Graduate Theses, Dissertations, and Problem Reports

2021

Topic Modeling and Cultural Nature of Citations

Marie Coraline Dumaz
West Virginia University, mcd0029@mix.wvu.edu

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Part of the Condensed Matter Physics Commons, Data Science Commons, and the Numerical Analysis and Scientific Computing Commons

Recommended Citation
https://researchrepository.wvu.edu/etd/8226

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
Topic Modeling and Cultural Nature of Citations

Marie Dumaz

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Donald Adjeroh, Ph.D., Co-Chair
Aldo H. Romero, Ph.D., Co-Chair
Brian Powell, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2021

Keywords: Density Functional Theory ; Citation impact analysis ; Topic Modeling

Copyright © 2021 Marie Dumaz
Abstract

Topic Modeling and Cultural Nature of Citations

Marie Dumaz

Ever since the beginning of research journals, the number of academic publications has been increasing steadily. Nowadays, especially, with the new importance of online open-access journals and databases, research papers are more easily available to read and share. It also becomes harder to keep up with novelties and grasp an idea of the general impact of a given researcher, institution, journal, or field. For this reason, different bibliometric indicators are now routinely used to classify and evaluate the impact or significance of individual researchers, conferences, journals, or entire scientific communities. In this thesis, we provide tools to study trends in any given area of science. However, we focus our work on the field of Density Functional Theory (DFT), an important methodology in physics and chemistry, used to describe materials at the atomic scale, which has demonstrated an exponential number of related publications (with 5,339 in 2010, 9,931 in 2015 and 14,265 in 2019). We measure the specific impact of this theory by means of the citation record of the most used solid-state first principle \textit{ab initio} computational packages. Along with this analysis, we developed a Python library, pyBiblio, to compute basic bibliometric analyses on any Web of Science database. To get a deeper understanding of the field, we also use the Latent Dirichlet Allocation (LDA) algorithm on the abstracts of the papers published in this field to classify documents into topics of interest. Indeed, LDA is a generative topic modeling algorithm that creates an efficient and reliable distribution of documents over topics constructed from the papers’ vocabulary. We find that DFT is a collaborative field, with tight international clusters, especially in Europe and between countries where packages are developed. We study the evolution of topics over the years and find evidence for the specialization of the software packages, even if they include similar capabilities.
I would like to first thank my advisor Dr. Aldo Romero for adopting me into his research group. I am thankful for his constant support, guidance, patience and advice. I appreciate all the time Dr. Romero put into my research, with all the ups and downs that came with it.

I would like to thank Dr. Donald Adjeroh for our helpful bi-weekly meetings and his advising help. I offer my sincere thanks to Dr. Brian Powell for agreeing to be in my committee, and for being such a great teacher.

I am grateful for my parents and my sister who supported me and trusted me to finish my education in the United States. To my extended family, wherever you may be in the world, I am extremely grateful for the strength and encouragements you gave me.

To my significant other, Jacob, I am so thankful for the constant support and help. I am excited to start our new adventure together and see where life will take us.

I am very appreciative of my roommates Shannon Biega and Faith McGuire during this last year. Thank you for always cheering my day up.

Finally, thank you to my lab mates Uthpala Herath, Pedram Tavadze and Viviana Faride Dovale Farelo for their assistance and positive aura.
Table of Contents

Abstract ................................................................. ii
Acknowledgments ......................................................... iii
List of Figures ........................................................... vi
List of Tables ............................................................ ix

1 Introduction ........................................................... 1
   1.1 Executive Summary .............................................. 1
   1.2 Density Functional Theory ...................................... 2
   1.3 Contributions .................................................. 3
   1.4 Challenges ...................................................... 4
   1.5 Thesis Organization ............................................ 4

2 Literature Review .................................................... 5
   2.1 Bibliometrics .................................................... 5
   2.2 Topic Modeling .................................................. 6

3 Methodology .......................................................... 8
   3.1 Data Extraction and Organization .............................. 8
      3.1.1 Collection Process ....................................... 8
      3.1.2 Data Cleaning Process .................................. 9
      3.1.3 Data Limitations ....................................... 10
   3.2 Bibliometrics .................................................... 11
   3.3 Topic Modeling .................................................. 12
      3.3.1 Latent Dirichlet Allocation .............................. 12
      3.3.2 Coherence Score ......................................... 20
      3.3.3 T-distributed Stochastic Neighbor Embedding (t-SNE) . 23

4 PyBiblio ............................................................... 25
   4.1 Overview ........................................................ 25
   4.2 TITLECLEAN .................................................... 26
4.3 FundingClean ................................................................. 27
4.4 CIT_BY and PUB_BY ...................................................... 28
4.5 CIT_NUM and PUB_NUM ................................................... 28

5 Bibliometric Analysis .................................................. 30
  5.1 Overview ................................................................. 30
  5.2 Author Impact ........................................................... 32
  5.3 Country Impact .......................................................... 34
  5.4 Journal Impact ........................................................... 41

6 Topic Modeling .......................................................... 46
  6.1 Topic Distribution ....................................................... 46
  6.2 Topics over time .......................................................... 50
  6.3 Topics and Journals ....................................................... 54

7 Conclusions ............................................................... 58
  Publications ................................................................. 60
  References ................................................................. 61

Appendix A Software References .................................... 65
List of Figures

1.2.1 Percentage difference per year between the number of publications citing at least one software package and publications citing the original DFT papers. .......................... 3

3.3.1 Graphical representation of a LDA model [9] ........................................ 13

3.3.2 Pseudo-code for Online Variational Bayes LDA [25] ............................... 16

3.3.3 Mean $C_v$ score for LDA models with different number of passes (Left) or iterations (Right) .......................................................... 17

3.3.4 $C_v$ score for LDA models with different topic number .......................... 18

3.3.5 Mean and Standard deviation of the $C_v$ score computed on LDA models with different number of cores. Left: on 384 models for each number of cores. Right: on 1920 models for each number of cores, with ten different seed values ........................................ 19

5.1.1 Total number of publications per year ...................................................... 30

5.1.2 Total number of citations per year .......................................................... 31

5.2.1 Total number of publications as function of the number of different authors present in the paper .............................................................. 32

5.2.2 Number of researchers as function of their average productivity per year (in terms of publications) for all researchers with at least 2 papers in our database ........................................ 33

5.2.3 Number of citations over number of publications as function of the number of authors per paper ........................................................................ 33

5.3.1 Total number of publications per country and per number of countries in a publication. Left: Total number of publications as function of the authors’ countries (Only the top 15 are shown). Right: Total number of publications as function of the number of different countries listed in authors’ institutions ........................................ 35

5.3.2 Total number of publications as function of authors’ country (Only top 10 are shown). Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP .......................... 36
5.3.3 Total number of publications as function of the total number of citations, on the logarithmic scale. The figures show the 10 countries with the highest H-index for a given code. Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP.

5.3.4 H-index computed for each country (only top 10 shown). Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP.

5.3.5 Country collaboration networks for papers published before 2012 (included) and after 2012, with at least one author affiliated in the same country as the code. From top to bottom: ADF, Castep, Quantum Espresso.

5.3.6 Country collaboration networks for papers published before 2012 (included) and after 2012, with at least one author affiliated in the same country as the code. From top to bottom: Siesta, Turbomole, VASP.

5.4.1 Total number of publications with respect to the publishing journal (Only the top 15 are shown).

5.4.2 Total number of publications per journal (Only top 10 are shown). Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP.

5.4.3 Total number of citations (Normalized) over the journal impact factor defined on the website SJR — SCImago Journal & Country Rank.

5.4.4 Normalized total number of citations as function of the number of words present in the titles of the publications.

5.4.5 Word clouds generated from the titles. Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP.

6.1.1 Document distribution over topics.

6.1.2 Top 10 words making up each topic in the LDA model for \( k = 35 \).

6.1.3 T-SNE reduction for the LDA model for \( k = 35 \).


6.2.1 Normalized and weighted number of documents over the years, per topic for \( k = 35 \).

6.2.2 Comparison of the two topic with the lowest and highest growth, fit to the best exponential curve and powerlaw curve. Left: topic 18 with lowest powerlaw factor. Right: topic 24 with highest powerlaw factor.

6.2.3 Normalized and weighted number of documents over the years, per topic for \( k = 35 \).

6.2.4 Normalized and weighted number of documents per country (only top 15 shown), per topic, in different time periods.

6.2.5 Normalized and weighted number of documents per country (only top 15 shown), per topic, for articles published in 2020.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.1</td>
<td>Normalized and weighted number of documents per journal (only top 15 shown), per topic for $k = 35$</td>
<td>56</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Normalized and weighted number of documents per topic and per publisher (only top 10 shown), with only open access journals considered, for $k = 35$</td>
<td>57</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Results of example for CIT_BY ....................................................... 28
4.2 Results of example for PUB_NUM ..................................................... 29
5.1 Country collaboration network properties, before and after 2012 ................. 38
A1 References used to collect citations for each code...................................... 65
Chapter 1

Introduction

1.1 Executive Summary

The number of academic publications have kept increasing through the years, coming from a large number of sources over the world. Nowadays, the importance of online, free-access journals and databases is making research papers even more available. While it is great for spreading science, it becomes harder to keep up with new ideas and the general state of any research domain. In this thesis, we give tools to study trends in any field of science, by using bibliometric indices and topic modeling to analyze communities interactions and journal interests.

The bibliometric analysis helps in identifying scientific growth and work recognition. With different bibliographic indicators now used to classify research output, analysis of authorship and citations, it is becoming more frequent to assess the impact, quality, or the development of a scientist, a university, a field, or a subset of researchers. We use the number of citations and publications in our database to quantify the activity of countries, authors and journals. We focus our study on the field of Density Functional Theory (DFT), an active research domain in physics and chemistry mostly, but the tools developed in this work can be applied to any other field. Indeed, we created a Python library, pyBiblio, to compute basic bibliometric analyses on any Web of Science database.

In this work, we strive to study communities and their different research cultures. To do so, we consider another extremely important aspect of any research paper that is the subject studied. As research nowadays is more and more connected, countries are more open to collaborations and researchers start exploring outside of the common subjects of their fields. In DFT especially, as the methodology spans across several scientific domains, it is important to decipher specific topics to understand the evolution of the field. To do so, we use the Latent Dirichlet Allocation (LDA) algorithm on the abstracts of DFT-related research papers. LDA is a topic modeling model that generates a distribution of documents over some topics based on the words in each topic.
1.2 Density Functional Theory

First introduced by Hohenberg and Kohn in 1964 \[28\], the theory of Density Functional Theory was only made practical by Kohn and Sham in 1965 \[29\]. This methodology, fundamental to the electronic structure field in physics, chemistry and materials science, helps characterizing solid-state materials at the atomic scale and will later won the Nobel prize in chemistry in 1998, which was awarded to Walter Kohn and John Pople. DFT is the workhorse of electronic structure, and is used in the vast majority of works that use computational methods to characterize materials. Consequently, the number of publications employing this methodology increase exponentially, extending its span across new countries, journals and publishers.

While twelve DFT related papers are ranked in the one hundred most cited papers of all time \[45\], the original papers, where the theoretical framework was developed, are not ranked near the top. The most cited papers reported by Van Noorden et al. \[33\], and connected to DFT, are from the Canadian chemist Axel Becke and the US-based theoretical chemists Chenteh Lee, Weitao Yang, and Robert Parr, in positions 8 and 7 respectively. However, we should point out that the classification presented in reference \[45\] is outdated. By May 2020, the development of the so-called PBE exchange correlation functional \[38\] has more citations than the theoretical works included in the one hundred most cited papers \[45\].

Hence, the two DFT-related papers written by Becke and Lee et al, as well as the PBE exchange correlation functional by Perdew et al all have more citations than the original papers presenting DFT. This is an indication of the “obliteration by incorporation” \[34\] concept: the theory of DFT has become so universally accepted and used that the original papers are not always required to be cited anymore.

To circumvent the lack of citations of the original papers where DFT has been introduced, we use the following premise: instead of studying the citations of the original DFT papers, we analyze the citations of the 31 most used solid state and electronic structure computational packages in condensed matter theory which have DFT as the underlying theory. The complete list of selected software is available in the Methodology section. This premise overcomes the problem that the citations record can be reduced in time when the basic idea is so spread out that citations to the original work are not required or used in research publications anymore.

While it is true that the number of publications citing the original papers is still exponentially growing, minimizing the case for obliteration by incorporation, it is to a lesser extent than software citations. Figure 1.2.1 displays the percentage difference between the number of publications citing the original theoretical work of DFT or an electronic structure package. We can see a clear growth in percentage difference, obtaining even 375% in 2019, meaning that even though the original papers were mainly cited before the 2000s, and that some authors are still citing them nowadays, the majority of authors do not reference the original papers anymore. Most of the electronic structure codes started to be developed after the 2000s, which seems to indicate that authors have now the tendency to cite the packages without acknowledging the theoretical work behind DFT.
This is a clear indication that using the original papers does not provide the clearest picture on how the scientific electronic structure community works. In that respect, software users who cite the original computational implementations define clearer citation networks on which to perform an analysis in the field of electronic structure methods. As a publication accompanies any computational package, and in most cases, users are advised to cite these papers when the package is used, we expect that users’ citations are more representative of the field.

1.3 Contributions

For this research, we study the trends of research papers in the field of electronic structure, analyzing the collaborations and the extent of those connections across authors, journals, publishers and countries. While previous work on quantifying the impact of DFT through publication records focused on creating databases depending on common topics or subclassifications, our approach uses a different methodology as we base our analysis on the computational packages that are used to study the materials, independently of the field of application. We also use topic modeling on the abstracts of research papers to study how different research communities interact. We analyze topic trends over the years, international interests and the journals preferences, including the way countries collaborate with each other by creating country collaboration networks, where clusters display strong connections. In the process, we developed tools to study paper attributes and created a python library called pyBiblio available freely on GitHub. PyBiblio includes several functions that take in Web of Science data and computes quantitative metrics such as number of citations and publications depending on the parameter of choice. It also contains two methods to clean the titles
of the papers and the column holding funding agencies and grant numbers.

1.4 Challenges

The challenge of our work is mainly two-fold and includes:

- Volume: the size of our dataset is significant, with over 120,000 research papers and 300,000 different words. Consequently, the time and resources needed to perform any analysis are amplified.

- Complexity: academic publications are intrinsically elaborate and difficult to process for machines, in particular in the DFT field. It is not rare to find equations, scientific notations, in-text citations or journal copyrights in the papers’ abstracts. However, they are very hard to comprehend, associate or even identify for computers. For example, a set of numbers in a paper can refer to details of the approximations used in the calculation, to experimental conditions or to actual results. It implies that we have to perform a feedback analysis on different models to guarantee the correct use of words.

1.5 Thesis Organization

This chapter provided a brief introduction to the field of Density Functional Theory and the active research community behind it. It touched on the need to quantify the exponential number of papers being published in this domain each year, and the methods to do so.

The thesis is organized as follows:

**Chapter 2** reviews the literature work for bibliometric analysis and topic modeling from research papers.

**Chapter 3** explains the theory behind the different algorithms and methods employed in this work.

**Chapter 4** gives a description of the Python library pyBiblio created for this thesis.

**Chapter 5** analyzes the results of the bibliometric analysis for our dataset of research papers in the field of Density Functional Theory.

**Chapter 6** explores the experiments made with the Latent Dirichlet Allocation algorithm and the results from it, exposing trends in the DFT scientific community.

**Chapter 7** summarizes the main points of this thesis and describes potential future work.
Chapter 2

Literature Review

In this chapter we will discuss the literature review proposed for bibliometric and topic modeling analyses of research papers.

2.1 Bibliometrics

Evaluating research papers has been an important discussion in the academic world and was first started by Garfield in the 1970s [15], who understood the potential of citations as quantitative estimates. From this discussion stems the field of bibliometrics (often equally called scientometrics) that is now widely accepted as a way to empirically study research fields. As a token of its importance, the journal *Scientometrics* was launched in 1978 to focus on the “science of science”. Nowadays, indicators such as the H-index [24] provide quantitative measures, for all levels of research: from the individual researcher, to the publishers and institutions. However, the field of bibliometrics is always evolving and creating more advanced indicators, especially with the rise of free large datasets on the Internet [32]. Indeed, bibliometric measures might soon become a common way to evaluate and compare authors, institutions or journals. In fact, the H-index of any author is already freely available on the Google Scholar database.

Indeed, bibliometric analysis has been invigorated in the last few years due to the creation of many different bibliographic databases. Based on those databases, authorship analysis of research articles has been performed by many different scholars, in a large set of different research fields. In particular, as a metric to quantify the research output and achievements of the scientific community [39] [12]. Examples of this approach include the assignment of authorship credit [19], research performance [36], the classification of journal impact factors [16] [2], the definition of trends in collaborative research [4], the evaluation and quantification of international scientific collaboration [17], national and university trends in publication output and the existence of scientific networks [3].

The bibliometric impact of DFT has already been approached by other research groups, for
example by classifying the different topics from a historic perspective [21], a materials focused analysis [22], from the marker paper approach or field classification [23]. However, our approach is different as we analyze citations of solid state packages.

2.2 Topic Modeling

The interest in topic modeling was first sparked in the years 1990 by two new algorithms: Latent Semantic Indexing (LSI) [13], later expanded as Probabilistic LSI (pLSI) [27], and Latent Semantic Analysis (LSA) [30] also expanded years later as Probabilistic LSA (pLSA) [26]. However, it is only after the development of the Latent Dirichlet Allocation algorithm that topic modeling started expanding to fields outside of computer science, and became a real tool in the analysis of research trends.

Indeed, LSA and LSI both use Singular Value Decomposition (SVD) on a $m \times n$ matrix where each element represent the number of times a word appears in a given document. Those two methods are similar to dimensionality reduction algorithms, which means they are quite fast to compute but also a lot harder to interpret. They are also not a good fit for non-linear dependencies as they are linear models. Moreover, it is impossible to use an “already-trained” model on new data as it is not a generative method. pLSA and pLSI both try to overcome those different issues by using a probabilistic approach instead of SVD. Those two new methods start thinking of documents as probabilities over topics and topics as probabilities over words, which is also LDA’s main idea.

The Latent Dirichlet Allocation algorithm was first introduced by Blei, Ng and Jordan in 2003 [9]. Indeed, LDA is a generative process and can accurately estimate the topic distribution of a new document, once appropriately trained. While pLSI and pLSA are prone to overfitting, LDA avoids this issue by creating a Bayesian version of pLSA/pLSI. As a result, even though LDA only works with one fixed vocabulary, it generalizes better to new data and is more routinely chosen for topic modeling applications.

Several variations of the original algorithm have been proposed since. We can cite the collaborative topic modeling algorithm [46] that combines LDA with collaborative filtering to recommend scientific articles, dynamic LDA [8], created to study the evolution of topics over time or relational topic models [10] building links between documents.

Another LDA variant is an online variational Bayes algorithm for LDA based on online stochastic optimization created by Hoffman, Blei and Bach [23], which is the chosen approach for this thesis. This particular algorithm is faster than others thanks to online learning, which is a type of learning feeding only parts of the data at one time to the model. Speed is an important factor for our large dataset, making it a key factor in our decision.

In parallel, some researchers focused on testing the limits of LDA and optimizing ways of pre-processing text and evaluating the results of the algorithms. Some examples of this work include Schofiled et al [42] testing different pre-processing technique such as the removal of stopwords or
stemming while Syed and Spruit [44] compared the results of LDA on the abstracts of papers against the entirety of the text. Mimno et al [35] examine and classify “bad” topics into four categories, providing a new way of evaluating topic models related to pointwise mutual information, a famous coherence score.

Only a year after the creation of LDA, Griffiths and Steyvers applied this new method to abstracts of papers publish in the PNAS journal [18]. They showed how LDA can be used to efficiently summarize large corpora into meaningful set of words as well as studying the increase and decrease of interest to some topics. Since then, topic modeling have had many applications in different scientific fields such as manufacturing [47] or medicine [48] but it has also been used as a tool to study interdisciplinary collaborations [37] and trends over time [20]. As for this study, most of the works cited here only use the abstracts of papers to create the topic models, as it often is the only part freely available.

Even though topic modeling has been applied to many different fields, only very little research in computing trends from texts of physics or DFT papers has been made. Moreover, we apply our methodology to condensed matter physics in particular, which is a narrower area than other works like the ones cited previously. Compared to interdisciplinary studies, or even papers focusing on publications from one journal, our dataset and consequently vocabulary is more specific, resulting in the formation of “sub-topics” rather than topics.

Some papers created similar analyses using the Physics and Astronomy Classification Scheme (PACS) numbers for the papers, which are a set of hierarchical numbers classifying physics and astronomy fields and subfields [1111]. Those studies derive the papers’ topics based on the PACS classification instead of creating topics from topic modeling algorithms, as proposed here.
Chapter 3

Methodology

In this chapter, we describe the methodology followed to download, clean and organize data; as well as creating, visualizing and evaluating LDA models.

3.1 Data Extraction and Organization

3.1.1 Collection Process

We obtained our citation database from the Institute for Scientific Information (ISI), which is a part of Clarivate Analytics, and that has the graphical interface “Web of Science” (WOS). The bibliographic search was performed since the first citation up to December 2019. We also repeated this process for papers published in 2020, in order to create a test set.

As we focus on the impact of electronic structure calculations from the user perspective, we selected the most well-known density functional theory packages used to study crystalline systems (independently from their use of a free or commercial license). These computational codes are all mostly used to describe periodic (crystal/amorphous/nanostructure) systems. From the large variety of packages we decided to center our analysis on the six most cited packages: VASP, Castep, Quantum Espresso, Siesta, ADF and Turbomole. On top of these six packages, we included all publications citing at least one of the following codes: Abinit, BigDFT, Casino, Conquest, CP2K, Crystal, Dacapo & ASE, Empire, EPW, Exciting, FHI-aims, FreeON, GPAW, JDFTx, NWChem, Octopus, Onetep, PySCF, QBOX, QMCPack, QuantumATK, RMG, TransSiesta, Wien2K and Yambo. The exact references considered for any of the codes are available in Table A1 in Appendix A.

The citation metrics were acquired by searching the references of all of the papers associated with each software package. The search was completed on the WOS database website. After locating each paper, the option “Times Cited” was selected in order to see all of the occasions other people had cited the specific paper. The whole list of citations was downloaded with all possible entries created by WOS.
3.1.2 Data Cleaning Process

Only a few computational libraries exist that can read those WOS files. Due to the flexibility this package allows for the analysis, we selected Bibliometrix [3], which works on R. Although it is an exceptional package when using already existing functions on the whole data set, some more precise analyses were easier to make and visualize with python tools. Hence, the dataframes created with the function readFiles from the Bibliometrix library were converted into Comma Separated Values (CSV) files that were used with our in-house Python script.

However, before exporting the dataframes to files, some pre-processing steps were taken. First, duplicates, identified using the Web of Science unique identifier of each publication, were removed. Then, countries were extracted from the “C1” column using the method MetaTagExtraction in the Bibliometrix library. For the bibliometric analysis, the pre-processing steps stop here.

For topic modeling, we use the abstract of a manuscript, as we can easily access it in the database column “AB”. A number of pre-processing steps were taken to clean the text and keep only the most relevant words, in order to capture the meaning of the paper. Here is the list, in order, of the steps:

1. Remove rows with no abstracts. This deletes 1,537 papers from the database.

2. Remove the following groups of words: vertical bar, center dot, double left right arrow, double right arrow, right arrow, left arrow, up arrow, down arrow. Those group of words are commands embedded in the text to represent arrows, dots, etc. They do not hold meaning within themselves as they can be used in a lot of different ways and mathematical formulas.

3. Remove characters between two dollar signs. Dollar signs are used as commands, often to create equations. An exception to this rule occurs if an underscore is detected between the two dollar signs. This particular pattern is a sign that this command is used to describe a chemical formula and not an equation. Chemical formulas are important for our study as different molecules appear in different sub-fields of DFT.

4. Remove publishing and copyright tags.

5. Remove DOI numbers.

6. Remove all non-alphanumeric characters.

7. Tokenize the abstracts. Tokenizing is the process of breaking down text into units, hence creating a list of tokens from the text.

8. Change all text to lowercase.

9. Remove stopwords. Stopwords are words that are so common that they span across fields. We used the standard NLTK list of stopwords that is mostly composed of everyday words such as “can” or “very” and added the following words to fit our particular dataset:
10. Remove all single letters and single digits. This is defined by one letter or digit between two white spaces, or at the end or beginning of a sentence. Single letters usually come from mathematical expressions where the letter is used to represent a variable in a short way. Single digits are often used to quantify a value such as "3 EV". They both do not hold meaning intrinsically so they are removed to create better topics.

11. Create bigrams and trigrams. Bigrams refer to a group of two tokens commonly continuous, that are linked as one by an underscore. Trigrams are the same process but for three tokens. An example of this would be "ab initio" or "density functional theory".

12. Tokens are lemmatized. This technique finds the root of a word. For example, both the words "studies" and "studying" become "study".

Once all of those cleaning steps are taken, we can create the dictionary of tokens. With the dictionary, word frequencies are computed and it allows us to remove tokens with a frequency too low or too high. We only keep tokens if they are in at least two documents and in less than 20% of the documents. The idea is that if a word appears too often, it is less likely to hold significant meaning and could create unbalanced topic models.

We then convert the corpora to a Bag of Words (BoW) in order to convert text to a more standard format. A BoW consists of tuples containing the numerical representation of a token, given by the dictionary, and the number of times this token appears in the document. Hence, a BoW does not maintain the order of tokens, and is only concerned with whether or not a token is in the document. This is not a problem for our study, as the BoW will be used to find topics in the corpora, which are easily inferred from a disordered list of words.

In the end, our data is composed of: a complete dataframe with each papers' attributes, a dictionary to map tokens to numbers, one corpus of tokens and one Bag of Words holding the vocabulary.

3.1.3 Data Limitations

The database Web of Science is very accessible and with a simple graphical interface, but it has also been criticized as its entries mostly come from journal titles in English (though it is the predominant language in scientific literature) and covers very few citations in books, web links and conference data.
proceedings. This is a difference with respect to other databases such as Scopus, Citeseer, or Google Scholar. Though this lack of references could be a handicap for the proposed analysis, the fact that the citation list is more controlled with respect to other databases allows us to extract more general conclusions on trends and author behavior.

Additionally, some data is not parsed correctly by WoS, resulting in missing data. For example the tags "DT" and "TC", respectively the document type and the number of citations, are always documented on the website. However, when downloading their records to text files, they are sometimes not included and consequently cannot be read into our dataframe.

Moreover, some references, such as books or conference proceedings, could not be added as they are not available in WOS. An example of this limitation would be the Wien2K manual [7], recommended for reference by the Wien2K developers but not included in the WOS database.

3.2 Bibliometrics

We analyze the entirety of the dataset up to 2019, to give us a general idea of the state of the DFT field, as well as the six software with the highest number of citations, focusing our views on more specific research communities. In decreasing order of number of citations, the six main packages studied in this work are: VASP, Castep, Quantum Espresso, Siesta, ADF and Turbomole.

We compare trends both in terms of publications, which refers to papers citing at least one of the software selected and from a citations point, which defines the number of citations those publications got.

The bibliometric analysis was created with both R and Python. In Python, we used an in-house library called pyBiblio, that was entirely built for this thesis, and is now made available for free on Github. The details of this Python library can be found in Chapter [4]. For each figure, we first delete all rows with null values in the field of interest. For example, if a paper have no institutions registered but does have the name of the journal it was published in, that particular paper would not be used to compute the number of countries per papers but would be represented in the number of publications per journal figure. This option was chosen as we had enough data and removing those rows would not substantially affect results. In the end, very few entries were removed: for example, out of 61 640 VASP citations, only 87 did not have any affiliated countries.

The country collaboration networks in Section [5,3] were created with the R package Bibliometrix [3], specifically the method biblioNetwork. First, a sparse matrix \( A_{n \times m} \) is built where \( n \) is the number of documents and \( m \) is the number of countries in the whole dataset. Elements in \( A \) are either 0 or 1, indicating if document \( i \) was created in country \( j \). Then, the network is obtained through \( B_{m \times m} = A' \times A \) where \( B[i, j] \) is the number of collaborations between countries \( i \) and \( j \). The diagonal of \( B \) (or element \( B[i, i] \)) is the total number of publications written in country \( i \).

Each node of the network represents a country and there exists an edge between two countries if
and only if they both collaborated on at least one publication. The method \textsc{networkPlot} provides a visualization of the network where nodes are colored by clusters and distanced by the number of collaborations.

For network analysis, we used different observables to characterize the network properties, in particular:

- **Density**: This is simply the ratio of actual edges in the network to all possible edges in the network. In the undirected network defined for the citation bibliographic record for each DFT package, there could be a single edge between any two nodes, but seen in the visualization, only a few of those possible edges are actually depicted. Network density gives a quick sense of how closely knit a network is. This value ranges between 0 and 1.

- **Diameter**: This is the longest of all shortest paths. This number provides a sense of the network’s overall size, the distance from one end of the network to another.

- **Transitivity**: like density, expresses how interconnected a graph is as a ratio of existing triad connections with all possible triad connections. The idea is that if A is connected to B and B is connected to C, what is the probability that A is connected to C. Transitivity is scaled to the range from 0 to 1.

### 3.3 Topic Modeling

#### 3.3.1 Latent Dirichlet Allocation

**Latent Dirichlet Allocation intuition**

LDA is a generative probabilistic model that assumes each document is a mixture of topics and each topic is a mixture of words. These combinations are actually probability distributions over latent topics and are computed on the basis that words that often appear in the same documents or close to each other are probably related to the same topic. In fact, a LDA model creates a set of topics where each topic is a probability distribution over a corpus of words drawn from $\beta_k \sim \text{Dirichlet}(\eta)$, and each document is a probability distribution over topics drawn from $\theta_k \sim \text{Dirichlet}(\alpha)$.

The generative process of LDA first starts by choosing $\beta_k \sim \text{Dirichlet}(\eta)$. Then, for each document $d$ in a corpus $D$:

1. Choose $\theta \sim \text{Dirichlet}(\alpha)$

2. For each of the $N$ words $w_n$:

   - Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   - Choose a word $w_n$ from $p(w_n|z_n, \beta)$, a Multinomial probability conditioned on the topic $z_n$
Here, we define $M$ as the total number of documents, $N$ as the total number of words in a document $m$ and $k$ as the number of topics. As shown in Figure 3.3.1, LDA spans over three layers. The parameters $\alpha$ and $\eta$ are corpus-level parameters, which means that they are sampled only once per corpus. $\theta$ is document-level variables, sampled once per document. Finally, variables $z_n$ and $w_n$ are at word-level: they are sampled once for each word in each document. $w_n$ is also the only observable variable, picked from the bag of words given as input.

We can also represent the LDA generative process with the following formula for the joint distribution for the observed and hidden random variables:

$$p(W, Z, \theta, \beta, \alpha, \eta) = \prod_{j=1}^{M} p(\theta_j; \alpha) \prod_{i=1}^{k} p(\beta_i; \eta) \prod_{n=1}^{N} p(Z_{j,n}|\theta_j)p(W_{j,n}|\beta, Z_{j,n})$$

where $p(\theta_j; \alpha)$ is the Dirichlet distribution of topics over words, $p(\beta_i; \eta)$ is the Dirichlet distribution of documents over topics, $p(Z_{j,i}|\theta_j)$ is drawn from a Multinomial distribution and represents the probability of getting a topic given a document and $p(W_{j,t}|\beta, Z_{j,t})$ is also drawn from a Multinomial distribution and represents the probability of getting a word given a topic.

With the generative process, LDA produces data in the form of documents, or set of words. From this sample, we need to use statistical inference in order to obtain the posterior distribution of the hidden variables, given a document. Put simply, we want to reverse the generative process to deduce the probability distribution of topics over a given document. That way, LDA would be able to take in new, unseen documents, and tell what topics are present in it. The posterior distribution of LDA cannot be directly computed which is why we need to apply inference techniques.

There exists many different methods that can be implemented to approximate inference, including Markov chain Monte Carlo (MCMC) and Gibbs sampling, both commonly used with LDA. However, in this work, we describe the Variational Inference (VI) technique, preferred by the authors of both the original LDA [9] and online LDA [25]. Even though MCMC is more precise, VI often is the favored method when dealing with large datasets, as it usually is computationally cheaper.

VI is used to approximate the intractable posterior distribution $p$ of latent variables with a tractable one $q$. The main idea behind VI is to pick a subfamily of distributions over the latent vari-

---

Figure 3.3.1: Graphical representation of a LDA model [9]
ables with its own variational parameters and finding the point in the subfamily that is closest to our true posterior. The similarity between the two distributions are measured with the Kullback-Leibler (KL) divergence. The goal is now to minimize the KL divergence, so that the VI approximation is as close as possible. However, the KL divergence cannot be be minimized exactly, so we apply the Jensen’s inequality on the log probability of the observations to get the Evidence Lower Bound (ELBO). To minimize the KL divergence, we need to maximize the ELBO. The exact VI process is described in the next section, as it is specific to online learning.

**Online Latent Dirichlet Allocation**

In this thesis, topic modeling is achieved through the LDA variant commonly called Online LDA and proposed by Hoffman, Blei and Bach [25]. This method was chosen for two reasons. First, it is the most practical, as this particular model is the one implemented by the Python library Gensim [40]. Most importantly, it converges faster than the original LDA algorithm as it is trained in batches and does not require a full pass through the corpus at each iteration. This is extremely important for our study as our dataset is fairly large.

The intuition and general method of LDA described previously stays untouched in the online LDA algorithm. For this reason, we will explain the VI inference here, assuming all notations, equations and processes developed above. As this explanation was taken from [25], we change the notation for the number of documents in a corpus previously called \( N \), now \( D \). This modification is important to understand the algorithm of online LDA presented in Figure 3.3.2. The goal of VI is to maximize the Evidence Lower Bound, or minimizing the KL divergence between the subfamily \( q(z, \theta, \beta) \) and the posterior \( p(z, \theta, \beta|w, \alpha, \eta) \). Let’s define the ELBO as:

\[
\log p(w|\alpha, \eta) \geq \mathcal{L}(w, \phi, \gamma, \lambda) \triangleq \mathbb{E}_q[\log p(w, z, \theta, \beta|\alpha, \eta)] - \mathbb{E}_q[\log q(z, \theta, \beta)] \tag{3.1}
\]

knowing that

\[
q(z_{di} = k) = \phi_{dw,i,k}; \quad q(\theta_d) = \text{Dirichlet}(\theta_d; \gamma_d); \quad q(\beta_k) = \text{Dirichlet}(\beta_k; \lambda_k)
\]

We can now factorize equation 3.1, which brings all “per-corpus” terms into a summation over documents, then divided by the number of documents \( D \) in the corpus.

\[
\mathcal{L}(w, \phi, \gamma, \lambda) = \sum_d \{ \mathbb{E}_q[\log p(w_d|\theta_d, z_d, \beta)] + \mathbb{E}_q[\log p(z_d|\theta_d)] - \mathbb{E}_q[\log q(z_d)] \\
+ \mathbb{E}_q[\log p(\theta_d|\alpha)] - \mathbb{E}_q[\log q(\theta_d)] + \mathbb{E}_q[\log p(\beta|\eta)] - \mathbb{E}_q[\log q(\beta)] \}/D
\]

By then expanding the expectations to be functions of the variational parameters, we find that
the variational objective relies only on \( n_{dw} \), which is the number of times the word \( w \) appears in document \( d \). Here, \( W \) is the size of the vocabulary and \( \ell(n_d, \phi_d, \gamma_d, \lambda) \) represents the contribution of document \( d \) to the ELBO.

\[
\mathcal{L} = \sum_d \sum_w n_{dw} \sum_k \phi_{dwk} [\mathbb{E}_q [\log \theta_{dk}]] + \mathbb{E}_q [\log \beta_{kw}] - \log \phi_{dwk} - \log \Gamma(\sum_k \gamma_{dk}) + \sum_k (\alpha - \gamma_{dk}) \mathbb{E}_q [\log \theta_{dk}] + \log \Gamma(\gamma_{dk}) + \log \Gamma(K\alpha) - K \log \Gamma(\alpha) + (\log \Gamma(W\eta) - W \log \Gamma(\eta))/D + \log \Gamma(\sum_w \lambda_{kw}) + \log \Gamma(\lambda_{kw}))/D
\]

\[
\triangleq \sum_d \ell(n_d, \phi_d, \gamma_d, \lambda)
\]

To optimize \( \mathcal{L} \), the online LDA algorithm applies the coordinate ascent over the variational parameters \( \phi, \gamma, \) and \( \lambda \). Coordinate ascent is similar to a gradient ascent, often used in Machine Learning but it only updates one parameter at a time unlike gradient ascent. Coordinate ascent will try to find a local maximum to \( \mathcal{L} \) by alternating between moving on the x-axis and the y-axis. Finally, we find that we can optimize our parameters with the following functions:

\[
\phi_{dwk} \propto \exp\{\mathbb{E}_q [\log \theta_{dk}] + \mathbb{E}_q [\log \beta_{kw}]\}; \quad \gamma_{dk} = \alpha + \sum_w n_{dw} \phi_{dwk}; \quad \lambda_{kw} = \eta + \sum_d n_{dw} \phi_{dwk}
\]

As VI is an extension of the Expectation Maximum algorithm, we can also see this process through two steps:

- **Expectation step (E-step):** holding \( \lambda \) fixed, iteratively update \( \gamma \) and \( \phi \) until convergence.
- **Maximization step (M-step):** updating \( \lambda \) given \( \phi \).

Hoffman, Blei and Bach [25] keep this analogy through their paper to explain the online LDA algorithm proposed in Figure 3.3.2 in the following way. On the \( t^{th} \) iteration, we first find local optimal values for \( \gamma_t \) and \( \phi_t \) while holding \( \lambda \) fixed. Then, we compute \( \hat{\lambda} \) given \( \phi_t \), which is the setting of \( \lambda \) that would be optimal if the entire corpus consisted of the document \( n_t \). Finally, we update the topics \( \lambda \) by using a weighted average of its previous value and \( \hat{\lambda} \). The weight \( p_t \triangleq (\tau + t)^{-\kappa} \) given to \( \hat{\lambda} \) introduces two new parameters \( \tau \) (called \( \tau_0 \) in the algorithm but simplified as \( \tau \) in this thesis) and \( \kappa \). \( \tau \) slows down the early iterations of the algorithm and \( \kappa \) controls the rate at which old values of \( \hat{\lambda} \) are forgotten.
**LDA implementation and training**

We implemented LDA with the Python library Gensim [40] that already has a handy set of functions to create, evaluate and visualize LDA models. The inputs to the models are the bag of words created with the methodology described in Section 3.1. We used two bag of words: one with all documents in our dataset from 1990 to 2019, and one with only documents published in 2020. The idea was to use the 2020 corpus as a way to test our LDA model and our analyses, as it can help us determine how well our model deals with new, unseen data. This means that we selected the best parameters and trained our LDA models on the corpus with documents from 1990 to 2019, and only fed the 2020 corpus to the trained model, to get the topic distribution over the unknown corpus.

During training, we tune the following hyperparameters:

1. $k$ is the number of topics. It is a know and fixed value decided ahead of training the model. However, $k$ also described the dimensionality of the Dirichlet distribution $\theta$ and the dimensionality of the topic variable $z$.

2. $\alpha$ is the parameter on the per-document topic distribution. It is the parameter for the Dirichlet distribution $\Theta$, which controls the distribution of topics in a document. The higher $\alpha$ is, the most likely it is that each document will contain a mixture of all topics.

3. $\eta$ is the parameter on the per-topic word distribution. It is the parameter for the Dirichlet distribution $\beta$, which controls the distribution of words in a topic. The higher $\eta$ is, the more likely it is that each topic contains a mixture of a lot of words.

4. $\tau \geq 0$ slows down the early iterations of the algorithm.
5. $\kappa$ is an exponential decay rate; meaning it controls the rate at which old values are forgotten. It is kept in the range $(0.5; 1]$ to guarantee asymptotic convergence.

6. passes is the number of times the model is trained over the whole corpus. It is similar to the term “epoch” in Machine Learning.

7. iterations is the maximum number of iterations through the corpus when inferring the topic distribution of a corpus (the number of iterations at the E-step).

Because of the significant size of our dataset, and the 7 parameters to tune, we divided training into several steps. All models were evaluated with the $C_v$ coherence score defined in Section 3.3.2. First, we wanted to know what were the best values for iterations and passes. For iterations, we ran four models with four different values of $k$: 10, 30, 50 and 70. However, for each of those four models, we changed the value of iterations between 10, 50, 100, 200, 300, 400, and 500. This gives us a total of 28 models. All other parameters were not specified. At the end, we computed the arithmetic mean of the $C_v$ score given by each model over the different values of iterations. We repeated the same process for passes, with values in the range (5, 10, 15, 20, 50, 100, 200). The reported results are available in Figure 3.3.3. For passes, on the left panel, we can see that after 50, the mean coherence score of the four models with different $k$ values stabilize, which is why we chose passes = 50 for the rest of our experiments. Similarly, for the number of iterations on the right panel, the mean score stabilizes after 50. However, 50 is often considered too restrictive for this parameter. Another good choice would be 200 but it would then make training longer. We chose to use iterations = 100 as a good middle-ground between time and performance.

Next, we created 16 models with a different number of topics, from 5 to 80, while keeping all other parameters null, except passes = 20 and iterations = 100. Once again, we computed the $C_v$ score for each model and reported the results in Figure 3.3.4. With this analysis, we chose to
tune the parameters left for 15, 25, 35 and 45 topics. We decided to study models with different \( k \) values, to examine the correlation between their topics and the coherence score. For example, from this figure, we expect the model for 15 topics to have a lower quality of topics and the model for 35 topics to give more coherent topics.

![Figure 3.3.4: \( C_v \) score for LDA models with different topic number](image)

For the remaining parameters, we created 480 models from combinations of the following values:

- \( \alpha \) in \(['\text{symmetric}', 0.1, 0.5, 1]\). \( \alpha \) is meant to be a vector of \( k \) values, representing our a-priori knowledge on topics distribution. However, here, we do not have such information. For this reason, we will use other parameter values that can help restrict our models. If ‘symmetric’ is passed for \( \alpha \), then the model is trained with a fixed symmetric prior per topic, which is a vector of \( k \) identical values. If a scalar is passed, this value will be used to create the vector.

- \( \eta \) in \(['\text{symmetric}', 0.1, 0.5, 1]\). \( \eta \) is a Dirichlet parameter, like \( \alpha \), which is why they have the same values.

- \( \tau \) in \([16, 32, 64, 128, 256]\). These parameters are the ones used by the authors of online LDA \cite{25} when evaluating their proposed algorithm.

- \( \kappa \) in \([0.5, 0.6, 0.7, 0.8, 0.9, 1]\) since, as explained above, \( \kappa \) is constrained in the range (0.5;1].

We ran those 480 models for each of the 5 different values of \( k \) and created a csv file with the parameters’ values and the resulting \( C_v \) score. However, we could not save all models to disk as it would be more memory space than available. For this reason, we only create one more model for each \( k \) value, with the parameter combination giving the highest \( C_v \) score. The final five models are then saved and used through the rest of this thesis to ensure the correct analysis of our topics.
To speed up training, we run the parameter tuning with a larger number of cores than when re-creating the models with the best parameters. After finding some discrepancies in the reported $C_v$ values of models created during training, and new models, we wanted to check if the number of cores used for parallelization of the algorithms were the source of those differences. Figure 3.3.5 renders the results of our experiments to quantify the impact of parallelization. We should note that the stochastic optimization process used in online LDA includes the randomization of data partition, which makes it impossible to ensure reproducibility if coupled with parallelization.

For each number of cores, we trained LDA models on 15 topics, as a lower number of topics is faster to run, but on different combinations of the four parameters $\alpha$, $\eta$, $\kappa$, $\tau$. In the left panel of Figure 3.3.5, we show the mean $C_v$ score obtained by those models for each number of cores, as well as the standard deviation, represented by an error bar. In the right panel, we re-created the same experiments, but with a lower number of parameter combinations as we also switched the seed value 10 times.

We can observe a light downward trend as the number of cores increases. However, the change in coherence score is too small to be significant. It is furthermore confirmed by the standard deviations, that are extremely similar across the the number of cores, with a range of only 0.0038 for the left plot, and 0.0076 for the right panel. From these experiments, we concluded that the number of cores used for the parallelization process does not have a significant impact to impair our results, which is to be expected. It is normal to have small changes between models, as we cannot reproduce the exact same results everytime and those experiments showed that parallelization is not aggravating the problem.
Derive trends from topics

With LDA, each document is a mixture of different topics. Indeed, when feeding a document to a trained LDA model, we get a probability distribution of topics for this document. For example, we could get a 0.2 probability for topic 1, 0.3 for topic 2 and 0.5 for topic 3. As a document can touch on more than one subject at a time, this is a very handy functionality of LDA. To discover trends and interests, we assign those topics to our publications bibliometric attributes. We assign topics by considering the probabilities as weight. We create a matrix $A$ where each row represents a document, and each column represents a topic. The elements of this matrix are the probabilities for a given topic to be assigned to a given document.

This matrix is particularly useful for the analyses made of topic distributions over countries, journals and years. This way, when computing the number of publications per topics per country for example, we can use those weight to create a more precise analysis. For these analyses, we create a matrix where each row is a topic and each column is a country. If a document was created in the USA for example, then each topic would add their respective weight for this document in the “USA” column. These analyses are also all normalized, which means each value in the matrix is divided by the sum of all values in the $A$ matrix, which is equal to the total number of documents.

3.3.2 Coherence Score

Coherence scores are common and accepted method of evaluation for LDA models. They seek to measure how coherent topics are. That is, if the words creating a topic can really form a semantically correct group. Hence, most coherence scores rely on words proximity and document frequency. In this work, we use the so-called $C_v$ measure \[41\], which is the coherence score with the strongest correlation to human rankings. The $C_v$ measure can be calculated in four steps that are laid out below.

Here, we define $W$ as a topic set, which contains $m$ words. We also set $N$ as the total number of documents in our corpus. For a more simplified explanation of the calculations, we will use the following example. We have three documents ($N = 3$) as such “The game is a team sport”, “The game is played with a ball” and “The game demands great physical efforts”. Our topic consists of four words ($m = 4$) such that $W = \{\text{game}, \text{sport}, \text{ball}, \text{team}\}$. This example is inspired by Röder, Both and Hinneburg \[41\].

**Segmentation**  The first step is to segment the set $W$ into “a set of pairs of subsets of $W$” \[41\]. The idea is to create a pair between every word in the topic, to then assess how well each word supports the others. The technique of segmentation used for $c_v$ is called “one-set” and is defined as such:

$$S_k = \{(W', W^*)|W' = w_i; w_i \in W; W^* = W\}$$ (3.2)
In our example, it would create a set $S$ of four individual sets: one for each word in $W$.

$S_1 = \{(\text{game}, \text{game}), (\text{game}, \text{sport}), (\text{game}, \text{ball}), (\text{game}, \text{team})\}$

with $W' = \{\text{game}\}$ and $W^* = \{\text{game, sport, ball, team}\}$

$S_2 = \{(\text{sport}, \text{game}), (\text{sport}, \text{sport}), (\text{sport}, \text{ball}), (\text{sport}, \text{team})\}$

with $W' = \{\text{sport}\}$ and $W^* = \{\text{game, sport, ball, team}\}$

$S_3 = \{(\text{ball}, \text{game}), (\text{ball}, \text{sport}), (\text{ball}, \text{ball}), (\text{ball}, \text{team})\}$

with $W' = \{\text{ball}\}$ and $W^* = \{\text{game, sport, ball, team}\}$

$S_4 = \{(\text{team}, \text{game}), (\text{team}, \text{sport}), (\text{team}, \text{ball}), (\text{team}, \text{team})\}$

with $W' = \{\text{team}\}$ and $W^* = \{\text{game, sport, ball, team}\}$

where $S = \{S_1, S_2, S_3, S_4\}$.

**Probability Estimation** Most coherence scores use Boolean document calculation defined by the two equations below. However, $C_v$ is calculated with the Boolean sliding window technique. In that case, a window of size $s$ moves over the document one token at a time. At each step, it defines a new virtual document by copying the $s$ tokens in the window. Boolean document calculation is then applied to all of these virtual documents. As a document can touch on several subjects, using the Boolean sliding window technique can help grasp the tokens proximity and refine the topic coherence.

$$p(w_i) = \frac{\text{number of documents containing } w_i}{N}$$ (3.3)

$$p(w_i, w_j) = \frac{\text{number of documents containing both } w_i \text{ and } w_j}{N}$$ (3.4)

In this work, we use the default value of $s = 110$ set by the Gensim library [40]. Because the documents in our example all have less than $s$ words, we can directly apply Boolean document calculations as such:

$$p(\text{game}, \text{game}) = p(\text{game}) = \frac{3}{3} = 1 \quad p(\text{sport}, \text{game}) = \frac{1}{3}$$

$$p(\text{game}, \text{sport}) = \frac{1}{3} \quad p(\text{sport, sport}) = p(\text{sport}) = \frac{1}{3}$$

$$p(\text{game, ball}) = \frac{1}{3} \quad p(\text{sport, ball}) = 0$$

$$p(\text{game, team}) = \frac{1}{3} \quad p(\text{sport, team}) = \frac{1}{3}$$

We compute the probabilities for each pair in $S$ in the same manner. Not all pairs are shown above for the sake of redundancy. We can note that when $w_i = w_j$, then $p(w_i, w_j) = p(w_i)$. 

21
Confirmation Measure As defined by Röder, Both and Hinneburg [41], “A confirmation measure takes a single pair $S_i = (W', W^*)$ of words or word subsets as well as the corresponding probabilities to compute how strong the conditioning word set $W^*$ supports $W'$”. In the case of the $c_v$ measure, we first compute the direct confirmation measure called Normalized Pointwise Mutual information (NPMI) with the following equation:

$$NPMI(w_i, w_j) = \frac{\log \left( \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)} \right)}{-\log(p(w_i, w_j) + \epsilon)}$$

(3.5)

where $\epsilon$ is added to prevent a logarithm of zero. Its default value in the Gensim library is $e^{-12}$, and is also used for our example in this section. Here are the NPMI calculations for each pair in $S_1$ of our example:

$$NPMI(\text{game, game}) = \frac{\log \left( \frac{1+\epsilon}{1} \right)}{-\log(1 + \epsilon)} = -1$$

$$NPMI(\text{game, sport}) = NPMI(\text{game, ball}) = NPMI(\text{game, team})$$

$$= \frac{\log \left( \frac{1+\epsilon}{\frac{1}{3}} \right)}{-\log(\frac{1}{3} + \epsilon)}$$

$$= 2.730758e^{-12}$$

We then define an indirect confirmation measure as follows:

$$\vec{v}(W') = \left\{ \sum_{w_i \in W'} NPMI(w_i, w_j) \right\}_{j=1,\ldots,N}$$

(3.6)

Indeed, indirect confirmation measures estimate the similarity of words in $W'$ and $W^*$ (defined in Equation [3.2]) with respect to the direct confirmation measures of all words in a topic. While direct confirmation methods solely rely on words frequency in documents, indirect confirmation estimations finds how much one word supports another by looking at other words they are commonly associated to. That way, indirect confirmation methods can also ind correlations between words that rarely appear together, but are often found near another group of tokens.

Using Equation [3.6] we define $\vec{x} = \vec{v}(W')$ and $\vec{y} = \vec{v}(W^*)$ to represent $W'$ and $W^*$ as vectors of dimension $N$. When $W'$ or $W^*$ contains more than one token, the vector elements are the sum of the direct confirmations.

In our example, $\vec{x}_{S_1}$ simply is a vector of the NPMI values for each pair in $S_1$. However, each vector element in $\vec{y}$ is the sum of those NPMI values. That is, the first value in $\vec{y}$ is the sum of all NPMI calculations made in $S_1$, the second value is the sum for all NPMI values of $S_2$, etc. Consequently, we only need to compute $\vec{y}$ once, as it is identical for all $S_i$ sets in $S$. 

22
Finally, we calculate the cosine similarity between those vectors to determine how one word is related to the other words in the topic. We use the equation below to calculate that value.

$$
\phi_{S_i}(\vec{x}, \vec{y}) = \frac{\sum_{k=1}^{N} x_k y_k}{\|\vec{x}_{S_i}\|_2 \|\vec{y}\|_2} \quad (3.7)
$$

For our example, the set $S_1$, related to the word “game”, yields a final cosine similarity of about 0.5 as shown below.

$$
\phi_{S_1}(\vec{x}_{S_1}, \vec{y}) = \frac{\vec{x}_{S_1} \cdot \vec{y}}{\|\vec{x}_{S_1}\|_2 \|\vec{y}\|_2} \approx \frac{0.99999}{1 \cdot 2.00946} \approx 0.49764
$$

In the same manner, we also compute the cosine similarities for all sets in $S$ and get the following results.

$$
\phi_{s_2}(x_{S_2}, y) \approx 0.86504, \quad \phi_{s_2}(x_{S_3}, y) \approx -0.85744, \quad \phi_{s_3}(x_{S_4}, y) \approx 0.86504
$$

Notice how $S_2$ and $S_4$, respectively related to “sport” and “team”, obtain the highest scores. This is explain by the fact that they are both in the same document, and also alongside “game”, which puts both of those words in the most central position out of any of the other words in our topic. The token “game” appears in all documents, reducing its chances of being related to any of the other words, as it could mean anything.

Aggregation Finally, we apply the arithmetic mean on all $\phi_{S_i}$ measures to create a single score. This score encompasses the coherence of one topic. For our example, we get $\frac{1}{4} \sum_{i=1}^{4} \phi_{S_i} = 0.34257$. When computing the coherence measure of a LDA model, we aggregate the scores for each topic with the arithmetic mean, giving us one final evaluation of our model. Note that by using the arithmetic mean, topic coherence is sensitive to outliers and a few topics with very high or very low scores can skew the final results.

3.3.3 T-distributed Stochastic Neighbor Embedding (t-SNE)

Van der Maaten and Hinton [33] introduced in 2008 a new dimensionality reduction technique called T-distributed stochastic neighbor embedding (t-SNE). This method creates a representation of data in a low-dimensional space, from data points in a high-dimensional space. The goal is to keep similar data points close together, such that we can visualize high-dimensional data without loosing cluster information. T-SNE is often used to visualize LDA models through the topic probabilities over documents. For example, an LDA model for 5 topics and 10 documents will have a dimensionality of 50, which is impossible for us to visualize. However, using t-SNE, we can plot each 10 documents as a point on a 2-dimension space and use the topic probabilities to determine how close they are.
from each other. This method helps grasp a general idea of how topics are distributed over the documents, if they form clusters or not, and identify similar topics.

T-SNE calculates the similarities between two data points in the high and low-dimensional space and then tries to optimize the two measures. We can describe those three steps as such:

1. **Measure similarities between points in high-dimensional space**: t-SNE starts by centering a Gaussian distribution over a data point $x_i$ to convert the high-dimensional Euclidean distance between $x_i$ and all other points $x_j$ into conditional probabilities. In Van der Maaten and Hinton own words, “the similarity of datapoint $x_j$ to datapoint $x_i$ is the conditional probability, $p_{ji}$, that $x_i$ would pick $x_j$ as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at $x_i$”.

2. **Measure similarities between points in low-dimensional space**: the same method is used for the low-dimensional counterparts $y_i$ and $y_j$ of the high-dimensional points $x_i$ and $x_j$. We denote this conditional probability $q_{ji}$. However, instead of using a Gaussian distribution, the conditional probability $q_{ji}$ is computed with the student’s t-distribution (or Cauchy distribution). The Cauchy distribution has heavier tails to avoid the crowding problem of the regular Stochastic Neighbor Embedding technique. Indeed, creating the conditional probabilities with a normal distribution, like in SNE, tends to congregate data points together in the center.

3. **Optimize similarity measures**: we minimize the sum of the Kullback-Leibler divergences over all data points so that the $q_{ji}$ is as close as possible to $p_{ji}$.
Chapter 4

PyBiblio

In this chapter, we propose a Python library called PyBiblio for bibliometric analysis.

4.1 Overview

We described in the previous section how we created dataframes in Pandas. To study those documents, we created all our analyses from scratch in Python. From the scripts, we developed a free and openly available Python library called pyBiblio. The full documentation is available at [https://romerogroup.github.io/pyBiblio/](https://romerogroup.github.io/pyBiblio/). The library can be installed through both PyPi or Github. It can run bibliometric analyses with any pandas dataframe, but we specifically designed the functions to work with data from the Web of Science database, providing more options depending on WoS tags. Particularly, the functions `cit_by` and `cit_num` are only made possible because any WoS data has a “TC” column, with the number of times a paper has been cited. As of now, pyBiblio only have one class called `Bibliometrics` that holds six functions:

- **TITLE_CLEAN**: Clean title names (remove punctuation, stopwords and numbers).
- **FUNDING_CLEAN**: Clean funding agencies (remove grant number and punctuation) and group similar names.
- **CIT_BY**: Computes number of citations as function of the chosen parameter.
- **CIT_NUM**: Computes number of citations per number of occurrences in the chosen parameter.
- **PUB_BY**: Computes number of publications as function of the chosen parameter.
- **PUB_NUM**: Computes number of publications per number of occurrences in the chosen parameter.

The last four functions all have the following list of possible parameters:
• **df** is a Pandas dataframe.
• **by** is the column name.
• **n** is an integer to select the n rows with the largest values.
• **subset** is a list of items to subset in the chosen column.
• **dpc** is a list of column names in which to remove duplicates.
• **sep** is the separator used in strings of variables (e.g. for China;USA in countries).
• **sort** is a binary variable. If true, the dataframe returned is sorted by decreasing frequency value.
• **norm** is a binary variable. If true, frequency values are normalized by the total number of citations.

To start working with the pyBiblio package, you first have to import the package and create an object of the class Bibliometrics like so:

```python
from pybiblio import Bibliometrics
analysis = Bibliometrics()
```

### 4.2 titleClean

The function `titleClean` is called by either `cit_num` or `pub_num` if the `by` parameter is equal to “TI”, which is the Web of Science tag for titles. However, this function can also be used directly by the user on any column containing string elements, or even on one single string variable. This function only has two parameters: **TI** and **sep**. **TI** is the list of strings to be cleaned. If a string is passed to this parameter instead of a list, it will create a list of length one containing only that single element. **sep** is the separator to be used when joining all cleaned words.

This function first converts all elements in the column to lowercase, and removes URLs, non-alphanumeric characters and digits. Each string is then tokenized, which means that all words are separated to form a list of words. From this list, we remove stopwords given by default in the nltk library and strip the white spaces from the remaining elements. Finally, the list is transformed into a string variable where all words are join together and separated by the **sep** parameter, which is a semi-colon by default. By doing this, the function can return a list of strings in a format similar to other columns of a typical WoS dataframe such as the “AU” column containing the authors like “John Doe; Jane Doe”. In this function, **sep** has a different use than in other functions like `cit_by` or `pub_num`, where it is used to separated the words and not join them. However, the two roles of **sep** are complementary, as preparing the data through `titleClean` will correctly format it to be used by other functions, where **sep** will then be used to separate the words and parse through them.
Below is an example of this function with a string variable. Note how the result is a list containing only one element: a string of all lowercase words separated by \texttt{sep}. We can also notice that stopwords “and”, “in” and “from” are removed.

\begin{verbatim}
title = "Authorship and citation cultural nature in density functional theory from solid state computational packages"
print(analysis.titleClean(title, sep=";"))
\end{verbatim}

\begin{verbatim}
>> ["authorship;citation;computational;cultural;density;functional;nature;packages;
solid;state;theory"]
\end{verbatim}

4.3 \texttt{fundingClean}

The function \texttt{fundingClean} has a similar role to \texttt{titleClean} and has the same parameters. However, it is called by the other functions if the column of interest is “FU”, which is the WoS tag for funding agencies and grants. The other difference is the use of \texttt{sep}, that is both the separator to split the different funding agencies in the original WoS data, as well as to join the cleaned names like in \texttt{titleClean}.

In WoS data, all funding agencies for one paper are combined in one string, separated by semicolon and the grant numbers are in brackets, as shown in the coding example below. After splitting the agencies on the separator, the first thing \texttt{fundingClean} does is deleting the brackets and everything inside. It then removes all non-alphanumeric characters and strips the white spaces. Finally, it groups agencies names based on a file of about 500 of the most recurrent funding agencies in our dataset. Indeed, a lot of agencies are written in different manners. For example, “DOE”, “Department of Energy”, “US Department of Energy” and “US DOE” all represent the same agency. To minimize those difference, we created a csv file that is downloaded with the rest of the library, and that holds common variations of some of the agencies. The function parses an agency’s name through the file and changes its name to the official or more common name instead. Regrouping agencies in this way can help create more precise analyses of funding in the academic world.

\begin{verbatim}
agency = "US DEPARTMENT OF ENERGY [DE-SC0002139]; OFFICE OF SCIENCE OF THE US
DEPARTMENT OF ENERGY [DE-AC02-05CH11231]"
print(analysis.fundingClean(agency, sep=";"))
\end{verbatim}

\begin{verbatim}
>> ["US DEPARTMENT OF ENERGY;US DEPARTMENT OF ENERGY"]
\end{verbatim}

The example above shows how the function \texttt{fundingClean} removes the grant numbers, groups the agencies with their official name, and finally return a list of agencies separated by \texttt{sep}.
4.4 **cit_by and pub_by**

The functions `cit_by` and `pub_by` compute the number of citations or publications (respectively) per instance of parameter `by` (like per year or per author). They return pandas dataframes of two columns: one with the instance, the other with the corresponding number of citations or publications.

A good example for those functions is computing the number of citations per year, which can be achieved as such:

```python
analysis.cit_by(data, by = 'PY', n = 5, norm = True, sort = False)
```

Here, `data` is a pandas dataframe in which “PY” is the WoS tag for the year of publication so the function will only consider this column to compute the number of citations. Null values in the chosen column will be deleted. By setting `n = 5`, the above line of code returns a Pandas dataframe with 5 rows of highest frequency. Frequency values will be normalized over the total number of citations as the binary parameter `norm` is True and rows are sorted by year and not by frequency because the parameter `sort` is False. The function will return a pandas dataframe similar to the one displayed below in Table 4.1, in which the first column is the `by` parameter, and the second column called `freq` returns the number of citations or publications, if the function `pub_by` is used.

<table>
<thead>
<tr>
<th>PY</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>0.0993408</td>
</tr>
<tr>
<td>2008</td>
<td>0.0854208</td>
</tr>
<tr>
<td>2009</td>
<td>0.181217</td>
</tr>
<tr>
<td>2010</td>
<td>0.0727779</td>
</tr>
<tr>
<td>2013</td>
<td>0.0812602</td>
</tr>
</tbody>
</table>

Table 4.1: Results of example for `cit_by`

Note that if the parameter `subset` was not null, the frequency would be different because of the way the normalization is computed. Normalization is calculated by dividing the number of citations or publications in one year by the total number of citations or publications. However, when using `subset`, we remove rows from the dataframe `data` if the value in the `by` column is not in the `subset` list and the number of total citations or publications is only computed from the remaining rows.

4.5 **cit_num and pub_num**

The functions `cit_num` and `pub_num` compute the number of citations or publications (respectively) per number of different occurrences in the column `by` (like number of different authors, countries or words). They return pandas dataframes of two columns: one with the number of occurrences, the other with the corresponding number of citations or publications. These functions cannot be offered for any int or float type column.
Let’s demonstrate the usage of either of those two functions with the following example:

```python
analysis.pub_num(data, by = "AU_CO", n = 10, dpc = ["DI"], subset = ["USA", "FRANCE" ], norm = False, sort = True)
```

As in the previous example, `data` is a pandas dataframe in which “AU_CO” is the column name of the countries where a paper is written. By setting `n = 10`, we choose to only keep the 10 occurrences with the highest frequency. The parameter `dpc` contains a list of the column names in which we choose to remove duplicates. The tag “DI” is a WoS tag for DOI numbers meaning that for this analysis, we want to remove duplicates in the database based on their DOI numbers. Here, the `subset` parameter is set so that only rows with both “USA” and “FRANCE” in their “AU_CO” column are kept. Because the parameter `norm` is False, the function returns the raw values, without normalizing them and rows are sorted by frequency because the parameter `sort` is True. The function will return a pandas dataframe similar to the one displayed below, in which the first column is the number of occurrences, and the second column called `freq` returns the number of publications or citations, if the function `cit_by` is used.

For clarity, Table 4.2 below, shows that, from a dataframe where all papers have different DOI numbers, and were all written from authors in both the USA and France, 442 papers have two countries listed in the authors’ institutions.

<table>
<thead>
<tr>
<th>numAU_CO</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>442</td>
</tr>
<tr>
<td>3</td>
<td>382</td>
</tr>
<tr>
<td>4</td>
<td>225</td>
</tr>
<tr>
<td>5</td>
<td>93</td>
</tr>
<tr>
<td>6</td>
<td>46</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.2: Results of example for PUB_NUM

The parameters `dpc` and `subset` both consider all of their elements, meaning that if two column names are passed in the `dpc` list, only rows with duplicates in both columns are removed. Similarly, only rows containing all of the two or more values in `subset` are kept. This is the reason why the table above does not have any record of papers with only one country of origin: all papers considered had at least two (the USA and France).
In this chapter, we share the results of using the package PyBiblio, described in the previous section, on our dataset of DFT-related academic papers.

5.1 Overview

Figure 5.1.1: Total number of publications per year
Figure 5.1.1 shows the evolution of the research productivity in the field of density functional electronic structure by using the citations obtained from the most important computational packages, specialized mostly on periodic systems. It shows a strong growth in the number of publications with respect to publication year, with more than 14,000 publications in 2019. This demonstrates that computational DFT is a healthy methodology and an active research field. The best fit for this growth is a powerlaw with an exponent of 2.82 (an exponential fit gives a coefficient of 0.13 but with lower $r^2$).

Interestingly, the total number of citations per year (only until 2015) also follows a powerlaw dependence with an exponent of 1.41 as shown in Figure 5.1.2. 2015 is the year with the highest number of citations, and the last peak before a significant drop to 2019. Moreover, 49.14% of papers with less than 5 citations were published in 2018 and 2019 whereas those numbers drop for older papers. Hence, even though around 38.82% of all papers have less than 5 citations it seems that a lot of them are just too recent and will gain more momentum after two years. Hence, as publications increase exponentially, citations seem to follow a similar path, only shifted a few years.
5.2 Author Impact

Figure 5.2.1 shows the number of publications with respect to the number of authors in the publication. A skewed Gaussian like behavior is observed with a maximum at three authors but with a very slow powerlaw decay (with a mean value of 1.62 and standard deviation of 3.83). This slow dependence demonstrates the efforts of the community to produce publications with a large number of authors. In other fields, such as in medical sciences, social sciences and other natural sciences, the number of authors has a maximum between 4-10 authors [31]; however, this is not the case in electronic structure. With this analysis, we can conclude that there is no evidence of hyperauthorship. It is further supported by the fact that only 13 out of the 121 532 papers in our database have 50 or more authors. There is also a good number of single author papers (around 3%), which also supports that performing computational analysis can still be lone wolf research. A very similar skewed Gaussian behavior is observed for the number of citations per number of authors.

![Bar chart](image)

Figure 5.2.1: Total number of publications as function of the number of different authors present in the paper

About 45% of all authors (identified by their ORCID number) in our database have only one publication. This percentage is very high and probably comes from large collaborations or publications made by graduate students or young faculty who are still very new to the domain. A figure of author productivity can be found in Figure 5.2.2. It shows the average number of publications per year, for researchers with at least two publications in our dataset. The highest peaks, in order, are at two, one, and three papers a year.
Figure 5.2.2: Number of researchers as function of their average productivity per year (in terms of publications) for all researchers with at least 2 papers in our database.

Figure 5.2.3: Number of citations over number of publications as function of the number of authors per paper.

Figure 5.2.3 shows the ratio between citations and publications as a function of the number of authors per paper. This plot displays similarities with the number of publications depending on the number of authors, represented in Figure 5.2.1. They both exhibit a high peak at a low number of authors, followed by a rapid decrease. However, for the absolute count of publications, as well as citations, the peak happens at three authors per paper, whereas Figure 5.2.3 reaches its maximum for two authors. It is interesting to see that papers with two and five authors have similar publication counts, but drastically different ratios of number of citations over publications. We also notice a slight growth from five to seven authors, where it then stabilize around 23. This could mean that
even though large collaborations can be complicated, long or hard to maintain, they will pay off as most will have a significant effect on the community. We should note that this figure is sensitive to outliers and a few highly cited papers are significantly raising the peak for two authors per paper. However, the general trend is still valid without them, and papers written by two authors, as well as bigger collaborations, with seven or more authors, are often getting a bigger ratio of citations.

5.3 Country Impact

By looking at the right panel in Figure 5.3.1 displaying the number of different affiliated countries for each citing article, we find that the vast majority of publications are from a single country. Moreover, papers produced in three countries or more are uncommon and represent only about 9.4% of the dataset. Therefore, this field seems to be well geographically localized, while international endeavors probably come from large collaborations such as experimental/theoretical groups or code development teams. This is supported by an exponential fit with a strong decay (with an exponent of 0.64). Similar to the analysis of the number of publications, citation records are larger for papers performed in the same country, although the exponential decay is slower than for the publications case.

Left panel in Figure 5.3.1 shows the number of publications as a function of the countries where the authors are from. Though the vast majority of DFT software packages are developed in Europe (at least those which are highly cited), the larger number of papers come from USA and China by twice as much as Germany, which is ranked as the third country by publication number. Figure 5.3.2 shows a more detailed visualization of each country’s productivity for each of the main DFT packages. Castep and Turbomole are both interesting cases, where one country really distinguishes itself from the others. China cited Castep the most, and by over 3 times as the next countries in list: the USA and Germany. Turbomole is the only code that was the most cited in a different country than the USA or China: Germany, where it originated. ADF and Quantum Espresso are both more popular in the USA and benefit from good support in Europe while VASP and Siesta display a very split number of citations between the USA and China. We need to point out that the records in this plot do not add to the total number of publications considered. Since a publication can have contributions from multiple countries, the publication is counted independently for each country. It is also clear that the USA is the country where the majority of the codes are used and the USA happens to be the country with the largest number of publications. VASP is by far the most used code overall, even if it is not a free package and has similar capabilities to other free licensed packages such as Quantum Espresso.

Figure 5.3.3 reports a log-log plot of the number of papers with respect to the number of citations for a selected set of countries to study the relationship between the two. Each plot shows the 10 countries with the highest H-index [24], for a given code. If $N$ is the total number of papers affiliated to a country, the H-index of that country is defined such that $h$ papers out of $N$ have at least $h$
Figure 5.3.1: Total number of publications per country and per number of countries in a publication. Left: Total number of publications as function of the authors’ countries (Only the top 15 are shown). Right: Total number of publications as function of the number of different countries listed in authors’ institutions.

citations each and the other \((N - h)\) papers have \(\leq h\) citations each. The H-index is an overall good estimate of a country’s impact and corrects for the disproportionate weight of highly cited publications or publications that have not yet been cited. The exact H-index values for the 10 selected countries of each software are available in Figure 5.3.4.

The behavior displayed in Figure 5.3.3 is very similar independently of the DFT package used. Evidently, countries with a large number of publications end up having a larger number of citations. Therefore an almost linear dependence is observed for almost all codes. The USA, China and Germany are often part of the countries with the highest number of both publications and citations. Electronic structure is therefore dominated by countries with large computational facilities and scientific diversity. The cases of Castep, Siesta and VASP are interesting and similar. For Castep, the USA and China both have a similar number of citations, but China achieves it with more than three times the number of American publications. However, Siesta and VASP display the same situation where the USA obtain a lot more citations than China for a similar number of publications. Moreover, China often has a smaller H-index. For example, while China is the third country with the highest number of publications citing ADF, it ranks 10th in H-index. Those profound differences in ranking and number of citations between the two countries demonstrate a better performance for papers from the USA.

To further analyze the dynamics between different countries, we computed the number of publications with at least one country in South America and one in either the USA, Canada or Europe.
(considered as European Union members, Switzerland, United Kingdom and Norway). The same analysis was made for African countries. Although South America and Africa are both developing continents, records show that South America benefits from a lot more collaborations with every other group defined previously than Africa does. However, the highest number of publications are created within two countries in the same class, highlighting how challenging it can be for developing countries to collaborate on scientific research with more economically advanced countries.

In order to address the dynamics of the different collaborations, we now analyze the network properties of country collaboration networks. Basically, we create a network where every affiliated country in the same publication is connected. Country collaboration networks were created and visualized using Bibliometrix [3] and the exact methodology used to build this graph is described in Chapter [3]. Networks were generated from datasets where duplicates and observations with no address (no affiliated country) were removed. The analysis of the network properties is given in Table [5.1]. We added the constraint that at least one of the authors come from the country where
Figure 5.3.3: Total number of publications as function of the total number of citations, on the logarithmic scale. The figures show the 10 countries with the highest H-index for a given code. Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP.

the code is developed. The idea is to study the internationalization of each code and what is the dynamic of the user base. We note that there is an important number of publications from the countries of the main developers of the code, in particular from free license packages (see Figure 5.3.2). Some examples of high publication numbers in developers’ countries include Turbomole (Germany), Castep (United Kingdom), Quantum Espresso (Italy) and Siesta (Spain). This also impacts the community around each one of the packages, as many publications come from personal networking of the code developers. In Figures 5.3.5 and 5.3.6 we present a panel of two figures, for each of six main codes selected. The left hand side reports the network for publications up to 2012 and the right hand side refers to publications from 2013 to 2019. Colors represent clusters: groups of closely related countries.

A first thing to notice about the country collaboration networks after 2012 is the clear cluster created by European countries with smaller publication record like Sweden, Greece or Denmark. This could be explained by incentives given by the European Union for scientific collaboration, or simply by the fact that most of them are neighbors and can have more links, as researchers might have more chances to interact with each other. We can also observe, in this tight cluster, the presence of some non-European countries such as Cuba, Israel or Japan which could indicate a more general
difficulty for smaller research communities to collaborate with more productive and higher cited countries.

Table 5.1: Country collaboration network properties, before and after 2012

<table>
<thead>
<tr>
<th></th>
<th>ADF</th>
<th>Castep</th>
<th>Quantum Espresso</th>
<th>Siesta</th>
<th>Turbomole</th>
<th>VASP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country Collaboration Network Before 2012 (included)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Density</td>
<td>0.191</td>
<td>0.152</td>
<td>0.142</td>
<td>0.229</td>
<td>0.156</td>
<td>0.230</td>
</tr>
<tr>
<td>Diameter</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><em>Transitivity</em></td>
<td>0.361</td>
<td>0.333</td>
<td>0.312</td>
<td>0.390</td>
<td>0.342</td>
<td>0.449</td>
</tr>
<tr>
<td></td>
<td>Country Collaboration Network After 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Density</td>
<td>0.240</td>
<td>0.297</td>
<td>0.277</td>
<td>0.338</td>
<td>0.174</td>
<td>0.398</td>
</tr>
<tr>
<td>Diameter</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><em>Transitivity</em></td>
<td>0.556</td>
<td>0.735</td>
<td>0.705</td>
<td>0.760</td>
<td>0.411</td>
<td>0.742</td>
</tr>
</tbody>
</table>
Figure 5.3.5: Country collaboration networks for papers published before 2012 (included) and after 2012, with at least one author affiliated in the same country as the code. From top to bottom: ADF, Castep, Quantum Espresso.

The density of nodes is much higher for publications after 2012, which suggests that the number of publications increased, agreeing with Figure 5.1.1. Indeed, Castep, Quantum Espresso, Siesta and VASP all have at least a 47% surge in density and transitivity after 2012. The nodes involved before 2012 and after also represent the dynamics of the collaborations. VASP is a code where the connectivity with its country of development is more diffuse and where links between other countries are much stronger, as it has a larger number of publications.
Figure 5.3.6: Country collaboration networks for papers published before 2012 (included) and after 2012, with at least one author affiliated in the same country as the code. From top to bottom: Siesta, Turbomole, VASP.

It is interesting to notice that the density and transitivity double only for Castep and Quantum Espresso. While all other main DFT codes have records before 2003, the first publication citing Quantum Espresso was in 2009. Hence, only about 14% of the citations for Quantum Espresso existed in 2012 and before. This difference in starting year might explain the sudden surge for Quantum Espresso, but it does not apply to Castep. The transitivity increase in Castep may be due to the license change of this package which has happened in the last few years, as the package is now
freely available for academic use. On the other hand, Turbomole did not have as much of a surge in density and transitivity, in fact it is very small (about 12% and 20%), making it the network with the smallest density and transitivity after 2012.

5.4 Journal Impact

Studying the fields that are impacted the most by the use of DFT can offer an analysis over the strong effect of this methodology in the characterization and prediction of materials. This can be obtained by looking at the publication journal. This is what is reported in Figure 5.4.1. From all journals used for publication, Physical Review B is the preferred journal to report analysis and discoveries of density functional theory, followed by Journal of Physical Chemistry C. However, the diversity of journals is very large, demonstrating the use of DFT in a large variety of problems. Almost all editorial companies have journals where DFT calculations are being reported.

![Figure 5.4.1: Total number of publications with respect to the publishing journal (Only the top 15 are shown)](image)

Although the full analysis of all publications provides a clear picture on the behavior of the
electronic structure community, different applications and therefore packages are used in different journals, as can be concluded from Figure 5.4.2. Though Physical Review B is one of the most well known journals in condensed matter physics, ADF citations prefer to publish in solid state chemical journals, as this particular software was created for solid state chemistry applications. Around 20% of published papers from Quantum Espresso are published in journals of the American Physical Society while 18%, 15% and 35% from VASP, Siesta and ADF are published in journals of the American Chemical Society. It is also interesting to notice that the Nature Publishing Group is in the top ten of publishers with the highest number of publications, but when computing the number of citations, moves all the way up to the top 6.

Figure 5.4.2: Total number of publications per journal (Only top 10 are shown). Top row from left to right: ADF, Castep, Quantum Espresso. Bottom row from left to right: Siesta, Turbomole, VASP

Another source of analysis is how likely journals are to publish work on electronic structure calculations. We computed the number of citations normalized by the total number of citations for each one of the considered DFT codes with respect to the journals with the larger number of papers in this field. The normalized numbers are then divided by the impact factor reported in the SJR — SCImago Journal & Country Rank [43] for each journal. Figure 5.4.3 shows the results, for each 6 main codes. The idea is to recover the impact of the paper based on the journal recognition. Castep is the code which has the highest impact in Physical Review B, closely followed by VASP. This can be explained from the fact that these two codes are mostly focused on condensed matter materials, which are very important topics in these journals. On the other hand, Turbomole and ADF have
a larger impact in chemical journals such as Physical Chemistry Chemical Physics and Journal of Chemical Physics. VASP, which is the most used package, is very diverse and the citations are evenly distributed over different journals, which shows the diversity in users of this package. Therefore, this analysis shows that every one of the packages has created a localized community of users with VASP being the most general one, as it is the code with the largest journal diversity.

![Figure 5.4.3](image)

**Figure 5.4.3:** Total number of citations (Normalized) over the journal impact factor defined on the website SJR — SCImago Journal & Country Rank [43]

Figure 5.4.4 shows the normalized total number of citations, with respect to the total number of title words present in each paper. The words were selected following the procedure described in the methodology section. Papers for all 6 main codes reach their maximum number of citations between five and nine words, indicating that shorter titles may be the most successful. A possible explanation is that longer titles do not catch the attention of readers, therefore they are less cited. However, following findings from previous research [14], it is important to note that this figure does not demonstrate causality between the length of the title and the number of times a paper is cited.
Figure 5.4.4: Normalized total number of citations as function of the number of words present in
the titles of the publications

Moreover, the figure displays big disparities between the distributions of each main code, making it
harder to derive any trend.

Figure 5.4.5: Word clouds generated from the titles. Top row from left to right: ADF, Castep,
Quantum Expresso. Bottom row from left to right: Siesta, Turbomole, VASP

An analysis of the twenty most common words in titles for each DFT code shows a common
trend in the presence of theoretical-like words such as “First-Principles”, “Studies”, “Ab Initio” or
“Properties”. These are very general words and characterize many different theoretical publications,
as we can find them in almost all wordclouds displayed in Figure 5.4.5. It is also clear that some
codes have specific applications where users are more focused, for example “Graphene” in Siesta,
effects” in Quantum Espresso and “Transition metal” in VASP. This plot also stresses the idea that communities created through specific topics tend to use one package.

To give weight on the novel ideas, we find that only 1.6% of the whole dataset are defined as review papers (tag "DT" in the Web of Science database), but they represent about 17% of the 500 most cited papers, and 22% of the one hundred most cited papers. This implies that even though most published papers contain new ideas, representing more than 85% of the 500 most highly cited papers, the proportion of reviews increases in the top 100. It could be explained by the fact that reviews offer a great general panorama of a certain field, useful to introduce and explain the added value of a paper.

In the 500 most cited papers, we also noted that the number of pages is not letter like, with an average of 15.25 pages per paper (with a standard deviation of 23.36). Therefore, long papers with important and new field concepts are very well cited by the community.
Chapter 6

Topic Modeling

In this chapter, we discuss the results and derive trends from the Latent Dirichlet Allocation algorithm trained on the abstracts of DFT-related academic papers.

6.1 Topic Distribution

Figure 6.1.1 shows the distribution of documents over topics. This means that it displays how many documents contain each topic. Note, we use the probabilities of a topic appearing in a document as weight, as described in the Methodology section (see Chapter 3). We add up the weights for all documents in each topic, and divide the results by the total number of documents. The figures gives us an idea of how well distributed each model is, and what topic is more common within the documents.

We can see that the model with 25 topics is extremely unbalanced, with only 4 topics making up 80% of the documents. Next, the LDA models for $k = 15$ and $k = 45$ seem to be somewhat well distributed, as no topics is contained in more than 18% and 11% of the documents. This can easily be deduced via the y-axis of the histograms, that are also a quick way to get a general idea of a distribution: if the y-axis range is larger, it means that at least one topic takes a bigger part of the documents, which means the distribution is less uniform. In the same idea, we can study the range of the attributed normalized number of documents. The model with 25 topics has the bigger range at 0.255, followed by 15 topics at 0.175 and 45 topics at 0.105. The smallest range, and hence the model that has a more uniform distribution is for $k = 35$ with 0.036. Indeed, to get 90% of the documents, it takes the first 8 topics (or 53% of the topics) with the highest proportion for $k = 15$, 24% of the topics for $k = 25$, 57% for $k = 35$ and 37% for $k = 45$. Moreover, the model with 35 topics is also the one that has the highest minimum percentage of documents.

The diversity in models for $k = 35$ and $k = 45$ also seem to infer that 15 topics is not enough for our corpus. In fact, we find evidence for topics specialization that further confirm the benefits
of a higher number of topics. For example, in the $k = 15$ model, topic 3 contains the following words: “formation”, “activity”, “hydrogen”, “metal”, “reaction”, “mechanism”, “process”, “catalyst”, “oxidation”, and “li”. We can map this topic to two different topics in the $k = 35$ model: it shares the words “activity”, “metal”, “reaction”, “catalyst”, “oxidation” with topic 6 and the words “reaction”, “formation”, “mechanism”, “process” with topic 27. We can distinctly see that it splits into two categories, one is more focused on specific processes and applications (topic 6) and one is more general (topic 27).

We can conclude from these analyses that the most balanced model is for $k = 35$. This is also the model that obtained the highest $C_v$ score when we analyzed the best number of topics in Figure 3.3.4. For these two reasons, we will mostly focus on this particular LDA model and its results in the rest of this chapter.

The wordclouds for the 35 topics in our LDA model are displayed in Figure 6.1.2. Each wordcloud represents the 10 words with the highest probability in the topic. The size of each word is proportional to their probability within their topics. Bigger words are terms with a higher probabil-
Figure 6.1.2: Top 10 words making up each topic in the LDA model for $k = 35$
ity and a bigger impact in the topic. We also added topic names to give a quick idea of what each wordcloud represents. Even though some topics overlap, most topics are rather independent and clearly defined. For example, in topic 13, we can find words relating to magnetic problems, while topic 7 talks about hydrogen and hydride. Hydrogen is not a magnetic atom so those two subjects should never link, which they do not: they are completely different. Topic 1 on alloy, however, can be related to magnetism as some alloy metals are magnetic and yet, it is once again completely separated to topic 13, which shows how great LDA is at detecting different words relationships. Topic 14 is another good example of terms that could apply to many different fields with words such as “data”, “simulation”, “model” or “approach”, but are associated to each other in a unique topic.

Figure 6.1.3: T-SNE reduction for the LDA model for $k = 35$

We can also analyze the topics of our model by visualizing them with the t-SNE reduction method. Figure [6.1.3] displays the data reduction to 2-dimensions, where each point is a document, positioned depending on their similarity with other documents, and each color is their dominant topic. Note that some topics are not represented as they are not a dominant topic in any document, this is the case for topics 0 and 29 for example. Since the topics are well distributed over the documents, we can see a lot of topics being represented, instead of a few overtaking the entire map. We can also notice that some topics are more dispersed, such as the subject of new methods (31).
That is expected as researchers are developing new processes and data simulations in many different fields. However, other topics are clustered, and clearly defined like the first topic on alloying and crystal formation. This figure also helps visualizing the similarity between topics. The closest two topics are, the more interconnected they are. For example topic 27 is close to topic 6 and topic 3 but is on the opposite side of topics 1 and 13. Indeed, molecular properties and activation processes like catalysis are chemistry related while magnetism and alloying are more often studied in physics.

Figure 6.1.4 compares the distribution of documents over topics, for the 1990-2019 corpus on the left (that was already shown in Figure 6.1.1), and for the 2020 corpus on the right. The two plots are similar and both display a good distribution of the documents. While topic 10 and topic 14 were the most common subjects up to 2019, a number of other topics caught up to them in 2020, including topic 31 that now has the highest number of assigned documents. This could also mean that while topics 8 and 10 were very important up to 2019, and still are, topics 4, 24 and 31 are gaining more momentum recently. The similarity between the two plots also indicate that our model is generalizing correctly and can efficiently attribute topics to new documents.

6.2 Topics over time

We computed, for each topic independently, the normalized and weighted number of documents per year. Every year, there is an exponential amount of DFT-related articles being published, and consequently, almost all topics also grew exponentially. To compare them, we calculated the powerlaw factor of each curve since 2000. Even though some topics’ evolution are better fitted with an exponential curve, the powerlaw function yields the highest mean $r^2$ over all topics. The results are available in Figure 6.2.1 where the number in each top left corner of the subplots are the power factor best fitting for each topic. We chose to only consider publications after 2000 as this is the
A total of nine topics have a power factor lower than 1 and while they still grew in importance since 2000, a few of them are starting to stabilize and even lose some interest from the DFT community. For example, topic 18 on DFT schemes and topic 19 on nanotubes, which have the lowest factors recorded, have not gained any new momentum in the recent years and topic 18 also observed a decrease of publications in 2017 for the first time ever. On the contrary, topics 24 related to electrochemical processes and lithium batteries, as well as topics 4, 6 and 31 are growing at a significantly high rate and seem to be the most important subjects in the field of DFT today. Other subjects are all developing at intermediate rates, representing the diversity of DFT and how important the methodology is for different topics.

We report in Figure 6.2.2 the two topics with the smallest and highest growth out of the 35 in our model. We fit a powerlaw and an exponential curve to both of the curves and reported the factors in the top left corner. Topic 24 is an example of a growth better fitted with an exponential function, getting a $r^2$ of 0.998, even though a powerlaw function also get a high $r^2$ score of 0.988.

Figure 6.2.1: Normalized and weighted number of documents over the years, per topic for $k = 35$ period in which DFT software really started to get globally used.
On top of the important difference in growth, we can also notice how “bumpy” the evolution of topic 18 is. The many irregularities, compared to the smooth curve of topic 24 might also be a sign of interest lost.

Figure 6.2.3 also shows the growth of topics over the years, in the form of a heatmap. We notice that no important increase happened before 2000, which confirms our choice to study the exponential factors only after 2000. Moreover, most topics only start gaining momentum after at least 2005. It is interesting to note that the three topics with the highest exponential factor (topic 4, 6 and 24) grew extremely fast in a few years, which is the reason for the high factor. Topic 24 especially only started to differentiate itself from others in 2018. It is also interesting how out of those three topics, two of them (number 6 and 24) are not part of the topics with the highest proportion of documents. However, given their high power factor, we can only predict that they will become more important in the next few years. Similarly, by looking at the trajectory of some subjects such as topics 1 or 17, we can confidently forecast that the interest in those subfields will grow in the coming years. The 2020 column in this figure seems to agree with those predictions as topics 1, 4, 6, 17 and 24 all gained in proportion during the last year. On the other hand, we can also easily determine which topics are stabilizing and are loosing momentum. For example, topic 3 and topic 18 have had interest from the scientific community for at least a decade and yet, are not growing as fast as others.

It is interesting to compare the evolution of topics 14 and topic 31, as they are related to each other. Topic 14 contains words such as “data”, “model” and “simulation” which represent the community’s effort to incorporate code, Machine Learning and new computing methods into their work. This topic has been growing steadily over the years, as it becomes a bigger part of today’s experiments. Secondly, topic 31 relating to the improvement and optimization of code and processes, only started gaining momentum in 2014, about 10 years after topic 14. However, its growth was faster, which places both of the topics at the top of the most published subjects in 2020 in DFT. We can also mention the evolution of topic 4 related to graphene and monolayers. It started to grow...
a few years after 2010, which is the year that the Nobel Prize in physics was attributed to Andre Geim and Kostya Novoselov for their work regarding the two-dimensional material graphene.

We also study the evolution of topics, by countries, over the years. Figure 6.2.4 divides the heatmaps of topics per country over four different time periods. Before 2005, the USA were dominating the field of DFT and published the highest number of publications in a larger number of topics. Then, the proportions slowly shift to China that now produces more articles, and diversified its interests. China started to get involved in DFT after 2005 through the subject of crystal phase transitions (topic 10). Pressure crystal phase transitions is the main strength of DFT and one of the most active subfield, which explains why it would be the first topic to attract interest. Topic 10 stayed an important part of Chinese work until now, as it was still one of the main published subjects by China between 2015 and 2019, along with topics 4 and 12. Nowadays, the USA still contributes to many different topics in DFT, and grew interest in topic 14, but decreased its number of publications in topic 5 (adsorption/diffusion).

We can accurately predict that China will soon dominate the field of DFT as its number of publications will only keep growing. Figure 6.2.5 representing the normalized distribution of documents

![Topic distribution over years, $k = 35$](image)

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
per topics over countries, for the year 2020 confirms the influence of China in DFT. Moreover, even though most software counted in this work are made in Europe, they are the most used in China and the USA. Germany, the third country with the highest number of publications, and country of origin to Turbomole, the sixth most used code in our dataset is barely even competing with China and the USA after 2005. Similarly, the impact of the United Kingdom and France kept decreasing over the years, even though they were very active in several different topics before 2005.

6.3 Topics and Journals

Figure 6.3.1 shows the normalized number of publications (computed with the probabilities as weight) per topic, over journals. The 15 journals displayed have the highest number of publica-
Figure 6.2.5: Normalized and weighted number of documents per country (only top 15 shown), per topic, for articles published in 2020.

Even though some journals like Surface Science are specializing in one topic, we can notice how most journals are present in several different subjects. The two most important journals, Physical Review B and Journal of Physical Chemistry C are both influential in DFT and were expected to be the largest contributors to the field. Moreover, as DFT is mostly used in physics and chemistry, it is not surprising that those two journals both focused on condensed matter analysis each represent one of the two domains.

In fact, the three journals in positions 2, 3 and 4 are related to the chemistry side of DFT. Even though they have similar targets, they evenly distribute the publications between them. The Journal of Chemical Physics focuses on topic 14 and 18, Journal of Physical Chemistry C contributes more to topics 5, 6, 8 and a few others while Physical Chemistry Chemical Physics has a more diverse range of subjects. Hence, it seems that the Journal of Chemical Physics contributes more to methodologies and not material characteristics, judging by the words present in topics 14 and 18 (modeling/approximations). Even though physics researchers mostly publish their articles in one
particular journal, chemists have more options available depending on their interests.

Figure 6.3.2 is similar to Figure 6.3.1 but the x-axis shows publishers (and not journals). The other difference is that we created this plot with only articles published in open-access journals. It is interesting to see how the two figures have different interests. For example, open-access journals seem to have a strong interest in topic 31 while subscription journals in Figure 6.3.1 only published a small amount of their articles relating to this subject. The other three most important topics in open-access journals are topics 4, 8 and 10 which are also well published in paid journals but to different levels. While topics 4 and 8 grew in proportion, topics 10, 13 and 14 all saw their share
Figure 6.3.2: Normalized and weighted number of documents per topic and per publisher (only top 10 shown), with only open access journals considered, for $k = 35$

of documents decrease in open-access publishers, even though they still remain relevant. Moreover, it seems that open-access journals are more focused on a few topics, instead of spreading to many different subjects, like paid journals. We can also notice the rather good correlation with the evolution of topics over time as the most published papers in open-access journals are also the ones that are the most published in 2020 (see Figure 6.2.3).
Chapter 7

Conclusions

Bibliometrics and topic modeling are powerful tools to grasp the impact of any given field. In this thesis, we developed several analyses over the authors, countries, journals, publishers, institutions and subjects from over 120,000 papers in Density Functional Theory. This methodology, even though well known since the end of the 20th century, is still an important and influential field of science today, thanks to its implementation as software.

The development and the impact of electronic structure computational packages is revealed by the large number of publications per year, which is the result of the efforts of a wide range of applications from many different material science related fields. At the same time, computational electronic structure is a field where a small number of groups work on developing ideas, as the number of participant countries and number of authors per paper is small in comparison with other research fields.

Developing an electronic structure package takes an important effort from developers and it needs a strong relationship between the different developer groups. Network analysis for each DFT code shows that during the first years of code development, the interaction happens mostly between the country where most of the development is performed and some other countries, which are directly related to the code developers. After some years and with the code reaching maturity (enough debugging by users and developers), the networks become much more dense and connected. It is interesting to see the degree of interaction between different continents and even geographical zones. Independently of the code, east European countries tend to collaborate more with other countries that have a smaller interaction density. We also notice that larger user communities surround codes which have simple interfaces, which increases the number of publications and citations. Therefore, efforts along the lines of making the codes more user friendly and with simpler interfaces are valued by the user community.

Moreover, two of the most recurrent topics in our analysis, and that receive the most interest from influential journals and countries are related to code development, simulations and modeling.
As our work focuses on computational packages, it is not surprising that some papers focus on the methods and processes of software. However, the large amount of interest in the subjects suggests a recent and strong effort in the physics and chemistry fields to develop and optimize code packages and libraries.

Even though most packages are developed in Europe, they are mostly used in the USA and China, that are dominating the amount of publications created in the DFT field. While the two countries share interests in a lot of different subjects, American scientists work with many of the most important DFT packages, and Chinese users use mostly VASP, which is a licensed package. This difference might come from the intrinsic set of tools each software has, and the subfields they serve the most. Indeed, China seems to direct most of its work to materials science and applications as they focus on 2D materials (topic 4), chemical reactions (topic 6) or batteries (topic 24) while the USA published most of its work in simulations (topic 14) and new methodologies (topic 31).

The topic modeling done in this thesis was created using the abstracts of papers and already resulted in coherent and clear topics. Future work could include replicating our methodology on the full-text of all 120,000 papers in our dataset. We started taking steps in that direction and worked towards the development of an interface reading the PDF version of research papers, and extracting the main sections as well as important information. By doing so, we hope to simplify researchers’ work by efficiently highlighting the main information of any paper, as well as providing general information about the state of a given field. As we did previously, all code will be freely available on GitHub and added as features to our Python library pyBiblio, to help anyone create bibliometric analyses and topic modeling of research papers.
Publications

Parts of this work, including but not limited to the bibliometric analysis in chapter 5 has already been published in *Scientometrics*, see:

Dumaz, M., Boucher, R., Marques, M.A.L. et al. Authorship and citation cultural nature in Density Functional Theory from solid state computational packages. 
https://doi.org/10.1007/s11192-021-04057-z

The topic modeling aspect of this thesis is also in the process of being published.
Bibliography


Eugene Garfield. “Citation analysis as a tool in journal evaluation”. In: Science 178.4060 (1972), pp. 471–479.


Appendix A

Software References

Table A1: References used to collect citations for each code.

<table>
<thead>
<tr>
<th>Software</th>
<th>References</th>
</tr>
</thead>
</table>
• Te Velde, G t et al. (2001). “Chemistry with ADF”. In: Journal of Computational Chemistry 22.9, pp. 931–967. |


<table>
<thead>
<tr>
<th>Conquest</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Bowler, DR, R Choudhury, et al. (2006). “Recent progress with large-scale ab initio calculations: the CONQUEST code”. In: physica status solidi (b) 243.5, pp. 989–1000.</td>
</tr>
</tbody>
</table>


• Laino, Teodoro et al. (2006). “An efficient linear-scaling electrostatic coupling for treating periodic boundary conditions in QM/MM simulations”. In: Journal of chemical theory and computation 2.5, pp. 1370–1378


<table>
<thead>
<tr>
<th>Crystal</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DACAPO and ASE</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Empire</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>References</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| FHI-aims | • Blum, Volker et al. (2009). “Ab initio molecular simulations with numeric atom-centered orbitals”. In: Computer Physics Communications 180.11, pp. 2175–2196.  
<table>
<thead>
<tr>
<th>FreeON</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>GPAW</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Enkovaara, J e et al. (2010). “Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method”. In: Journal of Physics: Condensed Matter 22.25, p. 253202.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JDFTx</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWChem</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Octopus</td>
</tr>
<tr>
<td>Onetep</td>
</tr>
<tr>
<td>PySCF</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td>-----------</td>
</tr>
</tbody>
</table>