevancy while it represents the ratio of the number of times an article is proposed—for a given pair—and how often it is clicked. This dataset might be used as a ground truth to optimize our metric weights to favor expected recommendations. Its current drawback is that only document IDs are provided, and it does not contain any textual information that our approach could use⁴⁶.

Web of Science dataset.

Another interesting dataset is the Web of Science dataset [62]. Keywords, titles, domains and areas—*equivalent to our categories*—are provided for the 46985 articles composing the dataset. Recommendations based on our approach might be computed for these articles and categories / domains could be used to validate their quality. In other words, recommendations would be considered as correct when both articles are included into the same area (i.e., categories), or within the same domain if categories are not exploitable (i.e., if categories are too narrowed or specific).

PubMed.

The last usable dataset might be the pubmed dataset⁴⁷. Given that PubMed recommendations are recognized to be accurate, the linked articles IDs returned by its API might be a good way to create a ground truth dataset. Then, this new dataset might be used as a ground truth in a similar way as proposed for RARD datasets, where these known valid suggestions would be used to optimize the similarity equation and its different weights.

4.4.4 Neural Network

The multi-layer perceptron presented in Section 4.3.2.4 provides promising results. Training it with more accurate data involving improvements discussed above should obviously improve its accuracy. However, the drawback of this approach is that its evaluation can currently not be realized using the same methodology as the one proposed for WMD (Section 4.3.3.2), given that the MLP is only a binary classifier predicting whether pairs are similar or not. We plan to modify the network in order to return a normalized distance score before the activation function. Then, the distance matrix could be created, and the same analysis might be run. This will give more insights about the efficiency of the network and its capability to predict good suggestions for expected pairs. Also, this will make it comparable to any other approach delivering distance for two documents.

⁴⁵<http://mr-dlib.org>

⁴⁶RARD creators plan to create RARD III in 2019 including textual metadata along predictions information.

⁴⁷<http://www.ncbi.nlm.nih.gov/pubmed/>

4.4.5 Probabilistic Models

Given that the data is already prepared for deep learning algorithms—to feed the perceptron and MLP networks—testing how other classification algorithms perform on our data will definitely be a serious option for future works. In top of our head, the Random Forest algorithms, Decision Trees, Naive Bayes are potential candidates that may be interesting to test. Moreover, their comparison may help to identify the most suitable algorithms for our data.

4.4.6 An Interactive Evaluation Protocol

As discussed in the analysis of the online evaluation (Section 4.2.2), the small scale, the evaluators' behaviors and their methodology to rate recommendations might somehow affect the results obtained. To counter this, a further evaluation protocol might be put in place, where they will be better guided through the evaluation process. For example, they might be asked to first estimate the quality of the recommendation when only its title is given (the rating will go from 0 to 100). Once rated, the interface will display more information such as the connectivity details (i.e., intersections) and they will be asked to rate again. At the end—after two or three steps—they will have the full content (title, authors, abstract, connection details and so on) and will be able to revise their rating, if needed. This would ensure that the users evaluate the recommended articles taking into account all the exposed data. This could make the evaluation process much more robust and understandable.

4.4.7 Industrial Perspectives

The approach described in this thesis aims to be embedded into Scilit project. Its findings open several doors for new interesting functionalities to be developed. The first direct benefit being the categorization of articles, this data enhancement may be used to create a mapping of categories per journal or publisher in order to have an overview of its publishing fingerprint. This mapping—as well as the distance information—might also be used to implement a journal / conference suggester. For this, our categorization step might be applied on keywords (generated from text or given by the author) and our approach may identify the best journal / conference matching those in order to identify the best candidate to publish with. For doing this, either the most similar articles could be used, or a percentage of category representativeness per journal might be computed.

Relying on the same idea, the articles distance (or category mapping) might be used to create clusters of related journals / papers, or even to create weighted graphs (or hypergraphs). Salvador et al. [35] created this weighted graph embracing BabelNet synsets and their allocated weights, by following recommendations from Navigli and Lapata [86]. This graph dedicated to our corpus might bring flexibility to more easily generate other statistics such as the influence of an author within a specific community or also to determine the size of a field / topic. With a notion of temporality (i.e., article publication date), the analysis of the topic tendency might help to detect new hot topic emerging from publications. Knowing this information could help researchers to start working on newest fashion topics earlier while topics are emerging.

4.5 Conclusion

This chapter describes the use of the disambiguated keywords (i.e., synsets) in output of the categorization step (Chapter 3). Data is augmented by selecting every neighbor of article synsets, sharing at least one category in common with the article categories. After these stages, augmented data is available for every article from which categories were identified. A new distance metric based on different weighted articles intersections types—*synsets intersections*, *synset-neighbor* and *neighbors intersections*—is created. Both offline (Section 4.2.1) and online (Section 4.2.2) evaluations show promising results, opening doors for further research in this way.

Several variants using different weights are tested and compared (Section 3.3) in order to show the behavior of our distance metric while modifying intersection weights. We also compared our approach with the Word Mover Distance (WMD) in an offline dataset. The WMD relies on Word2Vec (word embeddings) and compute the smallest euclidean distance over all 300-D keywords vectors. This comparison highlights that our approach may compete with the probabilistic methods, and is much more scalable (i.e., lighter to use in terms of resources).

Neural networks (perceptron and multi-layer perceptron) were fed with cardinality information about intersections and unions of all different types of possible intersections inherited from our approach. This also brings satisfactory results, and more research might be done in order to compare performances of other machine learning algorithms.

Chapter 5

Conclusion / Discussion

Retrieving relevant documents is a multi-domain problematic and a lot of research aroused in any application field. Often, digital libraries rely on users activities to recommend articles rather than being fully content-based. This thesis presents a novel approach to retrieve relevant articles based on semantic similarities. The first part of our approach disambiguates keywords by finding categories in common. After this step, data is augmented by selecting semantic neighborhood and article similarity is calculated from all relationships involving keywords and their respective neighbors.

Contributions

Categorization. The first contribution of this thesis is the categorization of scientific articles. To achieve this, keywords are exploited and the proposed approach identifies all possible word senses from BabelNet synsets and tries to find common categories shared by keywords from the same article. When no entry is found within the knowledge database, a further search pre-process the keywords and split those on spaces and punctuation. Then, an enhanced n-gram extraction approach (i.e., skip-gram) is used to generate all potential linear combinations. All of these potential senses are used to disambiguate the correct senses / contexts by identifying shared categories among keywords. The assumption is that the more keywords sharing a category, the more chance this category is representative of the article. In other words, the ontology tree is used for every article's keyword and edges are aligned from their respective categories. This categorization validates the context of the article as much as disambiguates keywords. An average precision of 0.91 both from offline and online evaluation is obtained.

Information Retrieval. The second step of the proposed approach retrieves connected articles. To realize that, disambiguated keywords—*more precisely their synsets*—are exploited and data augmented. The augmentation process extracts all neighbors from the ontology graph sharing at least one of the article categories. This leads to two sets: a set of keywords (i.e., synsets) and a set of neighbors. All possible intersections involving synsets and neighbors are used to compute the similarity score between two articles. This new metric is evaluated by its capability to predict similarity for similar journals in an offline dataset containing articles inherited from four different journals. Among these journals, two are highly similar and the two others are fully disconnected. The precision depends on the selected threshold but the best point from the ROC curve obtains a precision of 0.92. The online evaluation also provides positive insights given that it revealed the capability of our approach to predict relevant articles from all types of intersections. However, the evaluation methodology may be improved in order to make it more robust.

Overall, the evaluation of this step is promising given that it provides comparable results of the widely known word mover distance, computing its distance from Word2Vec vectors. The proposed approach competes with the word mover distance in terms of precision and is much faster to compute. The speed difference comes from the fact that our approach is linearly proportional to the number of articles whereas the computation time of the word mover distance is quadratic. However our approach brings less intersections. This drawback is inherited from the recall of the categorization steps and several leads were proposed in order to improve this part. Its big advantage is that intersections might easily be represented in order to give insights to a user about the way two articles are linked. For example, we could imagine a small 2D graph where the article and its related articles will be represented by different bubbles, connected with weighted links (larger links would represent higher similarity). Then, a click on the links could show the connection details (i.e., which synsets / neighbors in common). This is something that is impossible in collaborative filtering recommendations or other probabilistic approaches.

Perspectives

Scientific This thesis provides an interesting data augmentation approach unlocking explainable semantical linking of scientific articles. Some improvements and further investigations are planned to be realized in order to push it further. Among those, the exploitation of domains to validate synsets seems to be a good lead to improve the recall of the categorization part¹. Another potential way to enhance the proposed approach would be the exploitation of additional data. For example, integrating a naive full-text exact matching from original-ambiguous-keywords into the similarity computation (weighted with a small coefficient) might help to bring connections when no other semantic relationships are found. Also, the similarity equation could take into account the number of keywords connected from articles relationships instead of only considering the ratio between the numbers of intersections and the number of potential connections.

Another really interesting lead would be to build a hybrid approach combining the proposed approach together with a probabilistic one. This should help to improve the low recall of our approach and therefore allow us to recommend related papers even for uncategorized articles. As a suggestion, we could use the vector space model associated with a dimension reduction technique. This association should theoretically scale well and represent a good alternative to our approach when suggestions are marginal (i.e., low similarity score), incomplete (i.e., only a few categorized articles match) or inexistent (i.e., categorization failed, or no matching retrieved).

Industrial This thesis also has an industrial dimension given that it aims to bring added value to MDPI (i.e., thesis funder, open access publisher), Scilit (platform of scientific articles developed within MDPI) and other internal projects. A possible lead might be the automation of the best matching reviewers retrieval from the submission system, similarly to what was done in [23]. Indeed, finding the closest papers of a new submitted article might provide a list of authors able to potentially review the content. This would save a significant amount of time to journal editors.

¹Given that domains are much more general than categories, connecting keywords synsets by their domains in common should bring more connections.

Another idea that could be developed is a cross-publisher journal suggester. For that purpose, the articles categories might be exploited in order to identify the main journal categories. Also, similarity scores could be used to cluster articles and their respective journals (i.e., projected in two or more dimensions). After these steps, the categorization of a new article might help to find the best journal to publish with. Indeed, the results could propose either journals having the closest scope—*inherited from main categories overlapping*—or the ones having the closest centroid—*from the dimension projection*.

Finally, realizing the combination of scientific and industrial perspectives could help to strengthen several parts of both the publishing process and other internal projects. We aim to push further both sides in order to bring more interesting functionalities and services to the scientific community.

Chapter 6

Résumé

6.1 Introduction et Motivation

Les bases de données numériques devenant de plus en plus volumineuses, les internautes peuvent facilement être noyés dans la masse de données. La même observation peut être faite au niveau des publications scientifiques. Aujourd'hui, les phases de recherche bibliographique et d'acquisition de connaissances sont des tâches complexes et fastidieuses auxquelles les scientifiques sont régulièrement confrontés. Par conséquent, les chercheurs consacrent un temps précieux à la recherche documentaire et sont parfois contraints de s'appuyer sur des services externes pour effectuer une veille scientifique, bien souvent superficielle. Pour remédier à ces problèmes, ils optent bien souvent pour les bases de données les plus complètes (alors que seules des fonctionnalités de recherche basiques sont proposées), explorent quelques revues réputées dans leurs domaines de compétence, ou recherchent des articles pertinents sur des plateformes sélectives—*et par conséquent incomplètes*. L'ensemble de ces solutions sont coûteuses en temps et le besoin d'une plate-forme scientifique centralisée apportant des résultats de recherche complets et guidant les utilisateurs vers la découverte de la littérature se fait de plus en plus sentir. La première étape pour atteindre cette ambition consisterait à explorer la littérature scientifique, y extraire les principaux concepts des articles et identifier les similitudes sémantiques entre eux. C'est l'objectif de ce manuscrit.

Les deux contributions principales s'articulent autour de :

- La catégorisation, le pré-traitement et la désambiguïsation des mots afin de sélectionner les mots clés les plus représentatif d'un article
- La recherche d'informations, l'augmentation des données et la détection de similarités entre deux articles afin de matérialiser une relation à un niveau sémantique

6.2 Contributions

Cette thèse a pour objectif d'exploiter des articles scientifiques et de trouver des similitudes sémantiques entre eux. La Figure 6.1 illustre notre approche. La première étape, la catégorisation (Section 6.2.1), exploite les mots-clés des articles pour les désambiguïser et identifier les catégories principales. La deuxième étape, la récupération des informations (Section 6.2.2), augmente les données en agrégeant les voisins des mots clés désambiguïsés. Ensuite, les similitudes entre articles sont estimées par les intersections de ces ensembles de mots clés désambiguïsés et augmentés.

Pour effectuer ces deux contributions, la base de données lexicographique et encyclopédique multilingue BabelNet [88] est utilisée. Son architecture est le fruit de la

superposition de plusieurs lexiques sémantiques (WordNet, VerbNet) et de bases de données collaboratives (Wikipedia et autres données Wiki). Ainsi, BabelNet est une base de connaissances très complète, utilisable à la fois pour la désambiguïsation et l'augmentation de données. Elle peut être vue comme une ontologie (c'est-à-dire un graphe de connaissances) ou encore un dictionnaire fournissant tous les homonymes d'un mot donné. Ces homonymes sont appelés *synsets* dans BabelNet et représentent des mots contextualisés avec un sens spécifique. Les données supplémentaires telles que les sens (c'est-à-dire les définitions de dictionnaire), les catégories et les domaines¹ sont attachés aux synsets, ainsi que tous les mots en relation que l'on nommera plus généralement *voisins*.

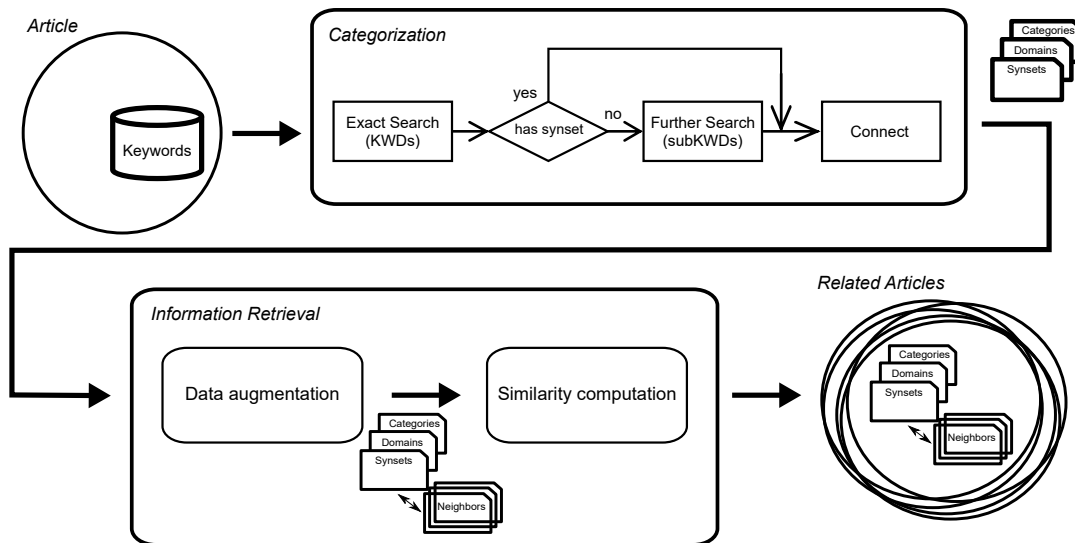


FIGURE 6.1: L'aperçu général de notre approche.

6.2.1 Catégorisation

Notre approche de catégorisation des articles scientifiques exploite les mots-clés qui proviennent soit des auteurs, soit d'un extracteur de sujets. La partie supérieure de la Figure 6.1 illustre le travail de catégorisation qui désambiguïse les mots-clés et les renvoie avec leurs catégories ainsi que d'autres données contextualisées.

6.2.1.0.1 Extraction des synsets de BabelNet. La recherche exacte est la première tentative pour trouver les données dans l'ontologie de BabelNet, et traite les mots-clés sans aucun prétraitement préliminaire. Lorsqu'aucune donnée n'est extraite pour ces mots-clés une seconde solution est tentée. Dans un premier temps, les mots vides et la ponctuation sont supprimés des mots-clés d'origine et une approche par réduction progressive du n-gramme tente plusieurs combinaisons linéaires (appelées sous-mots clés) afin d'étoffer le nombre d'entrées potentielles. Notre décomposition en n-grammes est proche de la logique du "skip-gram" car elle ne modifie pas l'ordre des mots et permet de sauter un ou plusieurs mots lors de la génération des n-grammes. La première étape de cette génération utilise le nouveau mot clé pré-traité—composé de X mots—et recherche leurs entrées dans BabelNet. Si aucun synset n'est extrait, la taille de la fenêtre est réduite et tous les X-1-grams sont testés.

¹Les domaines sont beaucoup plus généraux que les catégories (principalement héritées de Wikipedia) dans BabelNet.

La génération de n-grammes se termine à la fin d'une étape dès lors que des synsets ont été trouvés.

Un exemple théorique comportant un mot clé à quatre mots ("ABCD") générera, lors de la troisième étape les bi-grammes classiques AB , BC et CD , ainsi que ceux incluant les sauts AC , AD et BD .

6.2.1.0.2 Désambiguïisation du sens des mots-clés. Les synsets potentiels—provenant soit de la recherche exacte, soit de la recherche avancée (décomposition en n-grammes)—sont désambiguïsés par la connexion de leurs catégories communes. En effet, les mots-clés ont une liste de tous les synsets potentiels (c'est-à-dire toutes les entrées de dictionnaire correspondantes), et notre approche les désambiguïse en supposant que plus une catégorie est partagée par des mots-clés, plus elle a de chance d'être représentative de l'article.

Pour ce faire, trois vecteurs colonnes binaires représentant le recouvrement des catégories par mots-clés, par synsets et par sous-mots-clés sont créés—une catégorie est alors activée seulement si elle est partagée par deux éléments (mots-clés, synsets ou sous-mots-clés) ou plus. Ces vecteurs sont utilisés pour identifier les catégories qui se chevauchent sans sélectionner une catégorie qui sera trouvée dans plusieurs synsets du même mot clé (ou sous-mot clé). Cela évite également de sélectionner une catégorie lorsque le même synset est présent dans deux mots-clés différents. Ensuite, le produit Hadamard de ces trois vecteurs donne un vecteur binaire unique synthétisant les catégories représentatives². Enfin, les mots-clés sont désambiguïsés en sélectionnant uniquement les synsets appartenant à ces catégories partagées.

6.2.2 Extraction de l'information

Afin d'identifier les relations sémantiques entre les articles, l'augmentation de données des synsets désambiguïsés est réalisée. Pour cela, tous les voisins—c'est-à-dire les synsets connectés dans le graphe d'ontologie BabelNet—partageant l'une des catégories de l'article sont extraits. Ensuite, une métrique de similarité est calculée (Equation 6.1), elle prend en compte toutes sortes de relations entre les synsets des articles et leurs voisins respectifs.

$$\begin{aligned} sim(A_i, A_j) = \frac{1}{\alpha + \beta + \gamma} * \left(\right. & \alpha jac(K_i, K_j) \\ & + \frac{\beta}{2} jacKN(K_i, N_j, K_j) + \frac{\beta}{2} jacKN(K_j, N_i, K_i) \\ & \left. + \gamma jacNN(N_i, N_j, K_i, K_j) \right) \end{aligned} \quad (6.1)$$

avec:

- K_x l'ensemble de synsets venant des mots-clés de l'article A_x
- N_x l'ensemble de voisins venant des mots-clés de l'article A_x
- $jac()$, $jacKN()$ and $jacNN()$ sont trois variantes d'index jaccard définies les paragraphes suivants.

Cette métrique de similarité a une plage comprise entre 0 (dissimilaire) et 1 (correspondance parfaite). $jac(K_i, K_j)$ est le coefficient Jaccard des mots-clés de l'article

²Le vecteur de catégorie par sous-mots-clés englobe également les données provenant de la recherche exacte (c'est-à-dire des mots-clés) afin que ses données ne soient pas neutralisées par le produit Hadamard.

A_i et de l'article A_j . Il représente le chevauchement entre les deux ensembles de mots-clés. Même si, théoriquement, ces connexions devraient également être extraites par une simple correspondance de mots clés (c'est-à-dire une comparaison exclusivement textuelle), celles-ci devraient être plus sûres étant donné que les mots clés sont désambiguïsés. En d'autres termes, les articles dont les termes sont utilisés de manière interchangeable dans différents contextes ne sont pas liés par notre approche, contrairement aux approches basées sur la correspondance exacte des mots clés.

$$jac(K_i, K_j) = \frac{|K_i \cap K_j|}{|K_i \cup K_j|}$$

Les deuxième et troisième relations sont des relations englobant les mots-clés et les voisins des deux articles. Ce sont les premières relations sémantiques étant donné qu'un voisin d'un mot clé d'un article peut être connecté à un mot clé d'un autre article. Par conséquent, même si les articles ne partagent aucun mot clé en commun, une similarité peut être identifiée.

$$jacKN(K_i, N_j, K_j) = \frac{|K_i \cap N_j|}{|K_i \cup N_j| - |K_i \cap K_j|}$$

La dernière relation incluse dans notre métrique est une connexion sémantique lointaine, qui identifie les connexions du voisinage des synsets. En d'autres termes, les mots clés de deux articles ne sont liés que par les voisins de leurs synsets respectifs.

$$jacNN(N_i, N_j, K_i, K_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j| - (|K_i \cap N_j| + |N_i \cap K_j|)}$$

Cette nouvelle métrique peut, en fonction des poids accordés aux différents coefficients, favoriser les connexions les plus sûres (mot-clé–mot-clé), les sémantiques (mot-clé–voisin) ou même les connexions sémantiquement éloignées (voisin–voisin). La Figure 6.2 fournit un exemple concret de notre approche appliquée à deux articles (à gauche et à droite). Le premier article contient deux mots-clés (à gauche) et le deuxième trois.

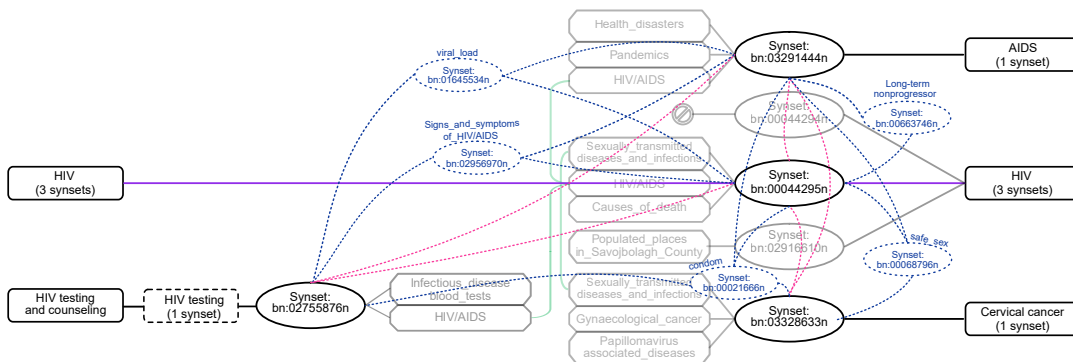


FIGURE 6.2: Exemple d'intersections héritées de notre approche.

L'étape de catégorisation des deux articles est affichée en transparence. Les deux mots-clés du premier article (à gauche) sont liés par la catégorie *HIV/AIDS*, tandis que le deuxième article contient un synset (hérité de *HIV*) reliant ses deux autres mots-clés *AIDS* et *Cervical cancer* par les catégories (*HIV/AIDS* et *Sexually transmitted*

diseases and infections). Les voisins sont extraits de ces synsets sélectionnés (seuls une partie d'entre eux sont représentés sur la Figure 6.2).

Enfin, les deux articles partagent le mot clé *HIV* (ligne violette) et deux mots clés—*AIDS et VIH*—du second article ont le synset *HIV testing* (hérité du sous-mot-clé *HIV testing and consulting*) comme voisin (lignes pointillées roses). Les intersections entre voisins (lignes pointillées bleues) montrent que tous ces synsets sont interconnectés.

6.3 Conclusion

La première contribution de cette thèse concerne la catégorisation des articles scientifiques en identifiant les catégories de mots-clés qui se chevauchent. Ces connexions de catégories valident également les mots-clés corrects (c'est-à-dire les synsets). L'évaluation de la pertinence de notre catégorisation fournit des résultats satisfaisants, car la précision est de 0,91 sur un jeu de données hors ligne ainsi que lors d'une évaluation en ligne comprenant 24 évaluateurs et 110 articles.

La deuxième contribution de cette thèse est l'évaluation de la similarité des articles à partir de synsets désambiguïsés et de leurs données augmentées. Une comparaison heuristique des différentes variantes de notre approche est donnée. De plus, l'approche proposée est également comparée avec la word mover distance [63] exploitant les projections multi-dimensionnels des mots Word2Vec [81]. Cette comparaison donne des résultats satisfaisants étant donné que notre approche fournit des résultats aussi précis que ceux obtenus par la distance word mover distance (respectivement, une précision de 0,92 VS. 0,91).

De plus, la cardinalité de toutes les intersections et unions des paires d'articles décrites dans ce résumé, ainsi que d'autres caractéristiques, ont été utilisées pour entraîner un réseau de neurones, à savoir un perceptron multicouche. Le réseau a appris à détecter la similarité des articles en fonction du nombre d'intersections et d'unions entre les combinaisons de mots clés et de voisins (voir Section 6.2.2), de mots clés textuels communs, d'un indice de confiance de catégories et d'autres données héritées de notre approche. Le modèle résultant qui a été entraîné révèle également des résultats prometteurs car il obtient une précision de 88% tout en apportant plus de paires d'articles.

Bibliography

- [1] Kjersti Aas and Line Eikvil. "Text Categorisation: A Survey." In: *Norwegian Computing Center* (1999).
- [2] Charu C. Aggarwal. *Data Classification: Algorithms and Applications*. 1st. Chapman & Hall/CRC, 2014. ISBN: 978-1-4665-8674-1.
- [3] Charu C. Aggarwal and ChengXiang Zhai. "A Survey of Text Classification Algorithms". In: *Mining Text Data*. Ed. by Charu C. Aggarwal and ChengXiang Zhai. Boston, MA: Springer US, 2012, pp. 163–222. ISBN: 978-1-4614-3222-7 978-1-4614-3223-4. DOI: [10.1007/978-1-4614-3223-4_6](https://doi.org/10.1007/978-1-4614-3223-4_6).
- [4] Hussein Al-Natsheh. "Text Mining Approaches for Semantic Similarity Exploration and Metadata Enrichment of Scientific Digital Libraries". PhD thesis. 2019.
- [5] Mehdi Allahyari et al. "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques". In: *arXiv:1707.02919 [cs]* (July 2017). arXiv: 1707.02919.
- [6] Xavier Amatriain et al. "Data Mining Methods for Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci et al. Boston, MA: Springer US, 2011, pp. 39–71. ISBN: 978-0-387-85819-7 978-0-387-85820-3. DOI: [10.1007/978-0-387-85820-3_2](https://doi.org/10.1007/978-0-387-85820-3_2).
- [7] Daniel Andor et al. "Globally normalized transition-based neural networks". In: *ACL* (2016).
- [8] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. "Modern information retrieval: the concepts and technology behind search". In: *Choice Reviews Online* 48.12 (Aug. 2011), pp. 48–6950–48–6950. ISSN: 0009-4978, 1523-8253. DOI: [10.5860/CHOICE.48-6950](https://doi.org/10.5860/CHOICE.48-6950).
- [9] Xiaomei Bai et al. "Scientific Paper Recommendation: A Survey". In: *IEEE Access* 7 (2019), pp. 9324–9339. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2018.2890388](https://doi.org/10.1109/ACCESS.2018.2890388).
- [10] S Balakrishnama and A Ganapathiraju. *Linear Discriminant Analysis - A Brief Tutorial*. 1998.
- [11] Joeran Beel. "Towards Effective Research-Paper Recommender Systems and User Modeling based on Mind Maps". PhD thesis. 2015.
- [12] Joeran Beel, Barry Smyth, and Andrew Collins. "RARD II: The 2nd Related-Article Recommendation Dataset". In: *arXiv:1807.06918 [cs]* (2018). arXiv: 1807.06918.
- [13] Joeran Beel et al. "Research-paper recommender systems: a literature survey". In: *International Journal on Digital Libraries* 17.4 (2016), pp. 305–338. ISSN: 1432-5012, 1432-1300. DOI: [10.1007/s00799-015-0156-0](https://doi.org/10.1007/s00799-015-0156-0).
- [14] Joeran Beel et al. "RARD: The Related-Article Recommendation Dataset". In: *D-Lib Magazine* 23.7/8 (2017). ISSN: 1082-9873. DOI: [10.1045/july2017-beel](https://doi.org/10.1045/july2017-beel).

- [15] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 281–305. ISSN: ISSN 1533-7928.
- [16] Pavel Berkhin. "Survey of Clustering Data Mining Techniques". In: (2006), p. 56.
- [17] Michael W. Berry and Malu Castellanos. *Survey of text mining II*. Springer, 2008.
- [18] Chris Biemann. "Ontology learning from text: A survey of methods." In: *LDV forum*. Vol. 20. 2005, pp. 75–93.
- [19] Christopher Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN: 978-0-387-31073-2.
- [20] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.
- [21] J. Bobadilla et al. "Recommender systems survey". In: *Knowledge-Based Systems* 46 (July 2013), pp. 109–132. ISSN: 09507051. DOI: [10.1016/j.knsys.2013.03.012](https://doi.org/10.1016/j.knsys.2013.03.012).
- [22] Devendra Singh Chaplot and Ruslan Salakhutdinov. "Knowledge-Based Word Sense Disambiguation Using Topic Models". In: *AAAI18*. ACM, 2018, p. 8.
- [23] Laurent Charlin and Richard Zemel. "The Toronto Paper Matching System: An automated paper-reviewer assignment system". In: *ICML*. 2013.
- [24] Arman Cohan and Nazli Goharian. "Scientific document summarization via citation contextualization and scientific discourse". In: *International Journal on Digital Libraries* 19.2-3 (Sept. 2018), pp. 287–303. ISSN: 1432-5012, 1432-1300. DOI: [10.1007/s00799-017-0216-8](https://doi.org/10.1007/s00799-017-0216-8).
- [25] Ellen E. Connor et al. "Gold Nanoparticles Are Taken Up by Human Cells but Do Not Cause Acute Cytotoxicity". In: *Small* 1.3 (Mar. 2005), pp. 325–327. ISSN: 1613-6810, 1613-6829. DOI: [10.1002/sm11.200400093](https://doi.org/10.1002/sm11.200400093).
- [26] Juncal Cunado, Luis A. Gil-Alana, and Rangan Gupta. "Persistence in trends and cycles of gold and silver prices: Evidence from historical data". In: *Physica A: Statistical Mechanics and its Applications* 514 (Jan. 2019), pp. 345–354. ISSN: 03784371. DOI: [10.1016/j.physa.2018.09.081](https://doi.org/10.1016/j.physa.2018.09.081).
- [27] David Dann, Matthias Hauser, and Jannis Hanke. "Reconstructing the Giant: Automating the Categorization of Scientific Articles with Deep Learning Techniques". In: *WI 2017*. 2017, p. 12.
- [28] Scott C. Deerwester et al. "Indexing by Latent Semantic Analysis". In: *Journal of the Association for Information Science and Technology* 41 (1990), pp. 391–407. DOI: [10.1002/\(SICI\)1097-4571\(199009\)41:6%3C391::AID-ASI1%3E3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9).
- [29] Antonio Di Marco and Roberto Navigli. "Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction". In: *Computational Linguistics* 39.3 (Sept. 2013), pp. 709–754. ISSN: 0891-2017, 1530-9312. DOI: [10.1162/COLI_a_00148](https://doi.org/10.1162/COLI_a_00148).
- [30] Robert H. Doremus. "Optical Properties of Small Clusters of Silver and Gold Atoms". In: *Langmuir* 18.6 (Mar. 2002), pp. 2436–2437. ISSN: 0743-7463, 1520-5827. DOI: [10.1021/1a011350h](https://doi.org/10.1021/1a011350h).

- [31] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. *Collaborative Filtering Recommender Systems*. IEEE, 2011. ISBN: 978-1-60198-442-5.
- [32] Felice Ferrara, Nirmala Pudota, and Carlo Tasso. "A Keyphrase-Based Paper Recommender System". In: *IRCDL*. Vol. 249. Springer, 2011, pp. 14–25. ISBN: 978-3-642-27301-8 978-3-642-27302-5. DOI: [10.1007/978-3-642-27302-5_2](https://doi.org/10.1007/978-3-642-27302-5_2).
- [33] G. P. Ferreira and F. B. Gil. "Elemental Analysis of Gold Coins by Particle Induced X-Ray Emission (pixe)". In: *Archaeometry* 23.2 (1981), pp. 189–197. ISSN: 1475-4754. DOI: [10.1111/j.1475-4754.1981.tb00305.x](https://doi.org/10.1111/j.1475-4754.1981.tb00305.x).
- [34] J. R. Firth. "A synopsis of linguistic theory, 1930-1955". In: *Studies in Linguistic Analysis*. Studies in Linguistic Analysis (1957).
- [35] Marc Franco-Salvador et al. "Cross-domain polarity classification using a knowledge-enhanced meta-classifier". In: *Knowledge-Based Systems* 86 (Sept. 2015), pp. 46–56. ISSN: 09507051. DOI: [10.1016/j.knosys.2015.05.020](https://doi.org/10.1016/j.knosys.2015.05.020).
- [36] Evgeniy Gabrilovich and Shaul Markovitch. "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge". In: *AAAI 06*. Vol. 2. ACM, 2006, pp. 1301–1306. ISBN: 978-1-57735-281-5.
- [37] Kata Gábor et al. "Semantic annotation of the acl anthology corpus for the automatic analysis of scientific literature". In: *LREC*. 2016, pp. 3694–3701.
- [38] Isidoro Gil-Leiva and Adolfo Alonso-Arroyo. "Keywords given by authors of scientific articles in database descriptors". In: *Journal of the American Society for Information Science and Technology* 58.8 (June 2007), pp. 1175–1187. ISSN: 15322882, 15322890. DOI: [10.1002/asi.20595](https://doi.org/10.1002/asi.20595).
- [39] G. Golub and W. Kahan. "Calculating the Singular Values and Pseudo-Inverse of a Matrix". In: *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis* 2.2 (Jan. 1965), pp. 205–224. ISSN: 0887-459X. DOI: [10.1137/0702016](https://doi.org/10.1137/0702016).
- [40] Gene Howard Golub. "Least squares, singular values and matrix approximations". In: *Aplikace matematiky* 13.1 (1968), pp. 44–51. ISSN: 0373-6725.
- [41] David Guthrie et al. "A Closer Look at Skip-gram Modelling". In: *LREC 06*. ACL, May 2006.
- [42] Brian Hammond, Amit Sheth, and Krzysztof Kochut. "Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content". In: *RWSWA*. IOS, 2002, p. 22.
- [43] Tony Hey and Anne Trefethen. "The Data Deluge: An e-Science Perspective". In: *Wiley Series in Communications Networking & Distributed Systems*. Ed. by Fran Berman, Geoffrey Fox, and Tony Hey. Chichester, UK: John Wiley & Sons, Ltd, Mar. 2003, pp. 809–824. ISBN: 978-0-470-85319-1 978-0-470-86716-7. DOI: [10.1002/0470867167.ch36](https://doi.org/10.1002/0470867167.ch36).
- [44] Wael H.Gomaa and Aly A. Fahmy. "A Survey of Text Similarity Approaches". In: *International Journal of Computer Applications* 68.13 (Apr. 2013), pp. 13–18. ISSN: 09758887. DOI: [10.5120/11638-7118](https://doi.org/10.5120/11638-7118).
- [45] Thomas Hofmann. "Probabilistic Latent Semantic Analysis". In: *UAI 99*. Morgan Kaufmann Publishers, 1999, pp. 289–296. ISBN: 978-1-55860-614-2.
- [46] Matthew Honnibal and Mark Johnson. "An Improved Non-monotonic Transition System for Dependency Parsing". In: *EMNLP*. ACL, Sept. 2015, pp. 1373–1378. DOI: [10.18653/v1/D15-1162](https://doi.org/10.18653/v1/D15-1162).

- [47] Andreas Hotho, Andreas Nürnberger, and Gerhard Paa's s. "A brief survey of text mining." In: *Ldv Forum*. Vol. 20. 2005, pp. 19–62.
- [48] Anna Huang. "Similarity measures for text document clustering". In: *Proceedings of the sixth new zealand computer science research student conference (NZC-SRSC2008)*, Christchurch, New Zealand. 2008, pp. 49–56.
- [49] Po-Sen Huang et al. "Learning deep structured semantic models for web search using clickthrough data". In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. San Francisco, California, USA: ACM Press, 2013, pp. 2333–2338. ISBN: 978-1-4503-2263-8. DOI: [10.1145/2505515.2505665](https://doi.org/10.1145/2505515.2505665).
- [50] Yinghao Huang, Xipeng Wang, and Yi Lu Murphey. "Text categorization using topic model and ontology networks". In: *DMIN*. IEEE, 2014, p. 7.
- [51] Germán Hurtado Martín et al. "Using semi-structured data for assessing research paper similarity". In: *Information Sciences* 221 (Feb. 2013), pp. 245–261. ISSN: 00200255. DOI: [10.1016/j.ins.2012.09.044](https://doi.org/10.1016/j.ins.2012.09.044).
- [52] Anil K. Jain. "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters* 31.8 (June 2010), pp. 651–666. ISSN: 01678655. DOI: [10.1016/j.patrec.2009.09.011](https://doi.org/10.1016/j.patrec.2009.09.011).
- [53] Anil K. Jain. *Data Clustering: A Review*. ACM, 1999.
- [54] Maciej Janik and Krys J. Kochut. "Wikipedia in Action: Ontological Knowledge in Text Categorization". In: *2008 IEEE International Conference on Semantic Computing*. Santa Monica, CA, USA: IEEE, Aug. 2008, pp. 268–275. DOI: [10.1109/ICSC.2008.53](https://doi.org/10.1109/ICSC.2008.53).
- [55] Yichen Jiang et al. "Recommending academic papers via users' reading purposes". In: *RecSys*. ACM, 2012, pp. 241–244.
- [56] Rob Johnson, Anthony Watkinson, and Michael Mabe. *The STM Report: An overview of scientific and scholarly publishing*. Tech. rep. 2018.
- [57] D Kim and B Yum. "Collaborative filtering based on iterative principal component analysis". In: *Expert Systems with Applications* 28.4 (May 2005), pp. 823–830. ISSN: 09574174. DOI: [10.1016/j.eswa.2004.12.037](https://doi.org/10.1016/j.eswa.2004.12.037).
- [58] Sang-Bum Kim, Hee-Cheol Seo, and Hae-Chang Rim. "Information Retrieval Using Word Senses: Root Sense Tagging Approach". In: *SIGIR*. ACM, 2004, pp. 258–265. ISBN: 978-1-58113-881-8. DOI: [10.1145/1008992.1009038](https://doi.org/10.1145/1008992.1009038).
- [59] Su Nam Kim et al. "Automatic keyphrase extraction from scientific articles". In: *Language resources and evaluation* 47.3 (2013), pp. 723–742.
- [60] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980 [cs]* (Dec. 2014). arXiv: 1412.6980.
- [61] Yehuda Koren and Robert Bell. "Advances in Collaborative Filtering". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA: Springer US, 2015, pp. 77–118. ISBN: 978-1-4899-7637-6. DOI: [10.1007/978-1-4899-7637-6_3](https://doi.org/10.1007/978-1-4899-7637-6_3).
- [62] Kamran Kowsari et al. *Web of Science Dataset*. Mar. 2018.
- [63] Matt J Kusner et al. "From Word Embeddings To Document Distances". In: *ICML*. ACM, 2015, p. 10.

- [64] Linyuan Lü et al. "Recommender Systems". In: *Physics Reports* 519.1 (Oct. 2012). arXiv: 1202.1112, pp. 1–49. ISSN: 03701573. DOI: [10.1016/j.physrep.2012.02.006](https://doi.org/10.1016/j.physrep.2012.02.006).
- [65] Thomas K Landauer, Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis". In: *Discourse Processes* 25.2-3 (Jan. 1998), pp. 259–284. ISSN: 0163-853X, 1532-6950. DOI: [10.1080/01638539809545028](https://doi.org/10.1080/01638539809545028).
- [66] Bastien Latard et al. "Catégorisation d'articles scientifiques basée sur les relations sémantiques des mots-clés". In: *EGC*. 2018, pp. 371–372.
- [67] Bastien Latard et al. "Towards a Semantic Search Engine for Scientific Articles". In: *TPDL*. Springer, Sept. 2017, pp. 608–611. DOI: [10.1007/978-3-319-67008-9_54](https://doi.org/10.1007/978-3-319-67008-9_54).
- [68] Bastien Latard et al. "Using Semantic Relations between Keywords to Categorize Articles from Scientific Literature". In: *ICTAI*. IEEE, Nov. 2017, pp. 260–264. DOI: [10.1109/ICTAI.2017.00049](https://doi.org/10.1109/ICTAI.2017.00049).
- [69] Michael Lesk. "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone". In: *Proceedings of the 5th Annual International Conference on Systems Documentation*. SIGDOC '86. event-place: Toronto, Ontario, Canada. New York, NY, USA: ACM, 1986, pp. 24–26. ISBN: 978-0-89791-224-2. DOI: [10.1145/318723.318728](https://doi.org/10.1145/318723.318728).
- [70] Keqian Li et al. "Unsupervised Neural Categorization for Scientific Publications". In: *Proceedings of the 2018 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2018, pp. 37–45.
- [71] Jimmy Lin and W. John Wilbur. "PubMed related articles: a probabilistic topic-based model for content similarity". In: *BMC Bioinformatics* 8.1 (2007), p. 423. ISSN: 14712105. DOI: [10.1186/1471-2105-8-423](https://doi.org/10.1186/1471-2105-8-423).
- [72] Shuang Liu, Clement Yu, and Weiyi Meng. "Word sense disambiguation in queries". In: *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*. Bremen, Germany: ACM Press, 2005, p. 525. ISBN: 978-1-59593-140-5. DOI: [10.1145/1099554.1099696](https://doi.org/10.1145/1099554.1099696).
- [73] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. "Content-based Recommender Systems: State of the Art and Trends". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci et al. Boston, MA: Springer US, 2011, pp. 73–105. ISBN: 978-0-387-85819-7 978-0-387-85820-3. DOI: [10.1007/978-0-387-85820-3_3](https://doi.org/10.1007/978-0-387-85820-3_3).
- [74] J Macqueen. "Some Methods for Classification and Analysis of Multivariate Observations". In: *Multivariate Observations*. 1967, p. 17.
- [75] Christopher D. Manning et al. "The Stanford CoreNLP Natural Language Processing Toolkit." In: *System Demonstrations*. ACL, 2014, pp. 55–60.
- [76] Nikolaos F. Matsatsinis, Kleanthi Lakiotaki, and Pavlos Delias. "A system based on multiple criteria analysis for scientific paper recommendation". In: *PCI*. 2007, pp. 135–149.
- [77] S. Menaka and N. Radha. "Text classification using keyword extraction technique". In: *International Journal of Advanced Research in Computer Science and Software Engineering* 3.12 (2013).
- [78] Robin van Meteren and Maarten van Someren. "Using Content-Based Filtering for Recommendation". In: *ECML*. 2000, p. 10.

- [79] Ingo Mierswa et al. "YALE: Rapid Prototyping for Complex Data Mining Tasks". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. event-place: Philadelphia, PA, USA. New York, NY, USA: ACM, 2006, pp. 935–940. ISBN: 978-1-59593-339-3. DOI: [10.1145/1150402.1150531](https://doi.org/10.1145/1150402.1150531).
- [80] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. "PageRank on semantic networks, with application to word sense disambiguation". In: *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*. Geneva, Switzerland: Association for Computational Linguistics, 2004, 1126–es. DOI: [10.3115/1220355.1220517](https://doi.org/10.3115/1220355.1220517).
- [81] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *arXiv:1301.3781 [cs]* (Jan. 2013). arXiv: 1301.3781.
- [82] George A. Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.
- [83] Amine Naak, Hicham Hage, and Esma Aïmeur. "A Multi-criteria Collaborative Filtering Approach for Research Paper Recommendation in Papyrus". In: *E-Technologies: Innovation in an Open World*. Ed. by Gilbert Babin, Peter Kropf, and Michael Weiss. Vol. 26. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 25–39. ISBN: 978-3-642-01186-3 978-3-642-01187-0. DOI: [10.1007/978-3-642-01187-0_3](https://doi.org/10.1007/978-3-642-01187-0_3).
- [84] Le Nguyen Hoai Nam and Ho Bao Quoc. "A Combined Approach for Filter Feature Selection in Document Classification". In: *ICTAI*. IEEE, Nov. 2015, pp. 317–324. ISBN: 978-1-5090-0163-7. DOI: [10.1109/ICTAI.2015.56](https://doi.org/10.1109/ICTAI.2015.56).
- [85] Cristiano Nascimento et al. "A source independent framework for research paper recommendation". In: *JCDL*. Ottawa, Ontario, Canada: ACM Press, 2011, p. 297. ISBN: 978-1-4503-0744-4. DOI: [10.1145/1998076.1998132](https://doi.org/10.1145/1998076.1998132).
- [86] R. Navigli and M. Lapata. "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.4 (Apr. 2010), pp. 678–692. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2009.36](https://doi.org/10.1109/TPAMI.2009.36).
- [87] Roberto Navigli. "Word sense disambiguation: A survey". In: *ACM Computing Surveys* 41.2 (Feb. 2009), pp. 1–69. ISSN: 03600300. DOI: [10.1145/1459352.1459355](https://doi.org/10.1145/1459352.1459355).
- [88] Roberto Navigli and Simone Paolo Ponzetto. "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network". In: *Artificial Intelligence* 193 (Dec. 2012), pp. 217–250. ISSN: 00043702. DOI: [10.1016/j.artint.2012.07.001](https://doi.org/10.1016/j.artint.2012.07.001).
- [89] Kezban Dilek Onal et al. "Neural information retrieval: at the end of the early years". In: *Information Retrieval Journal* 21.2-3 (June 2018), pp. 111–182. ISSN: 1386-4564, 1573-7659. DOI: [10.1007/s10791-017-9321-y](https://doi.org/10.1007/s10791-017-9321-y).
- [90] Aasish Pappu. "Using Wikipedia for Hierarchical Finer Categorization of Named Entities". In: *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*. Hong Kong, China, 2009, pp. 779–786.
- [91] Deuk Hee Park et al. "A literature review and classification of recommender systems research". In: *Expert Systems with Applications* 39.11 (Sept. 2012), pp. 10059–10072. ISSN: 09574174. DOI: [10.1016/j.eswa.2012.02.038](https://doi.org/10.1016/j.eswa.2012.02.038).

- [92] Michael J. Pazzani and Daniel Billsus. "Content-Based Recommendation Systems". In: *The Adaptive Web*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Vol. 4321. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325–341. ISBN: 978-3-540-72078-2. DOI: [10.1007/978-3-540-72079-9_10](https://doi.org/10.1007/978-3-540-72079-9_10).
- [93] David M. Pennock et al. "Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach". In: *UAI*. Morgan Kaufmann Publishers Inc., 2000, pp. 473–480.
- [94] M F Porter. "An algorithm for suffix stripping". In: (1980), p. 158.
- [95] Antti Puurula. "Scalable Text Mining with Sparse Generative Models". In: *arXiv:1602.02332 [cs]* (Feb. 2016). arXiv: 1602.02332.
- [96] Dorian Pyle. "Data Preparation for Data Mining". In: (1999), p. 466.
- [97] G H Rachman, M L Khodra, and D H Widyantoro. "Rhetorical Sentence Categorization for Scientific Paper Using Word2Vec Semantic Representation". In: *Journal of Physics: Conference Series* 801 (Jan. 2017), p. 012070. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/801/1/012070](https://doi.org/10.1088/1742-6596/801/1/012070).
- [98] Milos Radovanovic and Mirjana Ivanovic. "Text Mining: Approaches and Applications". In: *NSJOM*. 2008.
- [99] P C Rafeeqe and S Sendhilkumar. "A survey on Short text analysis in Web". In: *2011 Third International Conference on Advanced Computing*. Chennai, India: IEEE, Dec. 2011, pp. 365–371. ISBN: 978-1-4673-0671-3 978-1-4673-0670-6 978-1-4673-0669-0. DOI: [10.1109/ICoAC.2011.6165203](https://doi.org/10.1109/ICoAC.2011.6165203).
- [100] Alessandro Raganato. "New frontiers in supervised word sense disambiguation: building multilingual resources and neural models on a large scale". PhD thesis. 2018.
- [101] Haniyeh Rashidghalam, Mina Taherkhani, and Fariborz Mahmoudi. "Text summarization using concept graph and BabelNet knowledge base". In: *AIR*. IEEE, 2016, pp. 115–119.
- [102] Julian Risch, Samuele Garda, and Ralf Krestel. "Book Recommendation Beyond the Usual Suspects: Embedding Book Plots Together with Place and Time Information". In: *ICADL*. Vol. 11279. Springer, 2018, pp. 227–239. ISBN: 978-3-030-04256-1 978-3-030-04257-8. DOI: [10.1007/978-3-030-04257-8_24](https://doi.org/10.1007/978-3-030-04257-8_24).
- [103] Salvatore Romeo, Dino Ienco, and Andrea Tagarelli. "Knowledge-based representation for transductive multilingual document classification". In: *ECIR*. Springer, 2015, pp. 92–103.
- [104] Francesco Ronzano and Horacio Saggion. "Knowledge extraction and modeling from scientific publications". In: *SAVE-SD*. 2016.
- [105] Frank Rosenblatt. *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*. Tech. rep. VG-1196-G-8. Cornell Aeronautical Lab Inc Buffalo NY, Mar. 1961.
- [106] Y. Rubner, C. Tomasi, and L.J. Guibas. "A metric for distributions with applications to image databases". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Bombay, India: Narosa Publishing House, 1998, pp. 59–66. ISBN: 978-81-7319-221-0. DOI: [10.1109/ICCV.1998.710701](https://doi.org/10.1109/ICCV.1998.710701).
- [107] Ruslan Salakhutdinov and Geoffrey Hinton. "Semantic hashing". In: *International Journal of Approximate Reasoning* 50.7 (July 2009), pp. 969–978. ISSN: 0888613X. DOI: [10.1016/j.ijar.2008.11.006](https://doi.org/10.1016/j.ijar.2008.11.006).

- [108] G. Salton, A. Wong, and C. S. Yang. "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11 (Nov. 1975), pp. 613–620. ISSN: 00010782. DOI: [10.1145/361219.361220](https://doi.org/10.1145/361219.361220).
- [109] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information Processing & Management* 24.5 (Jan. 1988), pp. 513–523. ISSN: 0306-4573. DOI: [10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
- [110] Badrul Sarwar et al. *Application of Dimensionality Reduction in Recommender System - A Case Study*: tech. rep. Fort Belvoir, VA: Defense Technical Information Center, July 2000. DOI: [10.21236/ADA439541](https://doi.org/10.21236/ADA439541).
- [111] J. Ben Schafer et al. "Collaborative Filtering Recommender Systems". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. LNCS. Springer, 2007, pp. 291–324. ISBN: 978-3-540-72079-9. DOI: [10.1007/978-3-540-72079-9_9](https://doi.org/10.1007/978-3-540-72079-9_9).
- [112] Fabrizio Sebastiani. "Machine Learning in Automated Text Categorization". In: *ACM Computing Surveys* 34.1 (Mar. 2002). arXiv: cs/0110053, pp. 1–47. ISSN: 03600300. DOI: [10.1145/505282.505283](https://doi.org/10.1145/505282.505283).
- [113] Giovanni Semeraro et al. "Combining Learning and Word Sense Disambiguation for Intelligent User Profiling". In: *IJCAI*. ACM, 2007, pp. 2856–2861.
- [114] Parantu K. Shah et al. "Information extraction from full text scientific articles: Where are the keywords?" In: *BMC bioinformatics* 4.1 (2003), p. 20.
- [115] Shady Shehata. "A WordNet-Based Semantic Model for Enhancing Text Clustering". In: *2009 IEEE International Conference on Data Mining Workshops*. Miami, FL, USA: IEEE, Dec. 2009, pp. 477–482. ISBN: 978-1-4244-5384-9. DOI: [10.1109/ICDMW.2009.86](https://doi.org/10.1109/ICDMW.2009.86).
- [116] Utpal Kumar Sikdar and Björn Gambäck. "Language identification in code-switched text using conditional random fields and babelnet". In: *EMNLP 2016* (2016), p. 127.
- [117] Gerasimos Spanakis, Georgios Siolas, and Andreas Stafylopatis. "A Hybrid Web-Based Measure for Computing Semantic Relatedness Between Words". In: *ICTAI*. IEEE, Nov. 2009, pp. 441–448. ISBN: 978-1-4244-5619-2. DOI: [10.1109/ICTAI.2009.64](https://doi.org/10.1109/ICTAI.2009.64).
- [118] Michael Steinbach, Levent Ertöz, and Vipin Kumar. "The challenges of clustering high dimensional data". In: *New directions in statistical physics*. Springer, 2004, pp. 273–309.
- [119] Michael Steinbach, George Karypis, and Vipin Kumar. "A Comparison of Document Clustering Techniques". In: *Text Mining Workshop*. 2000, p. 20.
- [120] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. "Impact of Similarity Measures on Web-Page Clustering". In: *AAAI*, 2000, p. 7.
- [121] Xiaoyuan Su and Taghi M. Khoshgoftaar. "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence 2009* (2009), pp. 1–19. ISSN: 1687-7470, 1687-7489. DOI: [10.1155/2009/421425](https://doi.org/10.1155/2009/421425).
- [122] Dr S Vijayarani and J Ilamathi. "Preprocessing Techniques for Text Mining - An Overview". In: *International Journal of Computer Science & Communication Networks* 5 (2015), p. 10.

- [123] Zhibo Wang and Yanqing Zhang. "A Text Information Retrieval Method by Integrating Global and Local Textual Information". In: *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*. Atlanta, GA, USA: IEEE, June 2016, pp. 504–505. ISBN: 978-1-4673-8845-0. DOI: [10.1109/COMPSAC.2016.42](https://doi.org/10.1109/COMPSAC.2016.42).
- [124] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal Component Analysis". In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987), pp. 37–52. ISSN: 0169-7439. DOI: [10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).
- [125] Xindong Wu et al. "Data mining with big data". In: *IEEE transactions on knowledge and data engineering* 26.1 (2014), pp. 97–107.
- [126] R. Xu and D. WunschII. "Survey of Clustering Algorithms". In: *IEEE Transactions on Neural Networks* 16.3 (May 2005), pp. 645–678. ISSN: 1045-9227. DOI: [10.1109/TNN.2005.845141](https://doi.org/10.1109/TNN.2005.845141).
- [127] Yiming Yang and Xin Liu. "A re-examination of text categorization methods". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*. Berkeley, California, United States: ACM Press, 1999, pp. 42–49. ISBN: 978-1-58113-096-6. DOI: [10.1145/312624.312647](https://doi.org/10.1145/312624.312647).
- [128] Yujun Wen, Hui Yuan, and Pengzhou Zhang. "Research on keyword extraction based on Word2Vec weighted TextRank". In: *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. Chengdu, China: IEEE, Oct. 2016, pp. 2109–2113. ISBN: 978-1-4673-9026-2. DOI: [10.1109/CompComm.2016.7925072](https://doi.org/10.1109/CompComm.2016.7925072).
- [129] Arzucan Özgür, Levent Özgür, and Tunga Güngör. "Text Categorization with Class-Based and Corpus-Based Keyword Selection". In: *Computer and Information Sciences - ISICIS 2005*. Ed. by David Hutchison et al. Vol. 3733. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 606–615. ISBN: 978-3-540-29414-6 978-3-540-32085-2. DOI: [10.1007/11569596_63](https://doi.org/10.1007/11569596_63).