Universiteit Leiden

ICT in Business

Linking Science and Technology:
Reference Matching for Co-citation Network Analysis

Name: Shuo Yang
Date: 19/08/2016

1st supervisor: Dr. Frank Takes
2nd supervisor: Dr. Hendrik Jan Hoogeboom
External supervisor: Jos Winnink (CWTS)

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The patents underlying inventions offer an important indication of technology development. Citations of scientific publications in patent documents build links between patents and scientific publications. These links provide us with a way to analyze the interaction between science and technology. We match 27 million literature references from the *European Patent Office (EPO) Worldwide Patent Statistical (PATSTAT)* database with 45 million scientific publications from the *Web of Science (WoS)*. In this thesis, we present an approach to link patents literature references to publications in three steps: identify publication attributes in literature references, select matching candidates and apply approximate string matching algorithms to refine the match candidates. The matching results can be used as data source for studies on the interaction between science and technology. More specifically, we use social network analysis techniques to study the interaction between scientific disciplines in patents.

Acknowledgments

This thesis is a product of one and a half year part time work as IT student assistant and half a year of concentrated research. The process of writing this Master's thesis was wonderful and insightful yet sometimes stressful. Here, I would like to take the opportunity to express my sincere gratitude to the people who gave me guidance, support and assistance when writing the thesis.

First, I would like to thank my first supervisor Dr. Frank Takes. He helped shape my ideas into clear research objectives and structure my thesis into a coherent whole. His detailed comments and precision in academic writing continue to impress me. I would also like to thank my second supervisor Dr. Hendrik Jan Hoogeboom for his excellent suggestions and for reviewing my thesis.

Second, I would like to thank my external supervisor Jos Winnink, who introduced this topic to me and gave me a lot of useful advice throughout the entire process. With his help and patience, I gained a lot of inspiration for the methodology used. I would like to thank Professor Robert Tijssen, Dr. Alfredo Yegros Yegros and my other colleagues, who welcomed me so warmly and made working at CWTS one of the best experiences during my Master's study in the Netherlands.

Finally, I would like to thank my parents for their love and their support. I also thank my friends for the happy times and for distracting me from stress, and my boyfriend for the great company and for helping me review. Without the guidance and help from those amazing people, my life could not be this happy and wonderful.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, we will explain our research topic and introduce our research questions in Section 1.1. In Section 1.2, we will present an overview of the thesis.

1.1 Research topic and questions

In our present time, knowledge and innovation are valuable and intellectual properties are lucrative because they not only bring market competence and business values to companies but also contribute to the welfare of society. As an important type of intellectual property, patents represent a vast source of information covering every field of technology and offer a safe way to protect inventions and innovations [World Intellectual Property Organization, 2015]. Apart from their commercial value and legal uses, patent information is also used to operationalize the complex concept of 'technology', which forms the basis of this thesis.

Science, on the other hand, is the pursuit and application of knowledge and understanding of the natural and social world following a systematic methodology based on evidence [The Science Council, 2009]. Progress in science is communicated by researchers to the rest of the world primarily through scholarly publications, and as such the collective publication output of researchers is a representation of the scientific system.

Science and technology closely interact with each other. Science contributes to technology development by acting as a source of new technological ideas, engineering design tools, instrumentation

and analytic methods. In return, technology contributes new scientific challenges and measurement techniques to science [Brooks, 1994]. In this thesis, we aim at developing an approach to automatically identify publications mentioned in patents' literature references. The data obtained is useful to support studies on the interaction between science and technology with patents and scientific publications as proxies.

In patent documents, inventors, applicants or examiners use references to relate their novel technological claims to previous science and technology, as a means of contextualizing their specific technological contribution. These references include citations to other patents but also to non-patent literature like papers, abstracts, conference proceedings, books, databases and many others. In this thesis, we refer to them as *literature references*. They limit the scope of the inventor's claim to novelty and in principle, present a way to distinguish the novel claim of other patents and a link to the source of the information and existing knowledge the inventor used or referred to [Brusoni et al., 2005]. While patents' references to other patents show the technological context of an invention, literature references reveal the other types of knowledge that were used to either come to the invention or to contextualize the invention. In the case when these literature references point to academic literature, they indicate a linkage between the technological invention and the source scientific knowledge.

Researchers in the *Centre for Science and Technology Studies (CWTS)* in the Netherlands have worked on linking patents' literature references with scientific publications and they had matching results already. However, there are quite a lot of missed match in the results. So, in this thesis, we use a different method to link literature references with publications, aiming at improving the current algorithms used by CWTS and obtaining more reliable and complete links between references and scientific publications. Furthermore, with the matching results, we intend to build a publication co-citation network and perform a modest demonstrative study on the interaction between the science disciplines in inventions. According to the objectives, we ask the following research questions:

1. *How can we automatically identify which scientfic publication is cited in patents' literature references?*

(a) *Can we identify the part of literature references which do not point to scientfic publications?*

(b) *Which combination of algorithms allows for the discovery of reliable links between literature references in the PATSTAT database and publications in the WoS database?*

2. *Can we identify patterns of interaction of scientific disciplines using the publication co-citation network?*

1.2 Thesis outline

We present the data used in this thesis in Chapter 2 prior to other parts, as the methodology used is highly dependent on the data format and data quality. Preliminary observations about this data can be found in Chapter 3. In Chapter 4, we further elaborate on the problems and challenges in matching patents' literature references with scientific publications. After understanding the problems, we present the methodology in Chapter 5. Chapter 6 introduces the process of matching literature references with scientific publications and the implementation of the matching method and gives answers to the first research questions. In Chapter 7, we present the empirical results from the implementation of the matching approaches. It includes an evaluation of the matching algorithms and matching results. Chapter 8 introduces a way to utilize the matching results for specialized studies, where we build a publication co-citation network to study the interaction between various disciplines in inventions. It answers the second research question. In Chapter 9, we conclude the problems and effectiveness of the matching algorithms. The conclusions we obtain in this thesis and future work we can do to improve the research are described in Chapter 9.

2 Data

From the *European Patent Office (EPO) Worldwide Patent Statistical (PATSTAT)* database, we collect patents' literature references, which are specially separated from the patent literature references. On the other hand, the *Web of Science (WoS)* database provides us with over 45 million scientific publications. Both of these databases are described in the following two subsections.

2.1 The PATSTAT database

PATSTAT is published by the *European Patent Office (EPO)* and is updated every half year. It contains bibliographical and legal patent data relating to more than 90 million patent documents. PATSTAT offers bibliographic patent data and metadata, for instance, data on patent families, inventors & applicants, publications, citations and so on. Figure 2.1 shows the domain model of the PATSTAT database.

In this thesis, we only look into the references to non-patent literature, which are originally from the EPO's master bibliographic database DOCDB, also known as the EPO Patent Information Resource. The reference-publication relation is shown in Figure 2.2.

Publication applicants publish literature references for two reasons; one is to distinguish their patents from other patents by citing related patents directly; the second reason is to disclose the source of the information used in their patents, often times citing scientific literature or other types of non-patent literature [Guner, 2015].

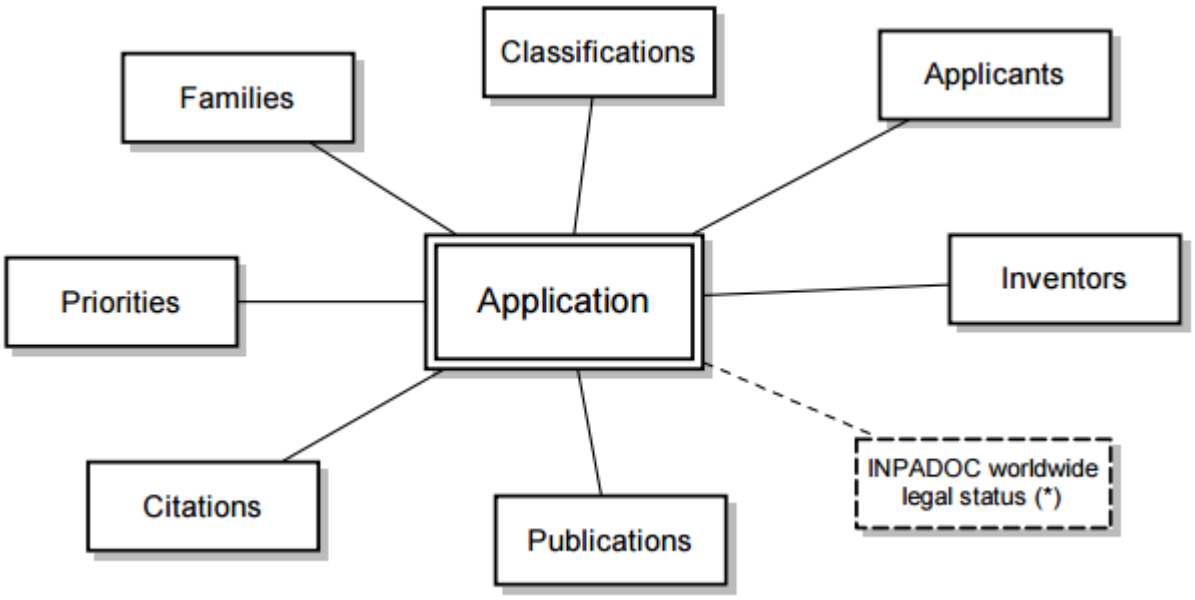

Figure 2.1: The domain model of the PATSTAT database in Microsoft SQL Server 2012 [European Patent Office, 2014]

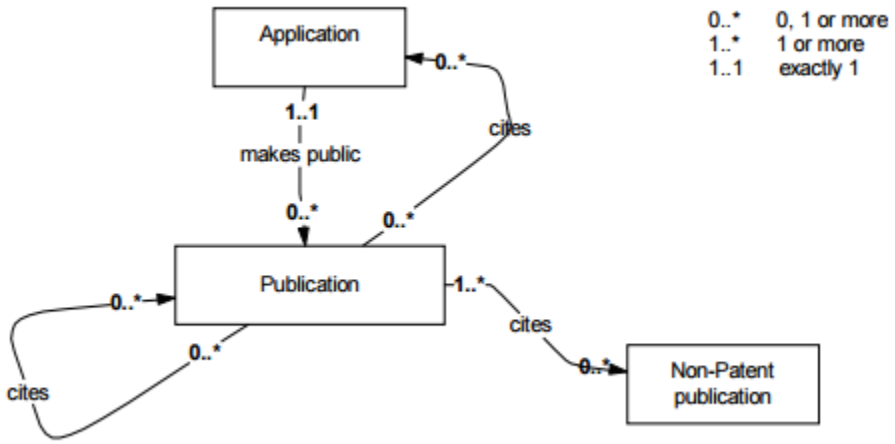

Figure 2.2: The reference-publication relation[European Patent Office, 2014]

Those two types of reference sources are stored separately in the PATSTAT database. The patent references point to patent applications or publications available in PATSTAT. However, the entities that the non-patent literature references point to are not classified within PATSTAT. As a result,

while PATSTAT is well-structured when it comes to presenting citations to other patent applications or publications, it does not provide well-organized and processed non-patent literature references for extensive and efficient use [Guner, 2015]. Within the PATSTAT database, such references are called *non-patent literature* references. For purposes of readability, we will simply refer to these as *literature references* or *literature* in the rest of this thesis. In Table 2.1, the first two columns give examples about how these literature references are stored in PATSTAT, while the third column (literature type) is not provided in the original PATSTAT database but is a classification made by researchers in CWTS in the Netherlands, which will be explained further in Section 6.1.2.

The literature references are in the form of single text strings and are recorded without making changes to what applicants provide. From the literature reference examples in Table 2.1, it is noted that the literature references are not limited to references to scientific publications but point to documents in various types. For instance, the references may point to books, abstract collections, legal documents or EPO search reports. All these references to other document types are, for our purposes, noisy data which should not be matched to scientific publications. The various documents cited as literature references do not have unique identifiers or follow standard formats. From the first two examples in Table 2.1, it is observed that usually literature references contain one or several of the following specific information terms which are referred as *publication attributes* [European Patent Office, 2014]:

1. Author(s);
2. Title of the publication;
3. Abstract;
4. Publication year;
5. ECLA Classification;
6. ISBN, ISSN or *Digital Object Identifier (DOI)*.

These specific pieces of information can (partially) link literature references to publications. In this thesis, they are the bridge which leads us to identify the cited scientific publications. We will elaborate on this notion in Section 6.1.3.

Literature ID	Literature reference text	Literature type
957964736	Pande, H., et al., Proc. Nat l. Acad. Sci. , USA, vol. 81, No. 15, 1984 pp. 4965 4969.	scientific publication
957964742	IPlotkin, S.A. et al. Protective Effects of Towne Cytomegalovirus Vaccine Against Low Passage Cytomegalovirus Administered as a Challenge. J. Infect. Dis., vol. 159, No. 5, May 1989.	scientific publication
957964697	See also references of EP 1166601A1	patent publication
957964808	PATENT ABSTRACTS OF JAPAN vol. 010, no. 363 (E-461), 5 December 1986 (1986-12-05) & JP 61 158672 A (FUJI ELECTRIC CO LTD), 18 July 1986 (1986-07-18)	patent publication
957964800	GenBank Accession No. N42635, Hillier et al., Jan. 25, 1996.	database
957964797	Strategene Catalog, p. 39, 1988.	miscelaneous publication
957965743	3G TS 33.102 version 3.0.0-Draft Standard, 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security Architecture (May 1999).	international standard
957966620	AGP Tutorial: Chapter 4-AGP Memory Mapping (visited Dec. 27, 1999) http:developer.intel.comtechnologyagp tutorialchapt_4.htm, 2 pages.	web page
957971168	DATABASE WPI Derwent Publications Ltd., London, GB; AN 1993-10414713 & JP 5 043 404 A (RIKEN VITAMIN CO) 23 February 1993	Derwent abstract
957967982	Aida, Chem Abs 86, 43746 (1976).	Chemical abstract

Table 2.1: Example of literature references in PATSTAT

2.2 The Web of Science database

Web of Science (WoS) is a bibliographic database, covering data from multiple scientific disciplines: science, social science, arts and humanities. The document types that are covered in the WoS

database are [Thomson Reuters, 2013]:

1. Article
2. Bibliography
3. Biographical Item
4. Book Review
5. Correction
6. Database Review
7. Editorial Material
8. Hardware Review
9. Letter
10. Meeting Abstract
11. News Item
12. Reprint
13. Review
14. Software Review

The full record of the scientific publications in the WoS database includes several pieces of information that may be useful to our linking, like:

1. Title
2. Author(s)
3. Source (source name, volume number, issue name, page(s), publication year etc.)
4. Link to full text and/or library holdings information

The WoS database was accessed via CWTS. CWTS not only owns a local copy of the WoS database, they also maintain an offline and enhanced copy of the database which they use for their own research purposes. The original database is referred to as the WOSDB database, while the enhanced is internally known as the the *WoS knowledge database (WOSKB)*. The WOSKB presents data is in a more flexible entity-relation data structure. It aggregates a part of the data and presents connected data from the perspective of the data functionality in the research work of the institute. It is constructed by linking publication information stored in separate tables by the

primary keys in the WOSDB database. The scientific publications can be uniquely identified by 'UT', the source item identifier. The examples of research data we use from WOSKB are represent in Table 2.2.

UT	000003907500005
title	Unusual mitochondrial DNA polymorphism in two local populations of blue tit Parus caeruleus
first author	TABERLET, P
other authors	MEYER, A; BOUVET, J
source	MOLECULAR ECOLOGY
year	1992
volume	1
issue	1
page	2736

Table 2.2: Example publication record from the WoS

3 Preliminaries

In this chapter, we will introduce previous scientific work on the topic of interpreting literature references to link technology to science and citation matching. We present relevant concepts and insights from these previous contributions. Following this, we address how our work differs from the previous work and how it makes a novel contribution to this scientific research field.

3.1 Interpretation of literature references

The literature references in patents are conceptually different from citations in academic publications, because the purpose of references in patents is to establish their novelty and as such they are strategically selected to further the applicant's legal interests, while references in academic publications serve an academic purpose.

Over the past a few years, extensive emphasis has been put on the interpretation of non-patent literature references. Some studies use literature references to reflect direct influence of science on technology [Narin and Noma, 1985]. However, some findings from detailed patent case studies questions using the literature references as an indication of a direct link or influence from science on technology [Meyer, 2001]. Studies from *Tijssen et al. (2000)* and *Verbeek (2002)* raise similar doubts about using patent citation data to trace the so-called "knowledge flow" [Tijssen et al., 2000; Verbeek et al., 2002]. *Tijssen et al. (2001)* suggests using non-patent literature references as a general indicator of interaction between science and technology [Tijssen, 2001].

However, as the literature references are selected strategically for the legal concerns by applicants or examiners, researchers raise their doubt about what these literature references actually represent, especially whether they really include the knowledge base of the inventions. The non-patent literature references in patents usually have two sources: some are from the inventors or applicants and are enclosed on the 'front-page' of a patent; others are chosen by the examiners as part of the legal examination of the patent, and may include all, part or none of the citations originally chosen by the inventors or applicants. *Tijssen et al. (2000)* performs a survey on *United States Patent and Trademark Office (USPTO)* patents for validating study about the meaning of literature references. The results indicate that 35% of all examiner-given references are the same as those given by the inventors and confirm that 94% of the research covered by literature is relevant to the knowledge presented in the patent, and that most literature references are science-based.

3.2 Citation matching

The reference-publication matching is essentially a problem of linking a citation to a specific paper. In the very early stage of citation studies, citation matching was done by people manually. As the number of scientific publications increased and citations as indicator for many research and academic fields became more important and widely used, more efficient and intelligent ways for citation matching were required. Meanwhile, the advances in computing power opened the gate for automatic citation matching. *Citeseer* was created as one of the first search engines for academic papers [Giles et al., 1998]. It automatically generated a citation index that could be used for literature search and evaluation. These days, major bibliographic databases like *Web of Science*, *Google Scholar* and *Scopus* offer information on partial citation networks and citation indexes which are generated by citation matching within their own databases. In addition, several *application programming interfaces (APIs)* for citation matching are available to facilitate relevant development. But most of these APIs only work well for a limited number of citation formats and for the content in certain databases. For example, *CrossRef Metadata Search API* matches citations with the publication records in the CrossRef's database [Lammey, 2014]. The fact that literature references in patents are not structured in a consistent way but follow numerous different citation formats makes the citation parsing and matching complicated. A parser has to recognize multiple formats, or else

it will be useful only for a very limited number of literature references.

3.3 Research contribution

The patents' reference-publication link is more sophisticated than matching references in academic publications to other academic papers because patents and scientific publications are from fundamentally different communities, necessitating cross-community matching. The patents' literature references are much less structured than publication citations for two reasons, one is because it contains a wide range of literature and scientific publications are just part of them; the other is that the literature references do not follow any consistent formats. So far, there is no specific literature reference parser for patents on the market. On the other hand, the existing citation matching tools like citation indexing databases and citation searching engines cannot handle a variety of patents' literature reference structures and are furthermore limited by the publication pool against which they can match. Our research develops advanced parsing and matching methods and uses the WoS as the matching target to mitigate these limitations.

We present a way to use approximate string matching methods to assist in reference-publication matching. Our work contributes an improved string alignment method compared to the ones currently in use. This method is designed to perform string alignment in a broader context and to be more flexible, in order to overcome hurdles which were problematic for previous string matching methods used in reference-publication matching.

In the context of science and technology studies, our research opens the gate for further macro-level studies on science and technology. The extensive reference-publication matching results that we provide cross the communities of science and technology and will allow researchers to start focusing on system-level patterns of science and technology interaction, while previously research was limited to smaller levels that did not represent either the complete patent literature or the complete scientific literature. The publication community detection performed on the co-citation network shows the way scientific disciplines interact in technology development.

4 Problem statement

PATSTAT and WoS data are from different publishers and are not aligned through mutual standards. This creates difficulties when trying to match entities from these two databases. This thesis focuses on how to build a bridge to overcome the inconsistencies of this cross-community matching. In addition, the fact that literature references are not classified within PATSTAT and are extracted from what applicants submit without processing and validation makes the link more complicated. The difficulties of finding reference-publication matches are:

1. Large volume of the data. PATSTAT includes around 27 million literature references while WoS has around 45 million scientific publications.
2. Literature references are in the form of not-very-well structured text strings.
   (a) Data entry errors like typographical errors in text strings and incorrect volume numbers, issue numbers or publication years.
   (b) Incomplete citation information in literature references.
   (c) One literature reference string may contain multiple entries.
   (d) Duplicated records occur in the literature references.
3. Literature reference cover a wide range of literature types. Only part of them point to specific scientific publications.
4. A literature reference may contain a reference to a patent document.
5. Literature references do not always point to scientific publications directly.
   (a) A literature reference may contain a citation to another patent document's reference list, for instance "See also references of WO 03064220A1"
   (b) A literature reference may simply be a URL, pointing to a PDF copy of a scientific

publication or a web page of an online database of scientific publications. These documents often cannot be matched to publications by only looking into the textual content of the literature reference.

6. Literature references can be written in three official EPO languages: English, German or French. Most records are in English. Publication data from WoS is only available in English.
7. Even if a literature reference points to a scientific publication, it does not mean the publication is covered in CWTS's version of the WoS.

Confronting these challenges, this master's thesis is aimed at developing an automatic solution to identify which scientific publications are cited in patents' literature references to a high degree of accuracy. The additional challenges caused by the poor structure of patents' literature references make this cross-community matching more complicated than 'normal' matching of references in scientific publications to their cited publications. Our work introduces a methodology specially for mapping patents' literature references to scientific publications, bridging the 'communities' of patent data and publication data.

We divide the reference-publication matching method into two phases, as is suggested to be common practice by [Fedoryszak et al., 2013]: *segmentation* and *entity resolution*. At the segmentation phase, the citations are parsed into individual components which fundamentally resemble the metadata of publications, like author name, title, volume number and so on. We look for and use patterns of patents' literature references according to the features of the literature references and use regular expressions to extract the separate components within these references.

During the entity resolution phase, the aim is to search the WoS for the parsed metadata extracted from the patents' literature references to determine whether these literature references point to scientific literature in our database and if so, which publication it matches. To match the components in literature references that resemble publication metadata, we combine several algorithms of approximate string matching with enhancements to specifically deal with the features of patents' literature references.

Looking back at the research questions mentioned in Chapter 1, it is clear that in order to define a concrete approach, more detailed sub-goals are required. Therefore, we propose the following sub-goals:

1. Develop an approach to extract publication metadata in literature references.
2. Develop algorithms for matching metadata in the format of text strings.
3. Establish a set of criteria based on which we can evaluate the matching quality.
4. Validate the matching results.
5. Build a model to analyze the matching results.
6. Build a publication co-citation network using matching results
7. Identify the patterns of discipline interaction from the publication co-citation network.

The sub-goals reflect the key problems we expect to encounter when building a publication co-citation network. We will gain more insights in the interaction between science and technology if we are able to achieve these sub-goals.

5 Methodology

In this chapter, we will elaborate the methods used in the process of reference-publication matching. In Section 5.1, we will introduce the methods which are widely used in string comparison and string alignment. Based on the essentials of string matching methods and features of literature references, we come up with a new string alignment method, *Improved Fitting Alignment*, which is described in Section 5.2.

5.1 Approximate string matching

The string similarity between a literature reference string and a publication helps to understand whether the reference-publication pair is matched or mismatched. The string similarity can be measured based on characters, or based on words. In this thesis, we use the number of character operations required to transfer one string to the other as the measure. The operations allowed under this scheme include character insertion, character deletion and character substitution. In this section, we describe string alignment and comparison methods based on the character operations. In Section 5.1.1, we will present the basic concepts of approximate string matching methods. We describe three classical string alignment methods for string similarity calculation in Section 5.1.2, Section 5.1.3 and Section 5.1.4. Furthermore, we introduce methods to compute the *longest common substring (LCS)* and the *edit distance* in Section 5.1.5 and Section 5.1.6, respectively.

5.1.1 Preliminaries

String alignment is a widely used metric to identify the similarity of two strings. Functionally, it finds the least cost to transform a string into another string globally or locally. We define two strings X and Y:

$$X = x_1x_2x_3...x_m$$

$$Y = y_1y_2y_3...y_n$$

For $X = \{\texttt{vermiform}\}$, $Y = \{\texttt{formation}\}$, the possible global and local alignments are as shown in Figure 5.1.

```
vermiform   vermiform-----
::||:::::         ||||
formation   -----formation
```

Figure 5.1: String alignment

In the example shown in Figure 5.1, some characters in one string are replaced by characters in the other string in the first alignment. This is the character substitution operation. Different from the first one, the dashes in the second alignment reflect gaps, which demonstrate that in order to align the two strings, one or more characters have been deleted from one string. Alternatively one could also say that there is an insertion of a gap in the other string. The character substitution, deletion and insertion are the basic character operations for string transformation. In this thesis, we focus on using dynamic programming techniques to perform string alignment. The alignment goal is to find the optimal string alignment with the least cost character operations, which is reflected by the *alignment scores*. The scoring system consists of four parts: designing a basic scoring scheme, constructing a similarity matrix, calculating the alignment score and finding the best string alignment.

Basic scoring scheme To find the optimal string alignment, we will have to decide, for instance, whether we should substitute characters or introduce gaps, how many gaps we can introduce and

the place to put the gaps. These decisions can have an impact on the optimal alignment that the algorithms find, and depending on the result we want, these may be more or less desirable. So, we score the character operations to reflect their impacts.

The basic step of aligning two strings is to compare two characters from two strings. When two characters are the same, they will be given a match score. When the two characters are different, we could substitute one character with the other one, in which case the pair incurs the mismatch penalty to reflect the cost of character substitution. We could also use character insertion or deletion to align them with gaps, which will result in a gap penalty. Introducing gaps with penalties ensures that alignment does not become meaningless, as the action of adding a gap must increase the overall score of the other aligned characters by more than the negative effect of the penalty. This simple condition puts a limit on the usage of gaps. The value of gap penalties is a parameter which can be changed during the alignment, thus controlling the number and positions of the gaps.

The values of the match score, the mismatch penalty and the gap penalty are parameters. They are be set depending on the goals of the string alignment. Here, we give the main definitions that we will use during further explanation of string alignment.

- X_i: the partial substring consisting of the first i characters of string X
- Y_j: the partial substring consisting of the first j characters of string Y
- $X[i]$: the i^{th} character of string X
- $Y[j]$: the j^{th} character of string Y
- $S(a,b)$: the score for aligning single character a and character b, incurring a mismatch penalty when character a and b are different and a match score when they are identical
- $SIM(i,j)$: the alignment score of string X_i and string Y_j, reflecting the string similarity

Similarity matrix We compute the alignment score of two strings in a recursive way by finding the optimal alignment for substrings starting from the beginning of the strings. The similarity matrix is used to present the process and optimal alignment. We take string $X = \{\texttt{frame}\}$ and string

$Y = \{\texttt{form}\}$ as example. First, we construct an empty matrix whose rows amount to $m+1$, with m representing the length of string X, and columns amount to $n+1$, with n representing the length of string Y. Then, we initialize the first row and the first column. We fill in the matrix with different values, depending on the purpose of the alignment. Local alignment, global alignment, semi-global alignment, edit distance and LCS all have different requirements for the way we initialize the similarity matrix. Figure 5.2 corresponds to the scenario when we compute the global alignment when the match score is set to 1 and both mismatch and gap penalties are -1. In the following sections, we will elaborate on how to initialize the matrix for the various purposes accordingly .

We compute the score for empty cells in the matrix from existing scores from the cells to their left, top or top-left (diagonal). The value in the cell in the i^{th} row and the j^{th} column ($2 \leq i \leq m+1, 2 \leq j \leq n+1$) corresponds to $SIM(i,j)$. When we get this score from the top ($SIM(i-1,j)$) or the left $SIM(i,j-1)$, it represents a gap in the string alignment. In this case, we get this score by adding gap penalty to the existing score from the top or the left. When we calculate scores from the diagonal ($SIM(i-1,j-1)$), this represents the two characters are matched or aligned with substitution through adding the proper $S(X[i],Y[j])$ score. The best score of these three operations is selected to fill in this cell. The filled matrix is shown in Figure 5.2. The pseudocode used to compute the global alignment score of optimal alignment follows in Algorithm 1 in Section 5.1.2.

Finding the optimal alignment The approach to identify where the best alignment score in the matrix lies depends on the specific alignment purpose that we choose. For instance, to find the best global alignment score, we should take the score in the cell at the last row and the last column as the answer. Then, we start at this cell and trace back, returning to the cell of the matrix that this value was derived from. We repeat this until we trace back all the way to the starting position. In Figure 5.2, the arrows show the direction. The cells on this path give the optimal alignment. In this example, the best alignment is:

```
for-m-
|:|:|:
f-rame
```

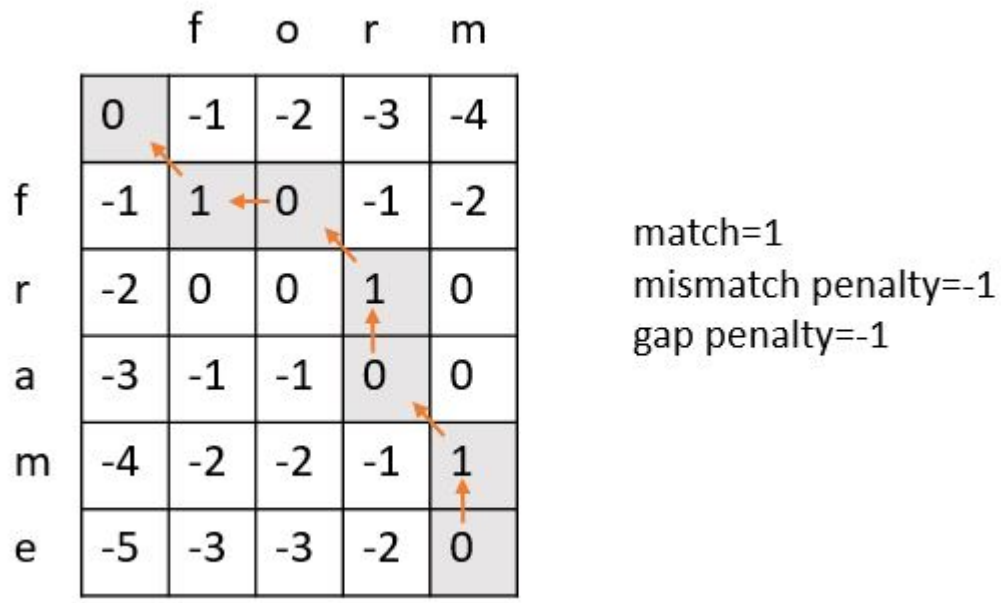

Figure 5.2: Similarity matrix with trace back

It is possible that there are more than one best alignments, when there are multiple optimal 'paths' that return from the final cell to the starting cell.

5.1.2 Global alignment

Global alignment is aimed at finding the best end-to-end alignment of both of the two strings and is suitable when the two strings are of similar length. The **Needleman-Wunsch algorithm** [Needleman and Wunsch, 1970] is a widely used method for global sequence alignment in the context of molecular biology. It computes the similarity of two strings recursively by starting with computing the shorter prefixes and storing alignment scores in the similarity matrix and reusing them for longer prefixes. This optimal global alignment can be found by recursively computing the similarity score in the similarity matrix with the formulas below.

$$SIM(i,0) = i \times g \qquad 0 \leq i \leq m$$

$$SIM(0,j) = j \times g \qquad 0 \leq j \leq n$$

$$SIM(i,j) = max \begin{cases} SIM(i-1,j-1) + s(x_i, y_j) & \\ SIM(i-1,j) + g & 1 \leq i \leq m, 1 \leq j \leq n \\ SIM(i,j-1) + g & \end{cases}$$

Based on the recursive calculation of the optimal alignment for each character, we are able to get the optimal global alignment. As we want to transform one string to the other from the beginning to the end, the similarity score for the global alignment is at the final position of the similarity matrix, namely $SIM(m,n)$. From this final position, we trace back along the direction over which the maximal score is obtained. The cells on this back-trace path constitute the final optimal alignment. The process to only compute alignment score without trace back is explained by the pseudocode in Algorithm 1.

Algorithm 1 Global alignment

```
1: procedure GLOBAL_SIMILARITY(X, Y)          ▷ X = x1x2x3...xm, Y = y1y2y3...yn
2:     for i = 0, 1, ..., m do
3:         SIM[i, 0] ← i × g
4:     for j = 1, 2, ..., n do
5:         SIM[0, j] ← j × g
6:     for i = 1, 2, ..., m do
7:         for j = 1, 2, ..., n do
8:             SIM[i, j] ← max(
                               SIM[i − 1, j − 1] + S(X[i], Y[j]),
                               SIM[i − 1, j] + g,
                               SIM[i, j − 1] + g
                               )
9:     return SIM[m, n]
```

Algorithm complexity We use dynamic programming techniques for the implementation of the global string alignment algorithm. For string X and string Y whose lengths are m, n respectively, the algorithm requires $O(mn)$ time. In terms of space, if the final alignment of the strings is

required, we need $O(mn)$ space, as a single matrix of size $(m+1)(n+1)$ is needed. If only the alignment score of two strings is desired, we can reduce the memory to $O(\min(m,n))$ by using **Hirschberg's algorithm** [Hirschberg, 1975]. It is described as a divide and conquer version of the NeedlemanWunsch algorithm and compute the optimal alignment score by only storing the current and previous row of the similarity matrix used in Needleman-Wunsch algorithm.

This algorithm complexity not only holds for global alignment, but also for the implementation of the local alignment, semi-global alignment, finding the longest common substring and the edit distance using dynamic programming techniques. This is because they all resemble the global alignment in the sense that they all compare two strings character by character and fill in a same-sized similarity matrix with alignment scores.

5.1.3 Local alignment

Local alignment is useful when we are only interested in identifying the locally similar regions between two strings instead of comparing strings end-to-end. The **Smith-Waterman algorithm** [Smith and Waterman, 1981] is widely used in local alignment. The mismatches at the beginning and the end carry lower cost than those in the middle of strings. As the prefixes and suffixes in the strings can be ignored, the Smith-Waterman algorithm offers a solution to see whether a substring of one string aligns well with a substring of the other. In addition, the negative values of similarity do not have any meaning in local alignment because negative matching substrings can be removed from the local match. So, if a negative value is obtained, we reset it to zero. The implementation of the local alignment using dynamics programming is quite similar to the global alignment. In Algorithm 2, the difference from the global alignment is highlighted in red.

5.1.4 Semi-global alignment

Semi-global alignment is useful when one string is significantly shorter than the other one. It can be used to identify whether a string is a part of the other one as well as find the overlaps between strings. It can find similarities that global alignments fail to find. The major difference of semi-global alignment from global alignment is that gaps at the beginning or the end of at least one

Algorithm 2 Local alignment

```
1: procedure LOCAL_SIMILARITY(X, Y)                    ▷ X = x1x2x3...xm, Y = y1y2y3...yn
2:     S ← 0
3:     for i = 0, 1, ..., m do
4:         SIM[i, 0] ← 0
5:     for j = 1, 2, ..., n do
6:         SIM[0, j] ← 0
7:     for i = 1, 2, ..., m do
8:         for j = 1, 2, ..., n do
9:             SIM[i, j] ← max(
                                 SIM[i − 1, j − 1] + S(X[i], Y[j]),
                                 SIM[i − 1, j] + g,
                                 SIM[i, j − 1] + g,
                                 0
                                 )
10:            S ← max(S, SIM[i, j])
11:    return S
```

of the strings are ignored. There are two types of semi-global alignment, which are explained in Figure 5.3, again using strings X and Y as examples.

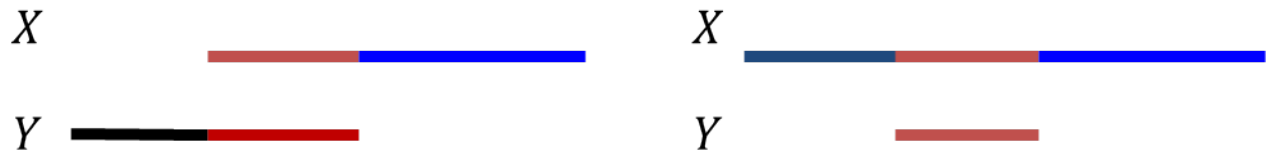

Figure 5.3: Semi-global alignment

The left image shows the scenario when we are interested in identifying the specific overlap between two strings while the right one indicates that the shorter string should be a part of the other, longer string. These two scenarios are distinguished by the gap penalties chosen for each. In the

Case	Change
Ignore gaps in the beginning of string X	Initialization: $S[i,0] = 0 (0 \leq i \leq m)$
Ignore gaps in the end of X	Traceback from: $\max(SIM[i,n])$, $(0 < i \leq m)$
Ignore gaps in the beginning of string Y	Initialization: $S[0,j] = 0 (0 \leq j \leq n)$
Ignore gaps in the end of string Y	Traceback from: $\max(SIM[m,j])$, $(0 < j \leq n)$

Table 5.1: The literature reference classification

left case, the gaps at the beginning of string X and the end of string Y should not be penalized. In the right case, the gaps at the beginning and the end of the longer string X should be free. Again, the scoring scheme plays a very important role to decide the alignment as a different one can yield a different result. Consequently, we need to design our scoring scheme carefully to meet our alignment goals.

The implementation of semi-global alignment is similar to the global alignment. The recurrence remains the same. But the way in which we initialize the similarity matrix and the method for finding the final optimal similarity score are different, as gaps at the beginning or the end of one string are ignored. Table 5.1 shows the changes from the global alignment when we implement the semi-global alignment for different cases.

In this thesis, we aim at fitting the shorter sting Y into the longer string X. Then we need to ignore the gaps in the beginning and the end of string X, the corresponding algorithm is shown in Algorithm 3. The lines highlighted in red shows the difference the global alignment and the semi-global alignment.

If we want to know the overlap between string X and string Y when we can ignore the prefix of string X and the suffix of string Y, then the best alignment similarity score is $\max(SIM[m,j])$.

Algorithm 3 Semi-global alignment

procedure SEMI_GLOBAL_SIMILARITY(X, Y) ▷ $X = x_1x_2x_3...x_m, Y = y_1y_2y_3...y_n$

  for $i = 0, 1, ..., m$ **do** ▷ assuming $n < m$ and we fit Y into X

    $SIM[i, 0] \leftarrow 0$

  for $j = 1, 2, ..., n$ **do**

    $SIM[0, j] \leftarrow j \times g$

  for $i = 1, 2, ..., m$ **do**

    for $j = 1, 2, ..., n$ **do**

      $SIM[i, j] \leftarrow \max($

$$SIM[i-1, j-1] + S(X[i], Y[j]),$$
$$SIM[i-1, j] + g,$$
$$SIM[i, j-1] + g$$
$$)$$

  return $\max(SIM[i, n])$, $(0 < i \leq m)$

5.1.5 Longest common substring

In this section, we aim at finding the longest string(s) that is a substring/substrings of two or more than two strings. We define string Z as one of the substrings of both string X and string Y. String Z is the longest common substring (LCS) between X and Y if there does not exist a LCS of both X and Y. There can be multiple LCSs of two given strings, but the length of the LCS is unique. The LCSs is like the local alignment with "infinite" gap and mismatch penalty and match score set as 1. We define $L(i, j)$ to represent the length of the LCS(s) of string X_i and string Y_j. Algorithm 4 shows the dynamic programming implementations for the LCS algorithm.

From the pseudocode in Algorithm 4, we are able to obtain the length of the LCS(s) of two strings. In addition, in the similarity matrix, the cell(s) with the maximal score, namely the length obtained, indicate(s) the end of the LCS(s). Tracing back along the diagonal(s) to the cell whose similarity score is 1, we can obtain the LCS(s).

Algorithm 4 Longest common substring

```
1:  procedure LCS(X, Y)                                   ▷ X = x1x2x3...xm, Y = y1y2y3...yn
2:      length ← 0
3:      for i = 0, 1, ..., m do
4:          L[i, 0] ← 0
5:      for j = 1, 2, ..., n do
6:          L[0, j] ← 0
7:      for i = 1, 2, ..., m do
8:          for j = 1, 2, ..., n do
9:              if X[i] = Y[i] then
10:                 L[i, j] ← L[i − 1, j − 1] + 1
11:             else
12:                 L[i, j] ← 0
13:             if L[i, j] > length then
14:                 length ← L[i, j]
15:     return length
```

5.1.6 Edit distance

The edit distance between two strings is the minimum number of character operations to transform one string into the other. It is also referred to **Levenshtein distance** [Levenshtein, 1966]. The number of character operations in the edit distance represents the dissimilarity of two strings. In this sense, the edit distance is opposite to the concept of score in the global alignment, as the former is an expression of dissimilarity while the latter expresses similarity. To compute the edit distance, we refer to the way we compute the global alignment but with the alteration of setting the match cost to 0 and a mismatch or a gap cost to 1. In addition, instead of obtaining the maximal similarity score, we need to obtain the minimal distance between two strings, and the final position then represents the edit distance. We define $DIS(X,Y)$ to represent the edit distance between string X and string Y.

Algorithm 5 Edit distance

```
1:  procedure DISTANCE(X, Y)                          ▷ X = x1x2x3...xm, Y = y1y2y3...yn
2:      g ← 0
3:      for i = 0, 1, ..., m do
4:          DIS[i, 0] ← i × g
5:      for j = 1, 2, ..., n do
6:          DIS[0, j] ← j × g
7:      for i = 1, 2, ..., m do
8:          for j = 1, 2, ..., n do
9:              if X[i] = Y[i] then
10:                 s ← 0
11:             else
12:                 s ← 1
13:             DIS[i, j] ← min(
                                DIS[i − 1, j − 1] + s,
                                DIS[i − 1, j] + g,
                                DIS[i, j − 1] + g
                                )
14:     return DIS[m, n]
```

Algorithm 5 presents computing edit distance using dynamic programming approach. As the edit distance indicates the dissimilarity of two strings, we compute the string similarity defined as $Similarity(X,Y)$ between string X and string Y in the range of $[0,1]$ as the formula defined below:

$$Similarity(X,Y) = 1 - \frac{DIS(X,Y)}{\max(m,n)}$$

5.2 An alternative: Improved fitting alignment

The results of the semi-global string alignment methods heavily depend on the alignment goal: fitting or overlap. They can yield different similarity scores and of course different alignments. As stated in Chapter 4, the literature references are in an ill-structured format and demand various

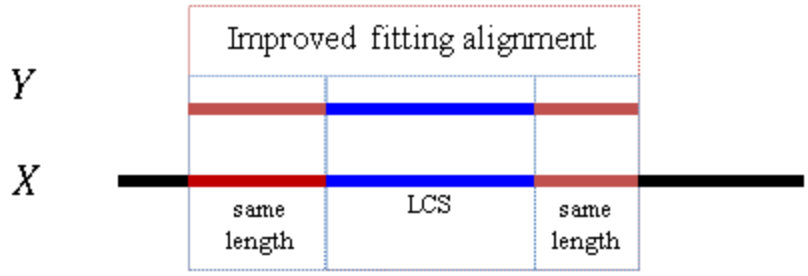

Figure 5.4: Improved fitting alignment

alignment strategies. In an ideal case, a publication title from WoS is part of the literature references that point to this publication. This implies that the best alignment method to use in this case is the semi-global alignment between two strings when one is part of the other. However, patent applicants or inventors may use incomplete titles in the literature references. Accordingly, the semi-global alignment algorithms for finding overlap between two strings should be chosen to solve this problem. Those two scenarios pose different demands about the scoring scheme for the semi-global string alignment algorithms. If we only take the first scenario into consideration when we design a scoring scheme, potential good matches will be overlooked. Conversely, only considering finding overlaps results in incorrect matches. As a consequence, it is difficult to find one specific scoring scheme which can be used to match literature references with publication titles generically. To mitigate this limitation and balance the false positives and the false negatives, we introduce *Improved Fitting Alignment (IFA)* based on the LCS and edit distance.

To explain the improved fitting alignment, we take string X and string Y. We assume that string Y is shorter than string X and we want to fit Y into X, by which we mean we aim to figure out whether string Y is part of string X. The fitting alignment is shown in Figure 5.4.

The improved fitting alignment algorithm consists of three major steps: finding the LCS between two strings, truncating the longer string to the same length of the shorter one, and computing the edit distance of the shorter string and the truncated longer string sequentially. If string Y is a part of string X, in principle, the LCS should indicate the approximate position of Y occurring in string X. In addition, the part in string X where Y is fitted should be of a similar length of string Y. Based

on the LCS and the length of string Y, we can extract the aligning part in string X.

For string X and string Y, the LCS starts from the i^{th} and the j^{th} position respectively and we define the length of the LCS as $k(0 \leq k \leq n \leq m)$. When length k is zero, it means that X and Y do not have any part in common, so we conclude that string Y is not part of string X. When length k is not zero, then we know:

$$LCS = x_i x_{i+1} \ldots x_{i+k-1} = y_j y_{j+1} \ldots y_{j+k-1}$$

$$(1 \leq i \leq i+k-1 \leq m, 1 \leq j \leq j+k-1 \leq n)$$

After finding the LCS, we compute how many characters occur before or after the LCS in both X and Y. We refer to these as the *prefix length* and *suffix length*. The lengths of the prefixes before the LCS in X and Y are represented by L_{xp} and L_{yp} respectively. Correspondingly, the lengths of the suffixes after the LCS in string X and Y are represented by L_{xs} and L_{ys} respectively. Then:

$$L_{xp} = i - 1$$

$$L_{yp} = j - 1$$

$$L_{xs} = m - k - i + 1$$

$$L_{ys} = n - k - j + 1$$

To obtain the alignment which is a substring of string X and is of the same length as string Y and contains the LCS, we first extract the substring directly preceding the LCS, consisting of the same number of characters as the prefix before the LCS in string Y. For instance, there are L_{yp} characters before the LCS in string Y, so we extract the substring in the length of L_{yp} containing from the $(i - L_{yp})^{th}$ character to the $(i-1)^{th}$ character in string X. If the starting index $(i - L_{yp})$ is out of the boundary, we set the starting index to 1, as shown in Figure 5.5.

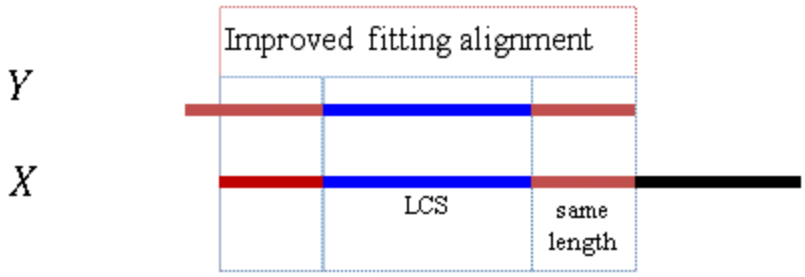

Figure 5.5: Improved fitting alignment

Second, from string X, we get the substring directly following the LCS, of the same length of the suffix in string Y. It means we extract the substring containing the $(i+k)^{th}$ character to the $(L_{ys}+i+k-1)^{th}$ character. If the ending index is out of boundary, we set it as m, the length of string X, and it means the substring extraction stops once the last character in string X is included as shown in Figure 5.6.

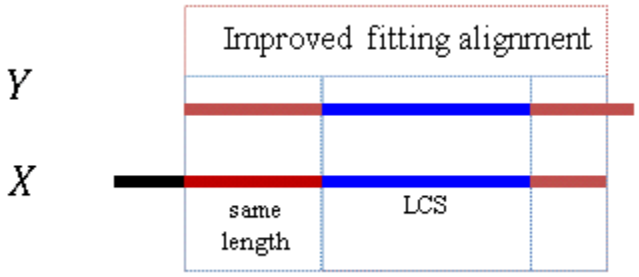

Figure 5.6: Improved fitting alignment

Finally, we concatenate the newly extracted prefix, the LCS and the newly extracted suffix. This concatenation is the part in string X where Y is fitted, which we represent as $\overline{X}$. We conclude that the position of string $\overline{X}$ in string X ranges from $\max(1, i-L_{yp})$ to $\min((L_{ys}+i+k-1, m)$. The longest possible length of string $\overline{X}$ is n.

After obtaining the improved fitting alignment, we score the alignment using edit distance. We

compute the edit distance between the improved fitting alignment $\overline{X}$ and string Y. We normalize the edit distance to the interval $[0,1]$ by dividing by n, the length of the shorter string Y. We use $FIT\_SIM(X,Y)$ to represent the similarity between X and Y when we fit Y into X. As the edit distance represents the dissimilarity, we score the similarity as:

$$FIT\_SIM(X,Y) = 1 - \frac{DIS(\overline{X},Y)}{n}$$

.

There are some particular cases we need to address. If there is no LCS of string X and string Y, then we stop after obtaining no LCS and the $FIT\_SIM(X,Y)$ is 0. If there are multiple LCSs of string X and string Y, then for each LCSs, we align strings and compute the edit distance. The minimal edit distance is considered the one that reflects the dissimilarity.

It is noted that, if the alignment between string $\overline{X}$ and string Y contains the LCS, then, we can skip LCS when computing the edit distance. In this case, we only need to compute edit distance between the prefixes of string $\overline{X}$ and Y and suffixes of both strings as well. We sum these edit distances up and obtain the total edit distance between string $\overline{X}$ and Y. But, in order not to miss out any better alignment between string $\overline{X}$ and Y with less edit distance, we will keep the LSC when computing the distance.

The time or space complexity of the improved fitting alignment algorithm consists of two major components of algorithm complexity: computing LCS and edit distance. The time complexity of finding LCS is $O(mn)$ because it needs to compare each character in string X with each character in string Y. The space complexity for finding the LCS is $O(mn)$, because matrix of size $(m+1)(n+1)$ needs to be stored for tracing back. Computing the edit distance between string $\overline{X}$ and string Y takes $O(n^2)$ time. We do not need to keep the alignment in memory, so we can use $O(n)$ space to compute the edit distance using Hirschberg's algorithm. So, overall, space complexity of the improved fitting algorithm is $O(mn)$. As computing edit distance is subsequent to finding the LCS, the time complexity is $O((m+n)n)$.

6 Implementation

In this master thesis, we aim at linking the literature references in the PATSTAT database with the scientific papers in the WoS database. As these two databases are unrelated as they originate from different sources and are maintained by different publishers, it is necessary to interpret and mutate data from one source to be in concordance with the other, before we can proceed with finding matches between the two sets of data. This entire procedure can be summed up in the following phases:

1. Data preparation
   (a) String cleaning
   (b) Document type cleaning
   (c) Reference parsing
2. Matching
   (a) Match candidate selection
   (b) Match candidate refinement
      i. Publication title matching
      ii. Author name matching
      iii. Publication source name matching

In Section 6.1, we will introduce our procedure for harmonizing literature references and retrieving publication metadata from patents' literature references. The selection of a set of candidate matches and subsequent refinement of this selection are described in Section 6.2.

6.1 Data Preparation

The poor structure of literature reference strings and the massive volume of the two databases are the primary challenges to overcome in properly linking patents' literature references from PATSTAT to publications in WoS. As a result, we perform data preparation as the first step of the procedure. Its aim is to bring the two data sources more in line, which facilitates future computing such as string pattern extraction, text matching and similarity calculation. The results and performance of this task are highly related to the data quality of literature reference strings and publications in WoS. Consequently, we perform string cleaning on references and publication metadata in the format of strings to ensure higher data quality, which is described in Section 6.1.1. We also clean the document types for both databases as described in Section 6.1.2, discarding unhelpful documents to reduce the massive data volume and decrease the computing work.

6.1.1 String cleaning

The goal of string cleaning is to make the text string more concise without changing the essential contents. As the text strings will be used to calculate the edit distance and similarity scores of strings, it is necessary to eliminate factors that hinder the comparison of strings. We need to remove string features that do not reflect the textual content of the strings, as these will just make the text extraction and comparison more difficult and inaccurate. Here, the literature reference strings in the WoS need to be cleaned.

The first case is that white space at the beginning or the end of the text strings are considered unnecessary. Removing them does not make any changes to the content, so they should be removed. The second case is that multiple consecutive space characters in the middle of strings can be replaced by one single space, as the multiple consecutive space characters result from transcription errors or typographical errors and do not convey special meaning. In the two cases above, processing white space does not introduce any modifications of the content of the text strings and it helps when computing edit distance and doing pattern extraction.

In addition, the entire string of publications' author names and source names in WoS are rendered in capital letters, while patent applicants or inventors usually capitalize only the first letter of words when citing them. The capitalization of literature references is not standardized among inventors or applicants. To harmonize capitalization practices between the two databases (which aids string matching, extraction and similarity calculation), all text strings in these two databases are processed to be in the form of lowercase characters. Table 6.1 shows an example of literature reference cleaning and Table 6.2 gives an example of cleaning the text records of WoS.

reference string	Wieman et al., Laser-frequency stabilization using mode interference from a reflecting reference interferometer,Opt. Lett., vol. 7, No. 10, pp. 480-482 (1982) Figure 2, abstract, p. 482 col. 1, par. 4.
cleaned reference string	wieman et al., laser-frequency stabilization using mode interference from a reflecting reference interferometer, opt. lett., vol. 7, no. 10, pp. 480-482 (1982) figure 2, abstract, p. 482 col. 1, par. 4.

Table 6.1: Example of literature reference string cleaning

item	uncleaned	cleand
title	LASER-FREQUENCY STABILIZATION USING MODE INTERFERENCE FROM A REFLECTING REFERENCE INTERFEROMETER	laser-frequency stabilization using mode interference from a reflecting reference interferometer
first author	WIEMAN, CE	wieman, ce
other authors	GILBERT, SL	gilbert, sl
source	OPTICS LETTERS	optics letters
abbreviated source	OPT LETT	opt lett

Table 6.2: Example of WoS record cleaning

6.1.2 Document type cleaning

The massive data volume complicates computing. PATSTAT includes around 24 million literature references while WoS has around 42 million scientific publications. $24M \times 42M$ potential matches

may be returned in the most complicated situation when we need to compare each literature reference with entries of each publication. This is too much to process with reasonable equipment. So, we perform document type cleaning to reduce the amount of computations we need to perform later on in the thesis.

As mentioned in Section 2.1, various types of documents are cited as literature references. Only part of the literature references point to scientific publications while others are considered noise. So in this section, we clean the document types by identifying and removing the records which do not point to scientific publications. A thorough classification scheme created by the researchers at CWTS is used here. The classification of reference categories is especially used to distinguish noisy data from those literature references that have a higher chance to point to scientific publications. The categories are identified as CWTS researchers go through the data and look for patterns or keywords in literature references representative in each class and sort the literature references accordingly. Table 6.3 presents an overview of the classification for 27,201,124 literature references in PATSTAT.

Considering the publication coverage of WoS, literature references belonging to the category 'article' have a high chance to point to publications in WoS. So only references in the category 'article' including 44.89% of all the references are used in this thesis.

In addition to building a smaller and manageable data set of literature references in PATSTAT, shaping WoS data to a smaller data set further facilitates the research. As patents describe inventions that solve technological problems or are technological products or processes, the documents that they cite as references are often more or less technology related. As a consequence, we only include publications from a technology-related discipline in the experiment data set.

The discipline categories of WoS chosen here have been developed by CWTS for the Science and Technology Indicators 2010 report of the Netherlands Observatory of Science and Technology [Netherlands Observatory of Science and Technology NOWT, 2001]. These NOWT categories are a tiered grouping system of WoS subject categories. The NOWT has three level of categories

NPL type id	NPL type	number of references	percentage
0	no references	452,748	1.66%
1	article	12,210,458	44.89%
2	proceedings	1,312,177	4.82%
3	(hand)book	573,891	2.11%
4	tech notes	79,171	0.29%
5	database reference	292,840	1.08%
6	Chem Abs abstract	316,110	1.16%
7	ref to patent pub	5,704,072	20.97%
8	Ref to Intl. Standard	102,174	0.37%
9	Adm. action	189,166	0.70%
10	Medline abstract	1,859	0.01%
11	Miscellaneous publication	842,864	3.10%
12	Nature preliminary	34,175	0.13%
13	Science preliminary	20,962	0.08%
14	not yet classified	5,068,457	18.63%

Table 6.3: The literature reference classification

in terms of broadness. At the most fine-grained level, these consist of 33 disciplinary subject categories, as well as one category for multidisciplinary journals such as “Nature” and “Science”, and one category named “social sciences, interdisciplinary”. The intermediate level contains 14 categories by grouping categories further. At the broadest level, publications are grouped into 7 categories. We use the categories at the intermediate level in this research. Based on these categories, the 45,279,947 publications in total in WoS is chopped into two parts: a technology related part and a part that is not related to technology, as shown in Table 6.4.

The records of WoS publications in categories 1, 3, 5, 6, 10, 11, 12 and 13, which include 38,113,479 publications, 84.18% of the total publications in WoS, are considered technology-related and thus comprise the experiment data set.

ID	Category name	Number of publications	Percentage of publications	Technology related
1	CHEMISTRY, PHYSICS AND ASTRONOMY	9,776,937	21.59%	yes
2	CULTURE	3,006,769	6.64%	no
3	EARTH AND ENVIRONMENTAL SCIENCES	2,547,255	5.62%	yes
4	ECONOMICS, MANAGEMENT AND PLANNING	1,089,417	2.40%	no
5	ENGINEERING SCIENCES	3,548,955	7.84%	yes
6	HEALTH SCIENCES	1,138,026	2.51%	yes
7	INFORMATION AND COMMUNICATION SCIENCES	351,259	0.77%	no
8	LANGUAGE, LINGUISTICS AND LITERATURE	1,231,031	2.72%	no
9	LAW	261,900	0.58%	no
10	LIFE SCIENCES	7,235,654	15.98%	yes
11	MATHEMATICS, STATISTICS AND COMPUTER SCIENCE	2,153,767	4.75%	yes
12	MEDICAL SCIENCES	15,960,838	35.25%	yes
13	MULTIDISCIPLINARY JOURNALS	886,547	1.96%	yes
14	SOCIAL SCIENCES	2,728,887	6.02%	no

Table 6.4: NOWT categories of scientific disciplines

6.1.3 Reference parsing

Reference parsing aims to locate, identify and extract publication information from references. It interprets references and tries to extract pieces of information that we can use to limit the set of potential WoS publication matches. As mentioned in Chapter 2, literature references in patents may contain information on attributes of publications in WoS, like author name, publication source name, publication title, source volume number, and so on. Those attributes are very important as they can work as glue to generate potential matches and further contribute to validating whether the link is correct or not. The identification of scientific publications hinges primarily on matching attributes from WoS with the extracted publication attributes from references, a process we will describe in more detail in Section 6.2. The publication attributes that carry useful publication information and have been mentioned in Chapter 2 play an important role in the publication identification. They are grouped according to the type of information contained in them in the list below.

- Identifier of publication:
  - DOI
- Source of publication:
  - Publication source name
  - Publication source identifier (ISSN or ISBN)
  - Page or page range
  - Volume
  - Issue
  - Publication year
- Publication entity:
  - Author name(s)
  - Publication title

These attributes are often accompanied in literature references by labels, which are specific attribute names or abbreviated attribute names that indicate where the attributes occur. In case lit-

erature references use such labels, we can use them to locate the attribute information and extract them. For example, a literature reference *"wieman et al., laser-frequency stabilization using mode interference from a reflecting reference interferometer, opt. lett., vol. 7, no. 10, pp. 480-482 (1982) figure 2, abstract, p. 482 col. 1, par. 4."* contains 'vol', which is the label used to locate the position of the volume information. In this case, the '7' immediately following the label is the volume number. Given the fact that attribute labels are not harmonized, we manually collected the possible forms of attribute labels to enable the extraction of attribute values.

Not all the attributes listed above can be extracted by using labels. The publication titles are displayed in text strings and do not have clear boundaries, and the same goes for the publication source titles. These attributes cannot be identified at this phase, and will be addressed in the phase of match candidate refinement which we describe in Chapter 6.2.2. The attributes we are addressing in this phase are either labeled or show clear positions in the literature references. This allows us to easily locate them using labels, and then identify and extract their values using the consistent patterns that exist within these literature references. For instance, whenever the label 'vol' appears, a subsequent number denotes the volume number, which we can then extract. In reality, these patterns are frequently more complex, because the a literature reference may contain an issue range ('issue 7-9') or may include additional words or characters between the primary labels and the attribute values ('issue supplement 3' or 'page a4'). Based on those publication attribute patterns, we use regular expressions to extract the publication attributes from references. If some attribute labels are missing, then the corresponding attributes are regarded as not available.

Parsing and segmenting references into individual publication attributes enables very exact interpretation of references and further contributes to linking it with records in WoS. Table 6.5 shows the results after parsing the example *"wieman et al., laser-frequency stabilization using mode interference from a reflecting reference interferometer, opt. lett., vol. 7, no. 10, pp. 480-482 (1982) figure 2, abstract, p. 482 col. 1, par. 4."*.

Author(s)	Volume	Issue	Starting page	Ending page	Year	ISSN/ISBN	DOI
Wieman	7	10	480	482	1982	-	-

Table 6.5: The reference parsing result

However, parsing according to labels is quite limited, because there are cases when attribute labels are omitted from literature references, for example, *"brown, r. l., et al., pdgf and tgf-a act synergistically to improve Wound healing in the genetically diabetic mouse, journal of surgical research, 56, (1994),562-570."* presents a situation where readers are supposed to interpret the sequence of numbers as certain attributes simply by looking at their format and order. However, the numeric attributes without labels do not follow a unified format or order. There are various ways to mention attributes without labels. The six formats below are common and easily readable for both people and computer programs. Even though the attributes are not labeled clearly, they are easily distinguished.

1 [year][volume][issue][page range] e.g. (1983) 43 (4): 1790-1797

2 [year][volume][page range] e.g. (1983) 43: 1790-1797

3 [year][page range] e.g. (1983) 1790-1797

4 [volume][year][page range] e.g. 43, (1983) 1790-1797

5 [year][volume][issue] e.g. (1983) 43 (4)

6 [volume][page range][year] e.g. 43: 1790-1797 (1983)

Apart from the references with labels and fixed formats of mentioning publication information, some references present publication information in a very vague or unstructured way. To resolve these one would need to expand the selection of patterns used for attribute recognition. However, if we did that we would end up introducing patterns that are not generic, and result in misinterpretation and introduce mismatches in later phases. So, in this thesis, we will only focus on the recognizable patterns of publication information in literature references.

6.2 Matching

We begin the matching with a selection phase, in which we generate a set of match candidates based on the numeric publication attributes extracted as described in Section 6.1.3, followed by a candidate refinement phase matching string attributes. Due to the large volume of data, it is not feasible to fully evaluate all potential reference-publication matches that the two databases offer, which is why the selection phase is required. Also, because the selection is based on rules, and a single rule cannot cover all variants accurately, we repeat the selection and refinement phases over a total of four different rounds as shown in Figure 6.1, each using slightly adjusted selection rules, which is described later in Table 6.7. This recursive approach of 'onion peeling' allows us to incrementally shrink the set of unresolved literature references.

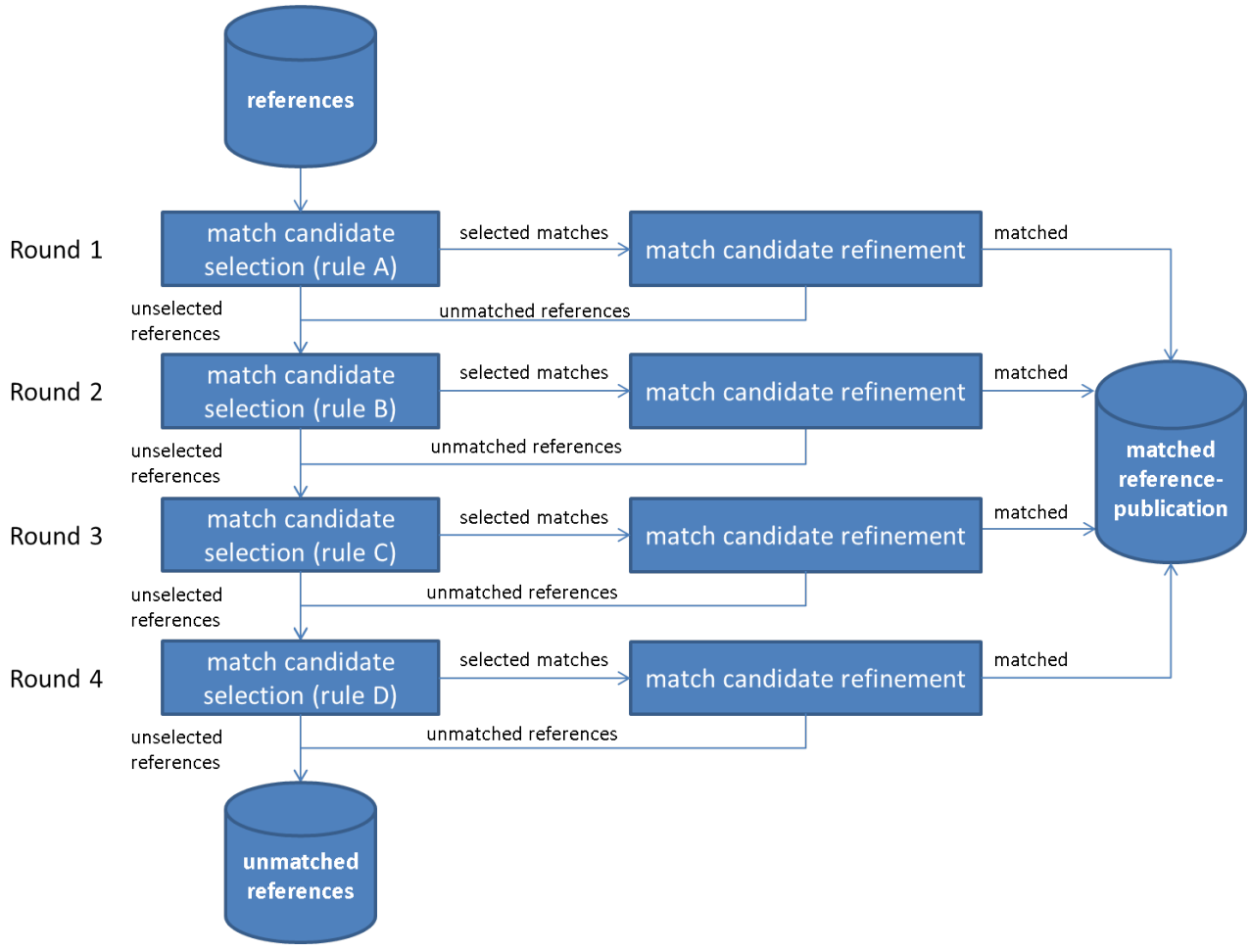

Figure 6.1: The matching process

In each round we first get a set of match candidates and subsequently we refine the set of match candidates by matching publication titles, author names and publication source names. At the end of each round, the matches we are confident about are moved into a pool of matched reference-publication pairs, while the leftover references that were not matched are passed to the next round of candidate selection. This process of match candidate selection will be described in Section 6.2.1. The detailed description of the process of match candidate refinement will be present in Section 6.2.2.

6.2.1 Match candidate selection

In this section, our aim is to select, for each literature reference in the patents, a set of potential reference-publication match candidates that we can later refine. It is the first step to link references in PATSTAT to publications in WoS. The further refinement approaches like publication title matching, publication source matching and author name matching are performed based on those potential reference-publication matches. The selection rules for match candidates are based on the publication attributes that have been extracted during the previous reference parsing phase. Extracted attribute information from patents' literature references can function as "glue" for reference-publication linkage by selecting an initial set of publications from WoS that share these attributes. The primary idea behind the selection is to match attributes or a combination of attributes to discover potential matches. Of course, it is entirely possible that references have no matching publication candidates in a certain round. These are simply sent on to the next round of selection and refining, where different rules are used.

We discussed the reference parsing and, with it, the extraction of attribute values in Section 6.1.3. As displayed in Table 6.5, the attributes are: *author(s)*, *volume number*, *issue number*, *starting page*, *ending page*, *year of publication*, *ISSN or ISBN*, and *DOI*. To develop selection rules for the subsequent rounds of the matching process, we first need to answer the question 'Which attributes or combination of attributes can be used as glue to select reliable match candidates?'

There is no doubt that DOI has the most potential to be used as matching glue, since it is a unique publication identifier. A DOI refers to one unique document and one document only. Logically we should focus on the references which have a DOI available first, and match references with publications simply by comparing the DOI. However, because the DOI is frequently not included in the literature references in our data set, and it is quite difficult to conclude patterns we can use to extract DOIs, we skip comparing DOIs in this thesis. The references that include a DOI are typically very structured, so they can also be matched by other attributes.

However, besides DOI, no single attribute is sufficient to positively identify a reference-publication match. As a result, the combination of available volume number, issue number, page ranges and publication year are taken as "glue". The matching is performed simply by comparing the attributes from literature references and publications and checking whether they are equal.

Note that author names are not considered as part of the glue even though they can be extracted. This is because author names can be written in various forms and fuzzy matching of text strings is more complicated than comparing numeric values with numbers. This also goes for the publication title and the publication source name, which are even more complicated as they cannot easily be extracted due to a lack of clear boundaries. Recall our earlier example from Section 6.1.3, repeated below, and its parsing result presented in Table 6.5.

> "wieman et al., laser-frequency stabilization using mode interference from a reflecting reference interferometer, opt. lett., vol. 7, no. 10, pp. 480-482 (1982) figure 2, abstract, p. 482 col. 1, par. 4."

We search the scientific publications in WoS with the same volume number (7), issue number (10), page ranges (480-482) and year (1982) and get the record in WoS shown in Table 6.6.

In this example, we use all the numeric publication attributes to compare references and records in WoS. We only obtain one potential matching publication which happens to be exactly the correct one. However, considering all the numeric attributes does not work for all the references, because attributes may be missing, incorrectly mentioned in references, or incorrectly parsed. In

UT	A1982PJ67900006
title	laser-frequency stabilization using mode interference from a reflecting reference interferometer
first author	wieman, ce
other authors	gilbert, sl
source	optics letters
year	1982
volume	7
issue	10
starting page	480
ending page	482

Table 6.6: Example of entry linkage

that case, we need to accept match candidates that are comprised of a reference and a publication that are not equal in strictly all their attributes. There is a trade-off between the number of potential match candidates and the accuracy of the mach candidate selection. With a combination of less attributes, more potential matches will be retrieved. Correspondingly, this approach will generate more incorrect matches.

To mitigate this limitation, we perform matching in a recursive way in four rounds. For each round, we use different combinations of attributes to link references with publications in WoS, as explained in Table 6.7. We take the parsing quality and significance of attributes into consideration when constructing the combinations. For instance, it is common not to mention the issue number in references, while the ending page of a page range is sometimes parsed incorrectly, so we have combinations in which either of them are not included. Through exploring the data set and using domain knowledge, we find that the following four combinations provide a good trade-off between coverage and reliability.

The whole matching process is divided into four rounds, as shown in Figure 6.1. This means we

Rule	Round	Publication attributes
A	round 1	volume, issue, starting page, ending page, publishing year
B	round 2	volume, starting page, ending page, publishing year
C	round 3	volume, issue, starting page, publishing year
D	round 4	volume, starting page, publishing year

Table 6.7: Linking rules

can start with a strict selection rule to get precise matches and shrink the data set by removing the matched references for the next round. The recursive way allows for more precise and less time-intensive matching.

First, we try to link all the attributes of all the references with publications, to generate a set of reference-publication match candidates. If a reference fails to be linked with any publication, it is sent to the next round, to try again with a different and less strict selection rule. If a reference is linked to some publications in this candidate selection phase, then we perform refinement by title matching, author name matching and publication source name matching to verify each of the match candidates. which is described in detail in Section 6.2.2. References which fail to be verified in this phase will continue to the next round along with the other unsolved references.

Then we start the next round, linking the unmatched references that remain with publications using less strict selection rules. The different combinations of publication attributes are able to catch new match candidates which were filtered out by the rules used in the previous rounds.

6.2.2 Match candidate refinement

During the match candidate selection phase, we used only numerical attributes and skipped the publication title, the publication source name and the author name. However, the numeric attributes alone are not sufficient to convince us of reliable matches. The match candidates we get so far only provide us with the initial and rough linkage. They present a set of potential matches but still contain plenty of mismatches. The three unused publication attributes are all in the form of a text

string. They carry more distinct information and thus are a potent aid for matching references with publications.

Simply put, after match candidate selection phase, for a particular literature reference, we may get a list of publications it may point to, when we only match them with numeric publication attributes. What we need to do next is to use the string attributes to validate which publication is correctly matched with this reference, if any. So, in this phase, the string attributes serve for refining and validating the reference-publication match candidates.

Among the three string attributes, publication names have stronger identifying power, because authors tend to give their publications distinct titles. In addition, compared with the other two attributes, publication titles are prone to be longer in length and very different from each other, whereas author names and publication source names are often shorter and less diverse. Because of this, we let publication title matching follow subsequently after the match candidate selection, as the publication titles have the greatest potential for easily reducing the number of potential matches. A closer analysis of the data revealed that, after candidate selection, when candidate reference and candidate publication titles match, this is sufficient to declare the match as correct. So, if the the publication titles from references and WoS in a match candidate are matched, then this match candidate is considered matched correctly and we send the pair to our final pool of correct matches.

However, some patents' literature references do not include publication titles, as shown in Table 6.8. These references cannot be matched with the publication titles from WoS. We need to use author name matching and publication source name matching together to verify match candidates with this kind of references. Author names and source names are not very unique (for example, there are less than 20000 distinct publication sources covering over 38 million publications in WoS), so we need to use the combination of the author names and source names to verify the match candidates. Only when both the author name and publication source name in a match candidate are matched after our candidate selection phase, we consider it to be a correct match.

ID	Literature references
961259352	v.sorokin et al, zhurnal organiceskoj khimii, vol. 30, no. 4, pp. 528 530 (1994).
959189039	shirai m. et al. zoolog sci. (1996) 13(2):277-283.
962548501	yonsei med j vol. 41, no. 1, 2000, pages 82 - 8
963047474	lalla a et al. west indian med j. vol. 50, no. 2, juin 2001, pages 111 - 6

Table 6.8: Patents' literature references

In principle, for the match candidates refined by publication titles, it would be ideal if we could further confirm this match by also using the author names and publication source names. However, analyzing the matching results (see Chapter 7), we find out only publication title matching is sufficient to obtain the correct matches. Furthermore, we may fail to match the author names and publication source names for the correctly matching candidates refined by publication titles, because the way author names and publication source names are mentioned in literature references are more diverse and flexible than the patterns we use to match them. Further refinement after successfully matching publication titles is thus unnecessary and unwanted.

Taking the above analysis into consideration, we devise a procedure of match candidate refining as shown in Figure 6.2. This figure presents a look inside the match candidate refinement step in the

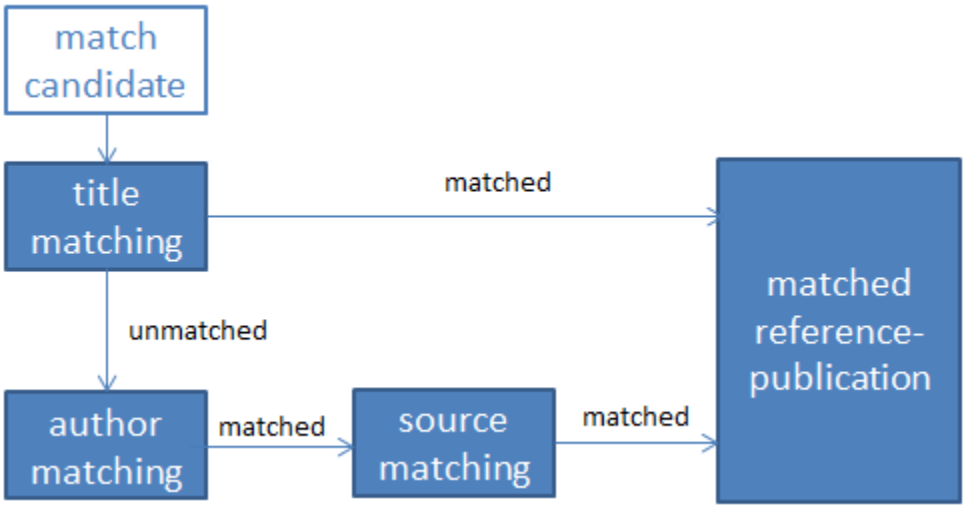

Figure 6.2: Match candidate refinement

larger process in Figure 6.1. We perform title matching first, after which we match author names and publication source names for the match candidates with unmatched publication titles.

Publication title matching

At this phase, we compare the reference string with the publication title for each of our match candidates. Because the publication titles are difficult if not impossible to reliably extract, we compare the entire reference string with the title by determining whether the title is part of the reference. The ideal situation is where the publication title is mentioned exactly as part of the reference, but this is often not the case for all the match candidates as typographical errors and content changes may appear in literature reference strings. There are various reasons:

1. Typing errors from applicants or inventors.
2. Errors or changes are introduced by editors or examiners in the process of transcription or in the process of filing patents.
3. Errors or changes happen when the database is updated.
4. The information is originally in another language and translated into English.

Because of the errors and changes, characters, especially punctuation, may be inserted, deleted, substituted and transposed. As a result, we fail to match most reference strings with the publications if we require references to contain the exact publication title. Instead, we use similarity measures described in Chapter 5 to determine the match quality. If a match candidate is correctly matched, it should meet the condition that the publication title should be similar to a part of the corresponding reference string.

We previously introduced several ways of comparing strings in Chapter 5. In order to have consistent and non-conflicting results, we need to choose one of these. Because the publication title should be part of the longer literature reference string, the two possible options are semi-global alignment and improved fitting alignment. We performed several experiments to measure the performance of these two approaches. The experimental setup and outcomes are described in Section 7.2.

Author name matching

The formats of people's names are difficult to parse because people have their own preferences regarding how to write their names. Even a single author may use different formats of his or her name in different publications. This is why the names cannot be compared with the author names from WoS directly, even though for most references the author names can easily be retrieved by the label 'et. al'. Table 6.9 gives examples of matched author names from correct reference-publication matches. The author names from WoS have already been processed by researchers at CWTS for other research projects. They are in the form of *'family name, initial(s)'*. For instance, *'pessino, a'* corresponds to an author whose family name is *'pessino'* and initial is *'a'*.

Patents' references	Publications in WoS
pessino, a	pessino, a
daugherty, p.s.	daugherty, ps
charng	charng, mj
bao. s	bao, sp
craig w. hodgson	hodgson, cw
k. uoto	uoto, k
wolfgang beckmann	beckmann, w
chyuan-der hwang	hwang, cd
saffard	safford, r

Table 6.9: Author name matching

The example names in Table 6.9 are the most commonly occurring author name representations in the references. Note that, no matter how flexible the forms of given names and the initials, authors tend to mention their complete family names. In addition, they will mention their given names or initials partially or fully before or after family names. Because author names are provided in a well-structured way in WoS, we use the WoS format as the starting point, from which we generate a set of alternate representations of author names, which we then compare to the listed author names in the literature references. These alternative representations are based on patterns which

Number of initials	Frequency
1	4,806,517
2	4,419,803
3	518,156
4	50,629
5	5,369

Table 6.10: Distribution of initials in author names

we have observed in the literature references. We will describe these using an abstract notation in the following paragraphs.

For an author name *'family name, initial(s)'*, we use $[FN]$ to represent the family name and use $[I_x]$ to represent the x^{th} initial. Usually, authors have one, two or three initials in the their names in WoS as shown in Table 6.10, which covers 99.43% of all the instances in WoS. Then, ignoring punctuation, we use n to represent the number of initials in author names ($n = 1,2,3,4,5$) and the author names in WoS are defined as:

$$[FN][I_1][I_2]...[I_n]$$

or

$$[I_1][I_2]...[I_n][FN]$$

In the references, typically, the authors' family names are put in either the first or the last part of names. For each part of the given names, authors usually mention either the initial or the given name starting with the same initial. For instance, for the author name 'Mark E.J. Newman', the first part of the given name is either mentioned as 'Mark' or 'M' and the order in which each part of authors' complete given names appear is fixed. We represent any alphabetical character with c and define $c*$ as a sequence of zero or more alphabetical characters. Then we find that the pattern of authors' given names in patents' references can approximately be represented as:

$$[FN][I_1c* \ I_2c* \ ... \ I_nc*], \ n = 1,2,3,4,5$$

or

$$[I_1c* \ I_2c* \ ... \ I_nc*][FN], \ n = 1,2,3,4,5$$

Note that we leave out all the spaces in names from both databases. This is because usually WoS records family names without spaces as a means to address family names with prefixes, such as commonly occurring prefixes in Dutch family names like 'van' or 'de'. For instance, 'van de Loo' may be mentioned in the literature references, but might appears as 'vandeloo' in WoS. It is also possible that authors only mention the first few parts of their given names. For instance, the author name 'Robert J.W. Tijssen' sometimes is mentioned as 'Robert Tijssen'. So we need to modify the name pattern above. We first only include the first initial in the name from WoS in the pattern. If it does not match the author name from patents' references successfully, this might be because there is a second part of the given name in the reference, so then we include the second initial in the pattern and use it to match author names. We continue to include initials in the pattern until the pattern matches to the author name or we run out of initials. Based on this approach, we run through possible patterns of the names which may be mentioned in patents' references. An example of given names with 2 initials is shown in Table 6.11. Note that, for the purpose of clear explanation, we define $c+$ as a sequence of zero or more alphabetical characters and divide I_ic* into I_i and I_ic+, to represent the i^{th} part in the given name. All punctuation in author names is replaced by whitespaces.

It is important to point out that we are not interested in which part of an author name in a literature reference stands for which specific name component, like family name, first name or middle name, or which order they are presented in. Instead, we generate a list of potential name representations from the WoS author name in the hopes that one matches some text string in the literature reference. It is not feasible to identify exact name components in literature references, because the way names are represented differs among for instance regions, languages and cultures.

Something that is still unaccounted for are name variations resulting from typographical errors and phonetic variations for non-English names occurring in the patents' references, like the last example in Table 6.9. Because of these errors and variations, we also need to match authors' family names approximately. We use string P to represent the extracted name from a reference or

Patterns	Examples
[FN] [I_1c+]	Lamers Wout
[FN] [I_1]	Lamers W
[FN] [I_1] [I_2]	Lamers W S
[FN] [I_1c+] [I_2]	Lamers Wout S
[FN] [I_1] [I_2c+]	Lamers W Solex
[FN] [I_1c+] [word staring with I_2]	Lamers Wout Solex
[I_1] [FN]	W Lamers
[$1I_1c+$] [FN]	Wout Lamers
[I_1] [I_2] [FN]	W S Lamers
[I_1c+] [I_2] [FN]	Wout S Lamers
[I_1] [I_2c+] [FN]	W Solex Lamers
[word staring with I_1] [word staring with I_2] [FN]	Wout Solex Lamers

Table 6.11: Patterns for author name matching based on the name 'Wout Solex Lamers'

the complete reference if the extracted name is not available. We need to compare string P with the author name from WoS.

First, we check if the family name is exactly contained in string P. If not, we compare every separate word in string P with the family name from WoS by calculating the string similarity based on the edit distance according to the formula defined in Section 5.1.2. We set a threshold for the similarity to say whether it is matched or not. We choose the word with the highest string similarity compared with the family name from WoS, if the similarity is higher than the threshold, then it is considered the approximate family name. If the similarity is lower than the threshold, then we say the family name does not feature in this reference. After the family names are matched, we use it in the patterns of author names in Table 6.11 to further explore whether the author names are really matched.

ID	Literature references
959189039	shirai m. et al. zoolog sci. (1996) 13(2):277-283.
961259352	v.sorokin et al, zhurnal organiceskoj khimii, vol. 30, no. 4, pp. 528 530 (1994).
962548501	yonsei med j vol. 41, no. 1, 2000, pages 82 - 8
963047474	lalla a et al. west indian med j. vol. 50, no. 2, juin 2001, pages 111 - 6

Table 6.12: Patents' literature references

UT	Source name	Abbreviation
A1994PT70300011	zhurnal organicheskoi khimii	zh org khim
A1996UR48400010	zoological science	zool sci
000085755000013	yonsei medical journal	yonsei med j
000171112400005	west indian medical journal	w indian med j

Table 6.13: Publication source names in WoS

Publication source name matching

Publication source names in patents' literature references are not extracted in the phase of reference parsing because we cannot collect patterns or labels to locate them in reference strings. The references shown in Table 6.12 actually contain publication source names in the string, but it is difficult to automatically extract them. However, from the side of publications of WoS, publication source names are well structured. The full names and common abbreviated names are provided as shown in Table 6.13.

We use the publication source names provided by WoS to build a query for matching. First, we simply search for whether the full source names are in the references. If the reference contains the source name, then we can conclude that the publication source matches. However, it is common that inventors or authors use the abbreviated forms of journal names. They tend to use the existing well-recognized names for better academic communication instead of just inventing abbreviations for source names according to their preferences. As a result, even though the usage of abbreviated names introduces some variations in publication source name matching, the abbreviated source

titles are still organized and not difficult to process. Looking into the publication source names in patents' references and the abbreviated source names in WoS, we find that references usually cover the abbreviations but leave the *stop words* out. Stop words refer to words that do not carry important information. In the situation of publication source name matching, stop words are 'to', 'in', 'on', 'of' and many others. They are usually prepositions or grammatical articles in the source names. Appendix A presents the list of stop words used in this section. We filter out the stop words in both references and abbreviations of source names in WoS to harmonize the usage of stop words between these two sources.

After removing the stop words, we use the source names in WoS to build patterns to identify the source names in references. If we are not able to extract source names from references, then we can know that the source names do not match. After studying the formats of source names in references, we find that source names in references are primarily composed of the identical words of source names in WoS in the same order. However, a slight difference between source names from these two sources may occur and these usually stem from the fact that some words in the publication source name in references may be longer than the corresponding words in the source name in WoS. For instance, a publication source name in a literature reference is 'west indian med j' while the abbreviated name in WoS is 'w india med j'. For the first word, authors or inventors use 'west' which contains the first word of publication source, 'w', in WoS. We use W_i to represent the i^{th} word in a publication source name in WoS. So a publication source name consisting of n words can be represented as:

$$[W_1\ W_2\ W_3 ...\ W_n]$$

We represent any alphabetical character with c and define $c*$ as a sequence of zero or more alphabetical characters. Then the pattern of source names in patents' references can be represented as:

$$[W_1 c*\ W_2 c*\ W_3 c* ...\ W_n c*]$$

According to this pattern, we are able to extract publication source names as shown in Table 6.14.

Literature references	extracted source name	source names in WoS
v.sorokin et al, zhurnal organiceskoj khimii, vol. 30, no. 4, pp. 528 530 (1994).	zhurnal organiceskoj khimii	zh org khim
shirai m. et al. zoolog sci. (1996) 13(2):277-283.	zoolog sci	zool sci
yonsei med j vol. 41, no. 1, 2000, pages 82 - 8	yonsei med j	yonsei med j
lalla a et al. west indian med j.vol. 50, no. 2, juin 2001, pages 111 - 6	west indian med j	w indian med j

Table 6.14: Extraction of patents' publication source name

7 Results

In this chapter, we will present our results. First, we introduce the measures we use to analyze results in Section 7.1. Then, we demonstrate the algorithm comparison of improved fitting alignment and semi-global alignment performed on publication title matching in Section 7.2. The algorithm comparison tells which is more suitable in this case. We present the reference-publication result analysis to measure the correctness and performance in Section 7.3.

7.1 Measures

We apply *precision* and *recall* analysis to evaluate the matching results. The precision and recall metrics reflect how precise and how complete the matching result is. The terms *true positives (tp)*, *true negatives (tn)*, *false positives (fp)*, and *false negatives (fn)* are used to measure the matching result. They come from combining the terms *positive* and *negative* with the terms *true* and *false*. Among them, the terms positive and negative refer to the matching result: if a pair match, then it is positive, otherwise, it is negative. The terms true and false reflect whether the matching result corresponds to the true condition. For instance, if a match is correct, then it is true, otherwise, it is false. In the context of string matching, precision is the percentage of the matched pairs that are correct while recall represents the percentage of real matches that are obtained in the matching process. *Accuracy* is a broader measure that refers to the percentage of correctly matched pairs over the entire population. These three measures are calculated as follows:

$$Presicion = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp+fn}$$

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn}$$

Both the precision and recall measures are important and we need to take both into consideration when we evaluate matching results. *F-measure*, which is the harmonic mean of recall and precision, is thus introduced to measure the results. As we aim for a matching result that reflects the true condition of the reference-publication title pairs and false positives are no better or worse than false negatives, we consider recall and precision equally important, so we choose the standard F-measure which is referred as F_1 and gives equal weight to recall and precision.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

7.2 Improved fitting alignment vs. Semi-global alignment

In this section, we will present a performance comparison between the improved fitting string alignment algorithm and the semi-global string alignment algorithm. In the experiments, we use the method as described in Section 5.2. The result is a normalized similarity score on the interval $[0,1]$. For the semi-global string alignment algorithm as described in Section 5.1.4, we set the match score as 1 when the character from the title publication equals the one from the reference and the match score as 0 if they mismatch. The gap penalty is set to -1 because the character insertion or deletion in the publication titles cannot be ignored when fitting titles into references.

We obtain the semi-global similarity to the interval $[0,1]$ by dividing the alignment score by the length of the shorter string, which means that it now signifies the proportion of the strings that is aligned. The similarity scores of the improved fitting alignment and the semi-global string alignment are now normalized to the same interval.

As there is no existing right answer to tell us whether a publication title truly match with a literature reference, we need to manually check the correctness of publication title results. To analyze the performance of these two string alignment algorithms, we randomly select 2000 matching

Threshold	Measure	Improved fitting alignment	Semi-global alignment
0.5	*Precision*	0.9974	0.9963
	Recall	0.9939	0.9351
	Accuracy	0.995	0.9605
	F_1	0.9959	0.9647
0.6	*Precision*	1	1
	Recall	0.9879	0.9307
	Accuracy	0.993	0.96
	F_1	0.9939	0.9641
0.7	*Precision*	1	1
	Recall	0.9801	0.9273
	Accuracy	0.9885	0.958
	F_1	0.9899	0.9623
0.8	*Precision*	1	1
	Recall	0.9749	0.9221
	Accuracy	0.9855	0.955
	F_1	0.9873	0.9595
0.9	*Precision*	1	1
	Recall	0.9541	0.9134
	Accuracy	0.9735	0.95
	F_1	0.9765	0.9548

Table 7.1: Matching result measures

instances. Each instance consists of one patent's literature reference and the title name of its candidate matching publication. We analyze the performance from the perspective of the matching results (see Table 7.1), overall obtained similarity distribution (see Figure 7.1), quartiles (see Figure 7.2) and the comparison of two algorithms (see Figure 7.3).

From Table 7.1 shows the precision and recall analysis for both alignment algorithms for increas-

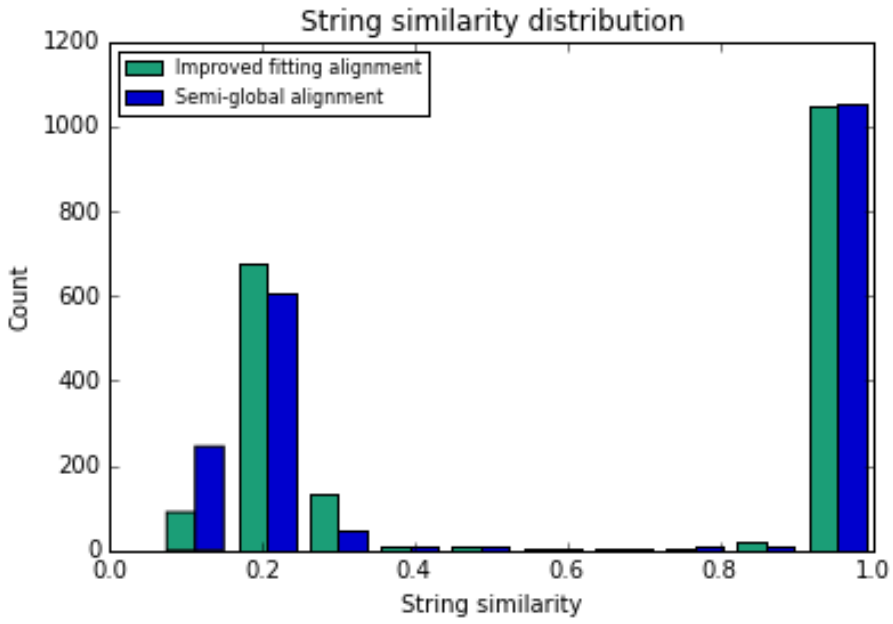

Figure 7.1: Similarity distribution of two string alignment algorithms

ingly similarity thresholds. It is observed that the improved fitting alignment yields better publication title publication results for all the measures. From Figure 7.1, we observe that in some similarity intervals, the improved fitting alignment yields a higher number of instances while in the other similarity intervals, it obtains less instances. But generally, they have very similar distributions.

Furthermore, Figure 7.3 shows that most observations are situated near the red line where these two algorithms yield identical string similarities. This is because in essence, these two algorithms perform similarly since they use dynamic programming techniques based on comparable similarity matrices. This helps validate the improved fitting alignment. The box plot shown in Figure 7.2 and variants in Table 7.2 further show that they yield a similar result distribution.

However, from Figure 7.3, we observe that there are a few outliers that score high in the improved fitting alignment and low in the semi-global alignment. We use $SIM_{fitting}$ and SIM_{semi} to represent the similarity between two strings obtained by the improved fitting alignment algorithm and the semi-global alignment algorithm respectively. Examples of those outliers are presented in Ta-

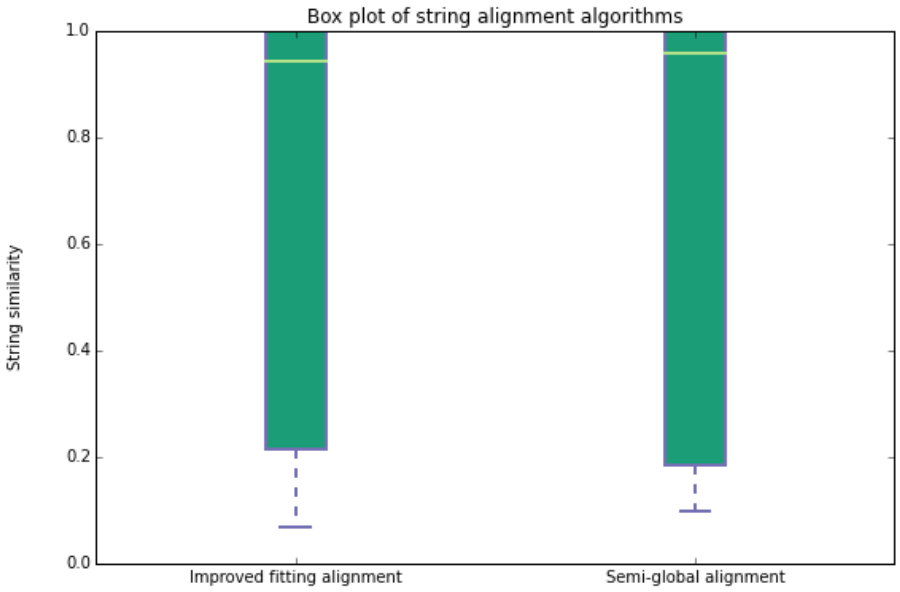

Figure 7.2: Similarity overview of two string alignment algorithms

	Improved fitting alignment	Semi-global alignment
minimum	0.0592	0.0968
25th percentile	0.2230	0.1860
50th percentile	0.9624	0.9578
75th percentile	1.0	1.0
maximum	1.0	1.0
mean	0.6567	0.6210
standard deviation	0.3820	0.4001

Table 7.2: Similarity obtained by two alignment algorithms

ble 7.3. The large difference between similarity is due to the incomplete publication title names mentioned in the literature references. It is the reason why we introduce the improved fitting alignment algorithm in the first place.

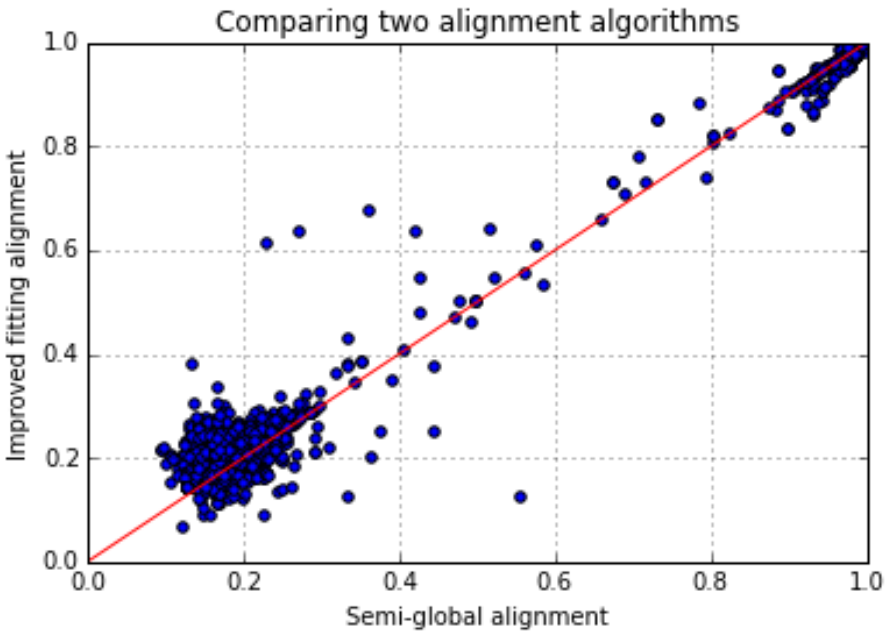

Figure 7.3: Result comparison of two string alignment algorithms

Literature reference	Publication title	$SIM_{fitting}$	SIM_{semi}
ieee transactions on energy conversion. vol. 4, no. 3, september 1989, new york us pages 436 - 445; ranade et al: 'a study of islanding in utility-connected residential photovoltaic systems'	a study of islanding in utility connected residential photovoltaic systems 1 models and analytical methods	0.6981	0.3962
archives of otolaryngology, vol. 109,no. 6, june 1983 pages 372-375, e. calenoff et al. 'bacteria-specific ig e in patients with nasal polyposis.'	bacteria specific ige in patients with nasal polyposis a preliminary report	0.6933	0.4267

Table 7.3: Examples of outliers

Figure 7.3 also contains a few outliers which score low in the improved fitting alignment and high in the semi-global alignment. Looking into those outliers, we find that they occur because the optimal string alignment in the semi-global alignment does not contain the LCS in the improved

fitting alignment. This results from short LCSs which fail to locate a precise position of the title in the reference. In addition, we find out that when a publication title is completely included in the literature reference, the semi-global algorithm will obtain slightly higher similarity. In this case, the LCSs from two strings are aligned, but the semi-global alignment obtains better alignment for the rest part while, in the improved fitting alignment, the truncated string from the reference used to compute the edit distance with publication title is not part of the optimal alignment.

So, we could say that when the publication titles are completely mentioned in the patents' literature references, semi-global alignment performs better than the improved fitting alignment when it comes to finding the best fitting alignment. Still, while the improved fitting alignment scores are slightly lower than the semi-global alignment scores in these cases, they both correctly find matches. The slight difference of the similarity values do not really yield different results, because, with a slightly lower similarity, the matching pairs can still pass the similarity threshold.

However, when the publication names are not mentioned completely, the improved fitting alignment outperforms the semi-global alignment. To confirm this, we constructed a sample data with 1000 randomly selected instances. Each instance consists of a publication title and a corresponding literature reference which contains this title incompletely. In this data set, each reference-title pair are truly matched. The results are shown in Table 7.4. Apparently, the improved fitting alignment gets higher average similarity and better matching results.

	Improved fitting alignment	Semi-global alignment
Mean similarity	0.7426	0.5992
Number of matching pairs with threshold 0.7	957	4
Number of matching pairs with threshold 0.6	1000	573

Table 7.4: Manual comparison of two alignment algorithms

From in this section, we conclude that the improved fitting alignment algorithm performs better in matching publication titles with the literature references due to its better performance with partial publication titles in references.

7.3 Reference-publication matching result

In this section, we talk about the final reference-publication matching results we obtained. As described in Section 6.2.2, we match patents' literature references to scientific publications in WoS in four rounds, each with different match candidate selection rules and a subsequent refinement phase based on either the publication title or the author name together with the publication source name. The match candidates selection rules and the refinement approaches influence the correctness of matches. For instance, we have more confidence in matching results obtained from the publication title matching phase of the first round, than in those from the author name and publication source name matching phase of the last round, because in the first case, more strict selection rules and a more unique and precise refinement approach are used. According to this, we give all the final matches a tag explaining which round they come from and through which approach they are refined. Table 7.5 presents the number of matches and Table 7.6 gives an overview of the matching results.

Total references	11,906,361
Unmatched	5,559,676 (47%)
Matched	6,346,685 (53%)

Table 7.5: Matching results

Note, from Table 7.6, that the first two rounds contribute the most matches and that most matches are refined by the publication title. This is quite ideal as it reflects that these matches are found using more strict and precise approaches. To evaluate the quality of the results, we performed precision and recall analysis. It is not feasible to check the whole result set because of its immense size. Instead, we randomly collected a sample of 3,000 references from the entire set of references

Round	Number of matches	Percentage of total references	Title matching	Percentage of matches this round	Author and publication source name matching	Percentage of matches this round
Round 1	2,908,418	24%	2,509,842	86.30%	398.576	13.70%
Round 2	3,005,061	25%	2,004,254	66.70%	1,000,807	33.30%
Round 3	150,848	1%	96,358	63.88%	54,490	36.12%
Round 4	282,358	2%	59,743	21.16%	222,615	78.84%

Table 7.6: Distribution of matching results

used in the experiment. We manually identified the scientific publications these references point to. These manually obtained results are considered the correct and real matching results. We compare these results with the results obtained automatically in this thesis, as well as with the matching results which CWTS are using for research projects now. The evaluation result is presented in Table 7.7.

	Predicted matched	Predicted unmatched
Truly matched	tp : 1545	fn :473
Truly unmatched	fp :15	tn :967

(a) Results obtained in this thesis

	Predicted matched	Predicted unmatched
Truly matched	tp : 810	fn :1208
Truly unmatched	fp :3	tn :979

(b) Results used in CWTS

Table 7.7: Result of automatic reference-publication matching

We can obtain measures to evaluate the results as shown in Table 7.8. We observe that the results we obtain using our methods are better than the CWTS results from the perspective of correctness and completeness.

When we compare the results of the two methods, we find that the CWTS results are developed in a more conservative way, in which exact matches of the text attributes form the backbone of the matching, while numeric attributes are disregarded. In contrast, the approach introduced in this

Measure	Results obtained in this thesis	Results used in CWTS
Precision	0.9904	0.9963
Recall	0.7656	0.4014
Accuracy	0.8373	0.5963
F_1	0.8636	0.5722

Table 7.8: Comparison of two alignment algorithms

thesis includes a detailed parsing method for patents' literature references and uses approximate string matching to process publication attributes in text format. These improvements explain why the overall performance of our new method is better.

However, even though we use a more flexible matching method, 97 references in this 3,000 data sample are matched using the existing CWTS method but not matched by our new method. We dug into those instances and found that the problem occurs in the phase of match candidate selection. We use four selection rules but those four still cannot cover all the situations. More diverse selection rules are required. For example, a combination of starting page, ending page and publication year seems a good candidate selection rule for an additional fifth round of matching.

The *Precision* in the results from our method is higher than 0.99, which is quite promising. Studying the mismatch instances, we find out that they are obtained by author name matching and publication source name matching. It indirectly reflects that we chose the right strategy in giving publication titles higher priority in the phase of match candidate refinement. It also demonstrates the promising performance of the improved fitting alignment algorithm introduced in this thesis. But it also indicates that the approximate string matching algorithms perform less well for author name matching and publication source matching algorithms than publication title matching. This is because author names and publication source names are shorter and less unique than publication titles. This is an inherent problem in the data and is therefore difficult to resolve.

8 Data utilization

In this chapter, we will further explore what type of insight we can derive from the matching results. The scientific contribution of the automatic identification of publications in patents' literature references is not only about introducing reliable and flexible methods, but also about generating data for studying the interaction between science and technology. For instance, our data can support studies from various perspectives, like author-inventor relation, valuable publication sources in technology development, science dependence of technologies, technology field evolution and so on. We will first introduce a patent-publication network using our matching results. This network is then transformed into a publication co-citation network to reflect co-citation relations in patents. Afterwards, we detect communities in this publication co-citation network. This gives us a way to study how scientific disciplines interact and cooperate in technology development.

8.1 Social network analysis

Social network analysis is meant to uncover the patterns of social structures and explain social phenomena from multiple disciplines [Borgatti et al., 2009]. It is used mainly within the field of sociology, but with the increasing demand of computational capacity and interests in complex network studies, it has become an interdisciplinary technique under the important influence from physics, mathematics and computer science [Otte and Rousseau, 2002].

Social network analysis has wide applications in bibliometrics. In this master thesis, we especially focus on co-citation network studies. Co-citation of publications is a method to measure the rela-

tion between publications [Small, 1973] and the study of the structure of scientific communication [Gmür, 2003]. Co-citation reflects subject similarity and association or co-occurrence of ideas in publications.

In this thesis, we study co-citation of scientific literature in patents. In this case, co-citation can be interpreted as the extent to which scientific publications are jointly used from the perspective of technology development.

8.2 Bipartite network projection

The *bipartite network* is a particular kind of network. The nodes are divided into two sets and each edge can only connect one node in a set to one node in the other set. Edges connecting nodes in the same set are not allowed. This kind of network is also referred as a *two-mode network* because of the two separate sets of nodes. Correspondingly, networks with only one set of nodes are *one-mode networks*. Usually to analyze the two-mode networks, we need to transform it to one-mode networks because most associated methods and techniques for network analysis are only suitable for one-mode networks. As a consequence, we usually transform a two-mode network to a one-mode network using *projection*. The information carried in the two-mode networks is compressed during projection. So, to retain important information, assigning weight to the network connections is necessary. Figure 8.1 gives an example of both unweighted and weighted bipartitie network projection.

The bipartitie network is represented as $B = (U, V, E)$. U and V are two sets of nodes. E represents the edges containing pairs $\{u, v\}(u \in U, v \in V)$. The one-mode network is defined as $G_p(V, E_p, \omega)$. V is the set of nodes from network B. For an edge e_p connecting nodes v_1 and v_2, the weight $\omega(e_p)$ is defined as the number of nodes in the set of U which are connected with both node v_1 and v_2.

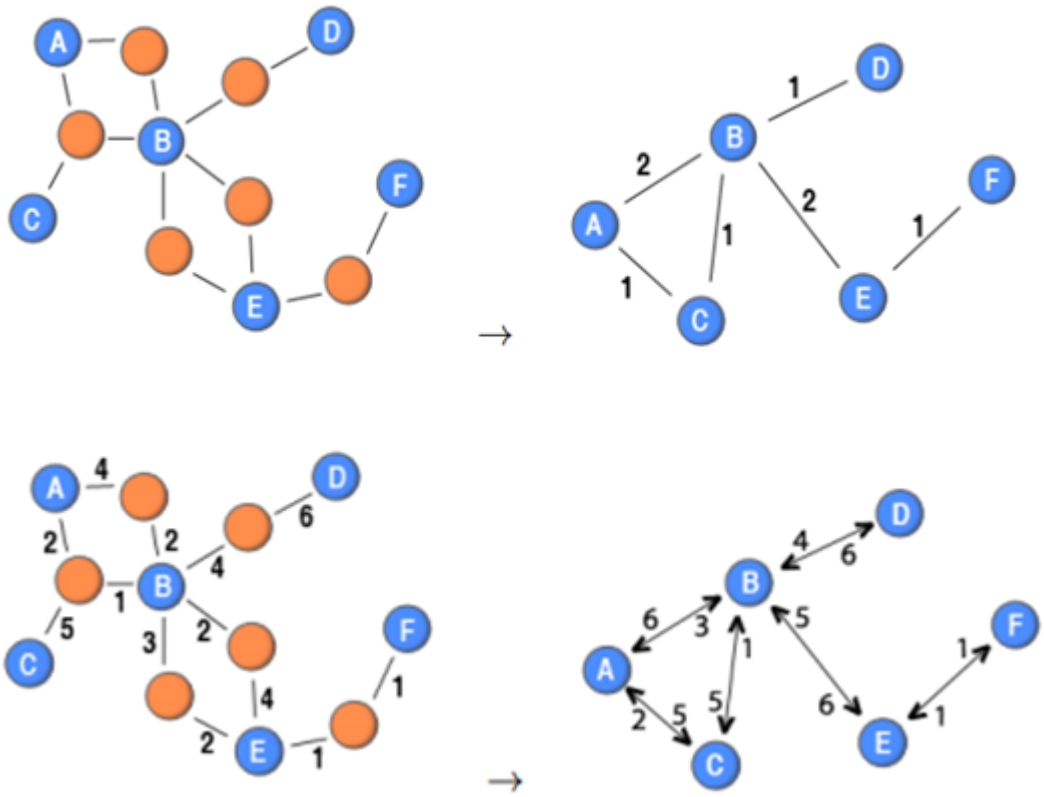

Figure 8.1: Bipartite network projection (`https://toreopsahl.com/tnet/two-mode-networks/projection/`)

8.3 Community detection

In networks, some groups of nodes are more densely connected with each other than with the rest of the network. This feature of network structure indicates that the network has certain natural divisions within it. A group of nodes with dense connections is referred to as a *community*.

The occurrence of community structure in networks is quite natural, but the task of selecting good partitions is not trivial. *Modularity* is a commonly used quality function to assess the division of a network into communities [Newman and Girvan, 2004]. High modularity in networks indicates dense connections between nodes within communities and loose connections between nodes in other communities. Based on this, optimization of modularity is used for network community detection by searching over possible divisions of a network, in order to find a division that has

particularly high modularity. *Resolution* is an important notion in modularity optimization as it is used to control the size and number of communities found. Higher resolutions produce larger amounts of communities, lower resolutions produce lower amounts of communities.

In this thesis, we apply the implementation of the *Louvain algorithm* relying on modularity optimization for community detection. The method consists of two phases. First, it looks for "small" communities by optimizing modularity in a local way. Second, it aggregates nodes of the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained [Blondel et al., 2008].

8.4 Network descriptives

Using the reference-publication matching results we obtained in this thesis, we build a reference-publication network, which is a bipartitie network $B = (U, V, E)$. The patents' literature references and publications in WoS are the two separate sets for the bipartitie network nodes. This reference-publication network indicates which publication is cited in which *patent family* – a set of patent applications or patent publications to protect a single invention. The patent families are represented by the set of nodes U while and the publications are represented by the set of nodes V. The set of edges contain pairs $\{u, v\}(u \in U, v \in V)$ between one patent family and one publication in WoS.

We apply the implementation of the bipartitie network projection explained in Section 8.2 to transform the reference-publication network to a publication co-citation network. The publication co-citation network is an indirect weighted network $G_p(V, E_p, \omega, \phi)$. We keep the same set of nodes V from the reference-publication network to present publications in WoS. The set of edges E_p contain pairs $\{v_i, v_j\}(v_i \in U, v_j \in V \text{ and } v_i \neq v_j)$ between two publications cited together in patent families. We use $deg(v)$ to represent the degree of node v, which refers to how many edges contain node v. For an edge e, the weight $\omega(e)$ is defined as the number of patent families which cite both publications on this edge. The label ϕ represents the scientific categories a publication belongs to. Table 8.1 presents some statistics about this co-citation network properties. The network density in the table describes the portion of the potential connections in a network that are actual connec-

Publication co-citation network	
Nodes (publications)	1,680,972
Edges (co-citation)	44,216,943
Density	3.130×10^{-5}
Average degree	52.609
Connected components	165,700
Giant component	
Nodes (publications)	1,408,691 (72.4%)
Edges (co-citation)	43,989,261 (99.8%)
Density	4.433×10^{-5}
Average degree	62.454

Table 8.1: Publication co-citation network properties

tions. We use N to represent the number of nodes and $|E|$ to represent the number of edges in the network. Then the density is:

$$Density = \frac{2|E|}{|N|(|N|-1)}$$

From the network properties, we can observe that it is a big network with millions of nodes and edges. Figure 8.2 presents the degree distribution of the publication co-citation network. It shows a skewed node-degree distribution following a power-law. It indicates that most publications have only few links but, by contrast, there exist some publications which are extremely active in co-citation as indicated by the heavy tail. These features are similar to the commonly occurring real-world networks [Muchnik et al., 2013].

The giant component contains 72.4% of the nodes but captures almost all the edges (99.8%). It indicates that a majority of publication co-citations are included in the giant component. Figure 8.3 shows the distribution of connected component sizes (the giant component is not included). It is observed that the smaller components tend to appear more frequently, while even the biggest

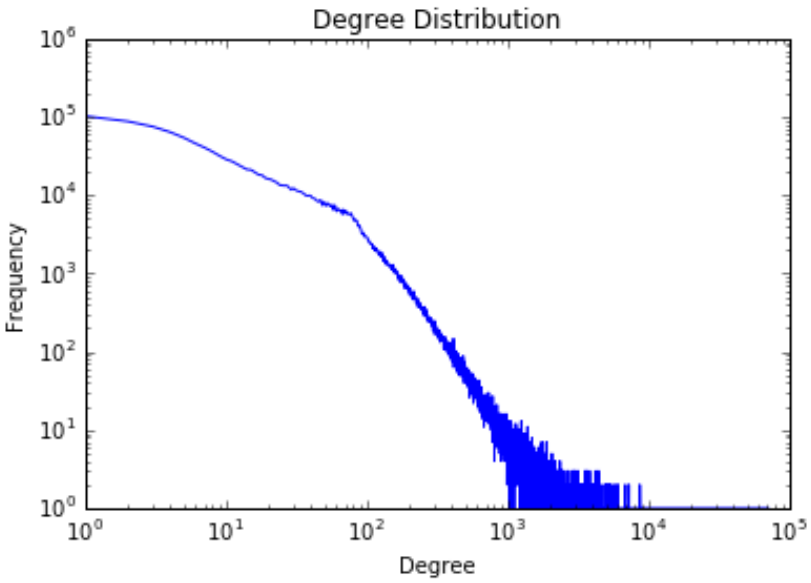

Figure 8.2: Degree distribution

component in this figure is still much smaller then the giant component. As a consequence, given the fact that the giant component covers a significant share of edges and other components are too small to be representative, further analysis and study is performed on only the giant component.

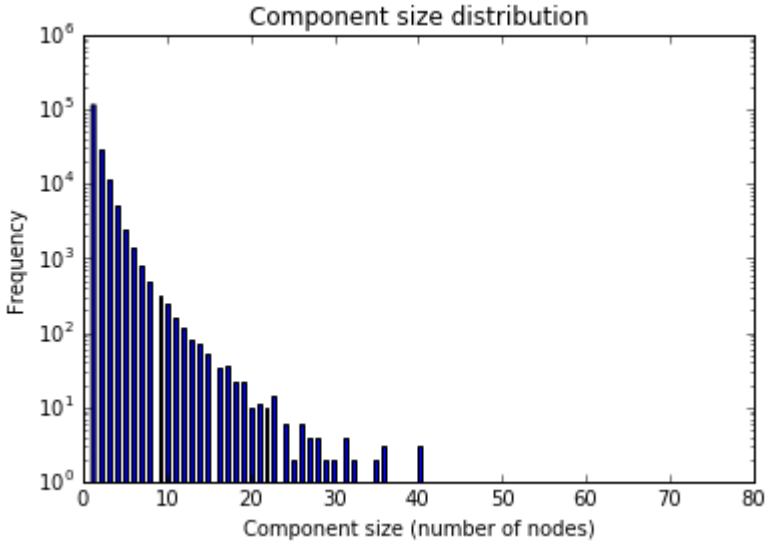

Figure 8.3: Component size distribution

8.5 Community analysis

We implemented the Louvain algorithm to detect communities in the publication co-citation network. Here, we only used the giant component as discussed before and because, in the co-citation network, all the connected components will be taken as separate communities through modularity optimization. The standard resolution parameter in the Louvain algorithm is 1 and we use multiple resolution parameters in the experiment to compare the results. Table 8.2 presents a description of the community detection results.

Resolution	Number of communities	Modularity
0.3	696	0.684
0.4	616	0.692
0.5	544	0.697
1	562	0.708
1.5	779	0.695
2	972	0.669

Table 8.2: Community detection results for varying resolution parameters

It should be noted that, in the implementation of Louvain algorithm, the number of communities becomes lower when the resolution is increased [Lambiotte et al., 2008]. But in Table 8.2, the number of communities and the resolution parameter values do not have a monotonic relationship. For instance, the resolution parameter of value '1' gains more communities then the resolution parameter of value '0.5'. This can be explained because, for the same resolution, we can get a different community partition with different modularity every time we run the community detection algorithm due to the algorithm's inherent randomness. It may happen that for instance a run with resolution parameter of value '0.5' gets a small number of communities with a particular modularity while a resolution parameter of value '1' gets a large number of communities in a certain partition and due to this, the lower resolution parameter obtains less communities.

In this section, we pick the community results obtained by setting resolution parameter as 1, which

produces a quite high modularity and large communities. Figure 8.4 shows the community size distribution in the publication co-citation network. We can observe that there are many communities with a very small amount of nodes and most nodes are included in a small number of larger communities. So, to clear up the visualization, we only keep the communities with 1,000 or more nodes. These are 80 communities, together containing 98.7% of the nodes.

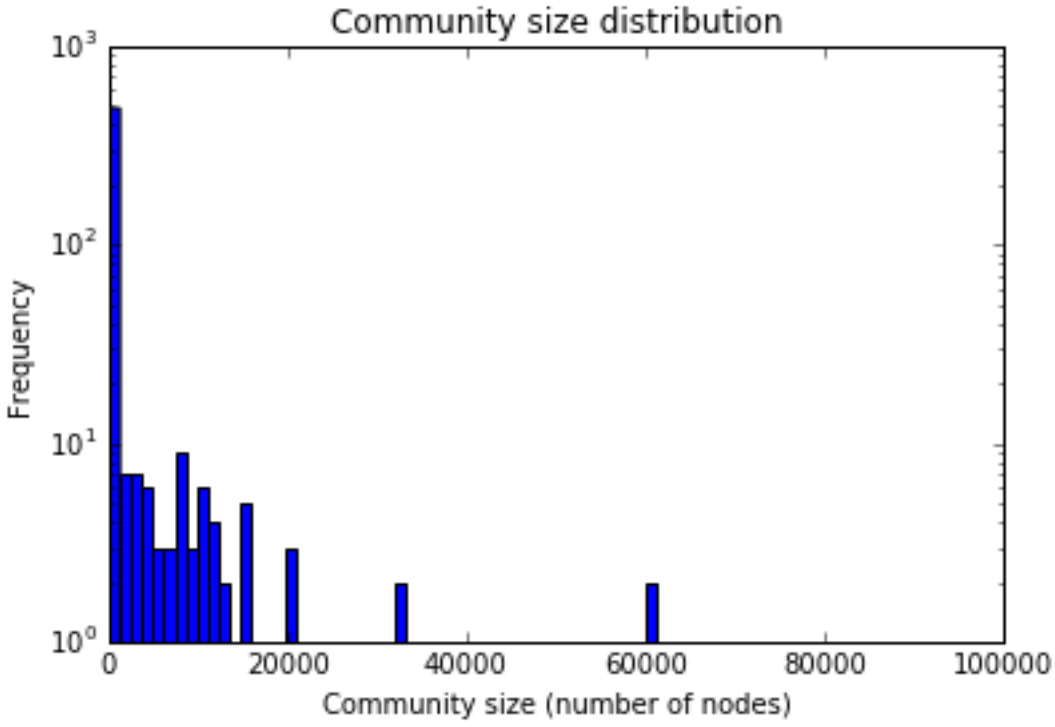

Figure 8.4: Community size distribution

After obtaining communities in the publication co-citation network, we explore the discipline distribution within each community. Here, we use the scientific disciplines used in the phase of publication type cleaning in Section 6.1.2. However, it is noted that around 20% of publications belong to multiple scientific disciplines with the same significance indicated by weight, shown in Table 8.3.

When we analyze the discipline distribution in each community, we take their weight into consideration. For instance, if the publication whose 'UT' is '000070743500002' is in a community, then we count it as one publication in the discipline 'MATHEMATICS, STATISTICS AND COM-

UT	Discipline	weight
000070743500002	MATHEMATICS, STATISTICS AND COMPUTER SCIENCE	1
000070787000010	LIFE SCIENCES	1
000070924700032	CHEMISTRY, PHYSICS AND ASTRONOMY	0.5
000070924700032	LIFE SCIENCES	0.5
000070924700034	CHEMISTRY, PHYSICS AND ASTRONOMY	0.5
000070924700034	LIFE SCIENCES	0.5
000070980800006	MATHEMATICS, STATISTICS AND COMPUTER SCIENCE	0.33
000070980800006	LIFE SCIENCES	0.33
000070980800006	MEDICAL SCIENCES	0.33

Table 8.3: Example of publications with disciplines

PUTER SCIENCE'. But for the publication with UT '000070924700032', it contributes 0.5 for the discipline 'CHEMISTRY, PHYSICS AND ASTRONOMY' and contributes 0.5 for the discipline 'LIFE SCIENCES'. We apply this method when we calculate the total number of publications in a specif discipline as shown in Table 8.4 and the proportion of disciplines in communities in subsequent analysis.

It is noted that the statistics in this table are quite different from the numbers in Table 6.4 in Section 6.1.2. This is because Table 8.4 is built using the reference-publication matching results while Table 6.4 includes all the publications in WoS. In addition, in Table 8.4, the weight indicating to which extent a publication belongs to a discipline is considered. The results of discipline distribution within network communities are present in Appendix B. The discipline distribution of the network communities is shown in Figure 8.5.

To show the discipline division in each network community, we use pie charts in combination with a network structure, as shown in Figure 8.6. This visualization is obtained by merging publications in the co-citation network based on their community. We merge all publications within a specific community into a singe node and we remove all the internal ties connecting publications within

Discipline	Number of publications
MEDICAL SCIENCES	550672.68
LIFE SCIENCES	403633.85
CHEMISTRY, PHYSICS AND ASTRONOMY	293971.77
ENGINEERING SCIENCES	66186.33
MULTIDISCIPLINARY JOURNALS	55394.00
MATHEMATICS, STATISTICS AND COMPUTER SCIENCE	23192.52
EARTH AND ENVIRONMENTAL SCIENCES	10596.43
HEALTH SCIENCES	5043.42

Table 8.4: Number of publications in disciplines cited in patents

the same community. As a result, in the publication co-citation network, only ties connecting publications from two different communities remain. This transforms the publication co-citation network into a community network. However, because of the large volume of publications, there are still a large number of cross-community ties left. The community network is fully connected indicating that each pair of communities is connected.

However, the fully connected community network does not help to understand community relations. So we take the percentage of ties between each two communities into consideration. If the number of ties between two communities is higher than 10% of the size of the larger community (represented by the number of publications), then we retain the ties between these two communities and replace the ties with a single weighted edge indicating the amount of ties divided by the larger community size. Otherwise, we remove the ties and regard the two communities as unconnected.

After obtaining the community network, each community node is substituted by a pie chart indicating the discipline distribution. The radius of the pie charts corresponds to the community size.

From Figure 8.5 and Figure 8.6, we can see that pie charts consist of multiple colors rather than one

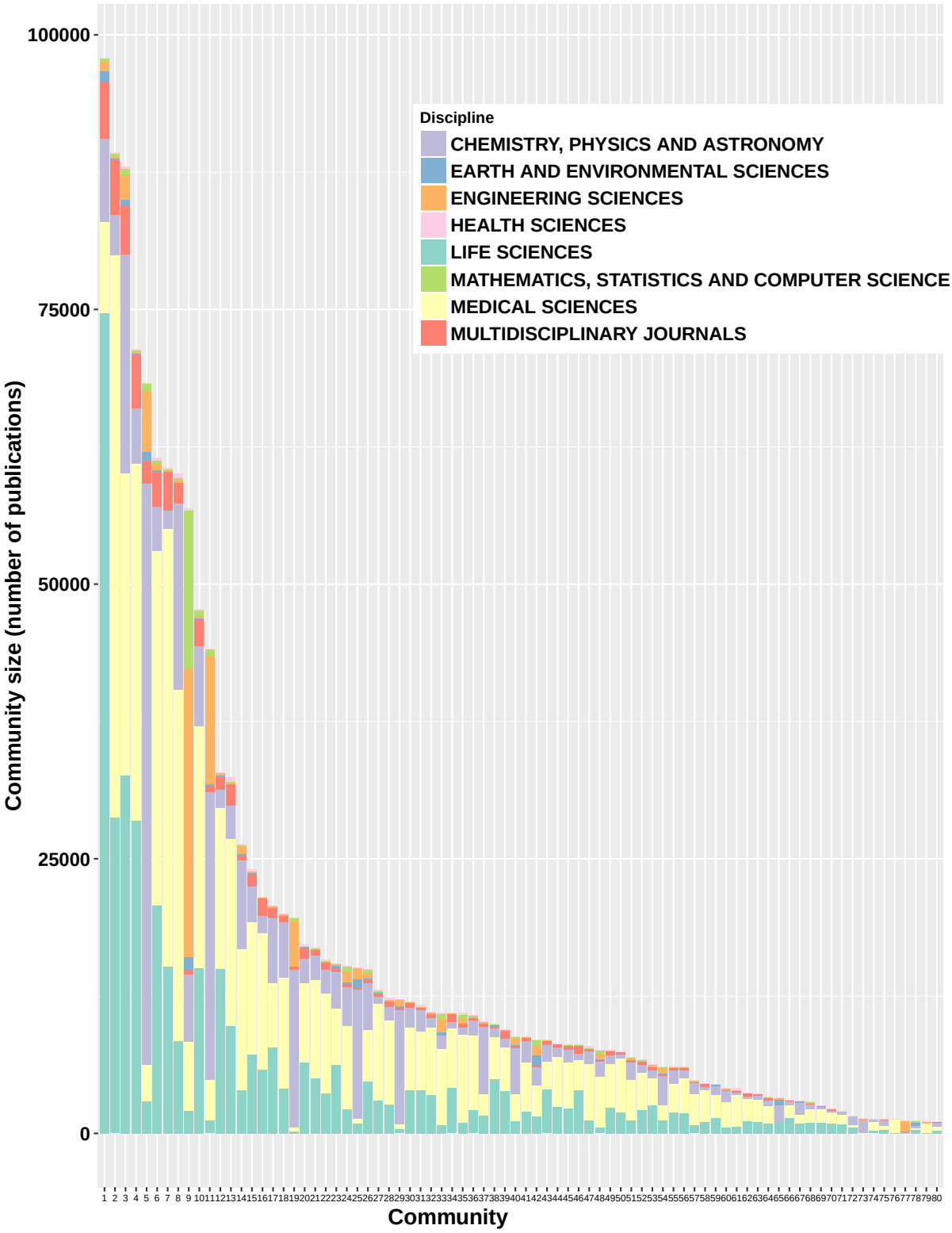

Figure 8.5: Discipline distribution of network communities

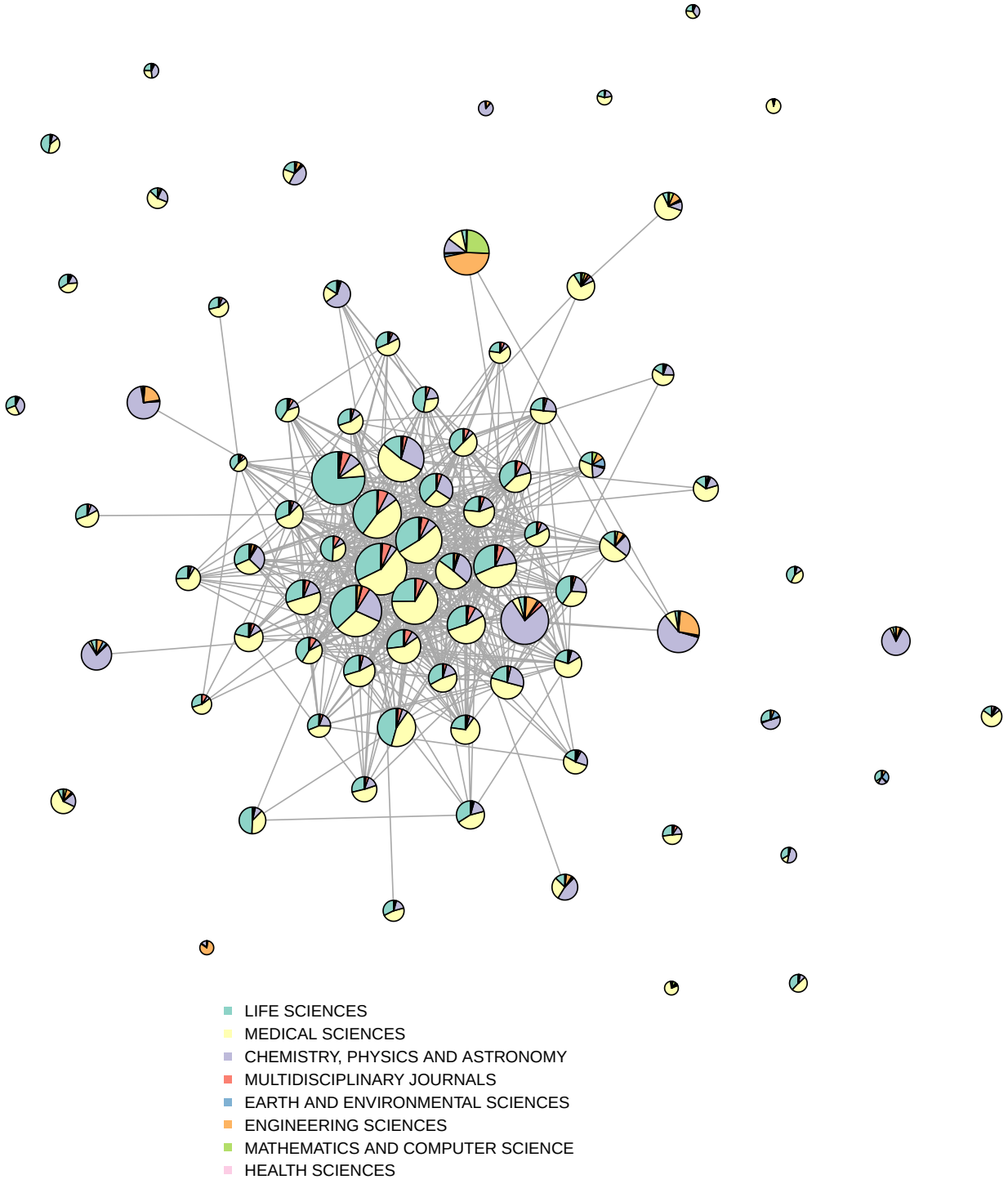

Figure 8.6: Component size distribution

distinguished single color. Our interpretation is that each community features multiple disciplines. This is a remarkable result, because it indicates that the network communities that we find are fundamentally different from the disciplinary classification of publications. Network communities should be regarded more from the perspective of the topology of the network, while the discipline classification is determined by the content of publications. This indicates that there are groups of publications that are similarly cited in technology development, but that belong to different scientific disciplines. It suggests that in technology development, knowledge from various disciplines is combined.

It has to be noted that, in Figure 8.5, parts in yellow and green are quite dominant in most pie charts, which means that the disciplines 'MEDICAL SCIENCES' and 'LIFE SCIENCES' are dominant disciplines in network communities. The number of publications in those two disciplines cited in patents is quite high (see also Table 8.4). It demonstrates that publications in 'MEDICAL SCIENCES' and 'LIFE SCIENCES' are very active in patents. There are several possible explanations: first, from the perspective of science, research development in the discipline 'MEDICAL SCIENCES' and 'LIFE SCIENCES' advances well; second, from the perspective of technology, inventions in or related to this area are dependent on scientific knowledge. The high co-occurrence of the disciplines 'MEDICAL SCIENCES' and 'LIFE SCIENCES' indicates they are highly related. However, their dominant position may partially be due to the large number of publications in these two disciplines, as we note that they are actually the two biggest disciplines in Table 8.4. To exclude the impact of the discipline size, we normalize the discipline size in community. In each community, we divide the number of publications in a particular discipline by the total number of publications cited in patents in that discipline, and use that for the adjusted discipline size of the corresponding pie slice. The discipline distribution with adjusted discipline size is present in Figure 8.7 and Figure 8.8.

Figure 8.8 shows very different discipline distribution in network communities from Figure 8.6. The discipline 'MEDICAL SCIENCES' and 'LIFE SCIENCES' still appear often in the pie charts but looks less dominant. Other disciplines, especially 'MULTIDISCIPLINARY JOURNALS' , are shown more often and in bigger proportion in the pie charts. Overall, with the normalized dis-

cipline size, the discipline distribution within network communities tends to be more even rather than consolidating the dominant position of the discipline ‘MEDICAL SCIENCES’ and ‘LIFE SCIENCES’. It further shows that the interaction between disciplines is very active in technology. Another interesting result is the ninth community in Figure 8.7, which contains 62.02% of all ‘MATHEMATICS, STATISTICS AND COMPUTER SCIENCE’ and 39.64% of all ‘ENGINEERING SCIENCES’ publications cited in patents. This suggests that mathematics publications, unlike publications from other disciplines, do not interact much across disciplinary boundaries when it comes to their use in technology.

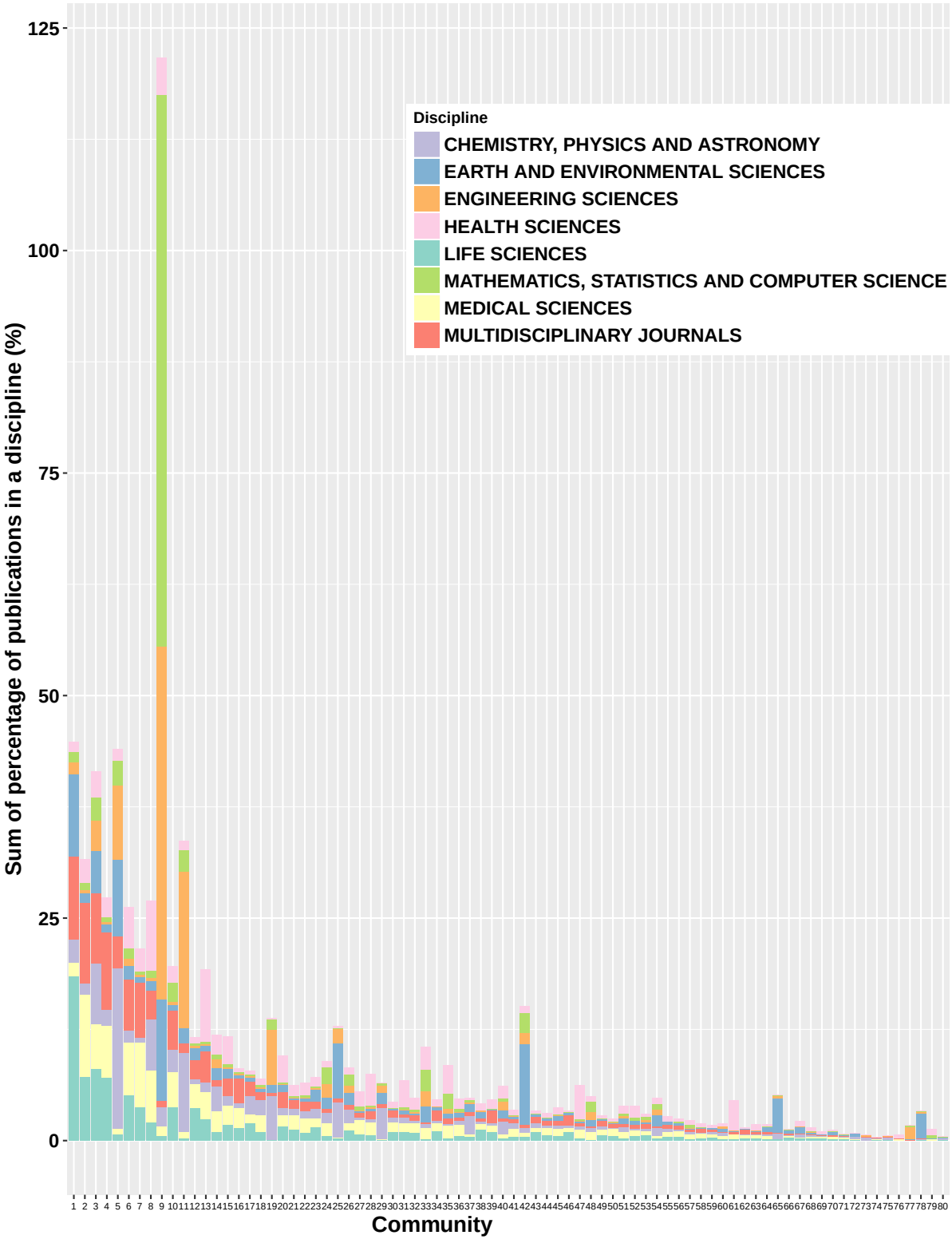

Figure 8.7: Discipline distribution in network communities (percentage of publications in one particular community)

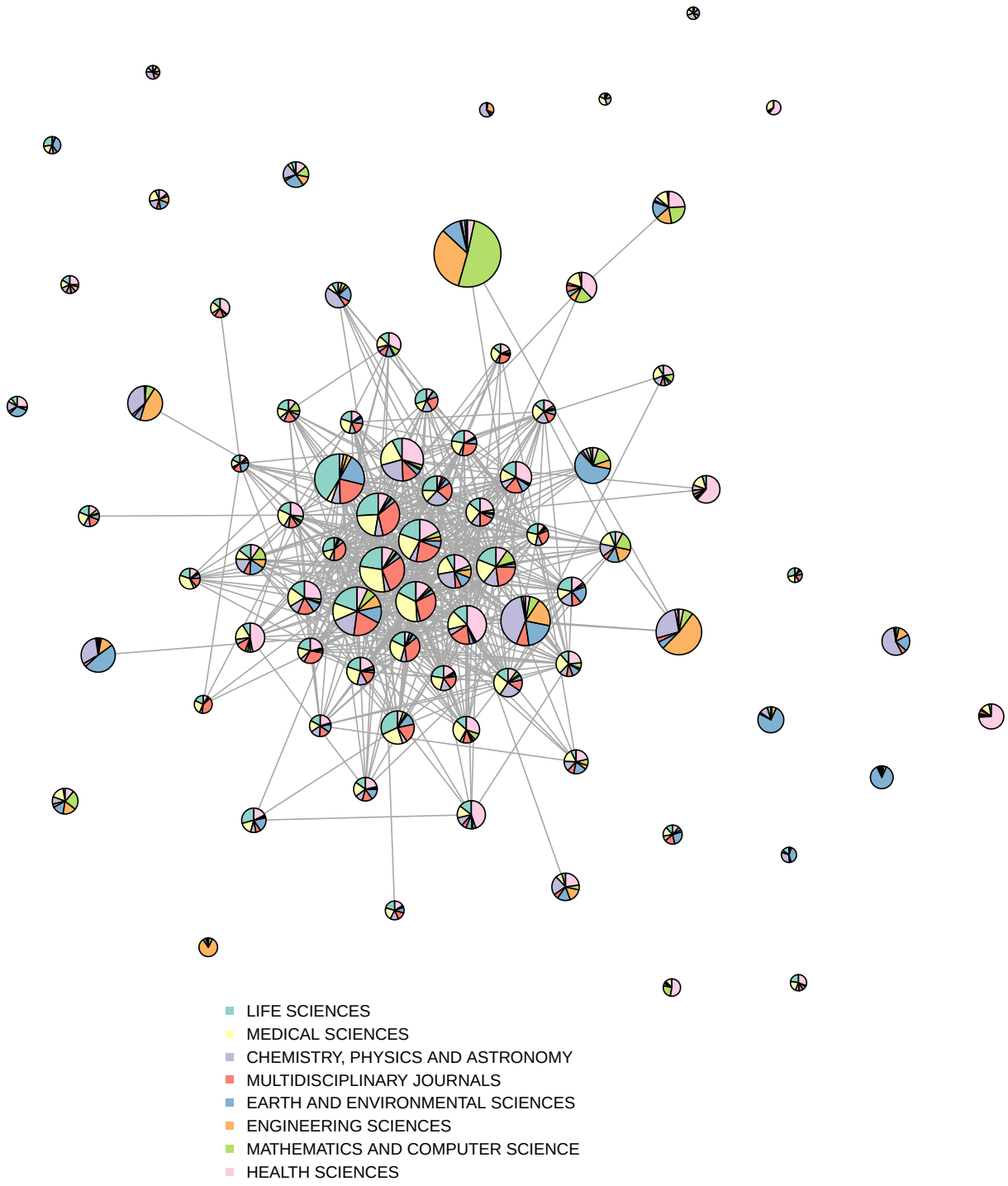

Figure 8.8: Component size distribution (percentage of publications in one particular community)

9 Conclusions and future work

This thesis has been aimed at exploring an approach to automatically identify scientific publications in patents' literature references and is specially focused on matching scientific publications in WoS with literature references from PATSTAT. This started with data description and a comparison of different features of literature references in patents with in scientific publications. We developed a reference-publication matching approach which is different from existing citation matching tools because of its capability to solve the problems brought by poorly structured literature references.

In the matching phase, we described a method for reference parsing and match candidate selection specifically tailored to the data structure and key data attributes in WoS and PATSTAT. The publication attribute patterns collected during reference parsing and the combinations of attributes used in candidate selection are likewise determined by observation and analysis of the available data. The promising matching results reflect the validity and effectiveness of the approach.

In the match candidate refinement phase, we developed the improved fitting alignment algorithm, which is an important contribution of this thesis. The final matching results and the comparison between the improved fitting alignment algorithm and the semi-global alignment algorithm indicate that the improved fitting alignment algorithm can handle fuzzy matching literature references with publication titles very well.

In Section 8, we present one way to use the data for studying the interaction between science and technology. We obtain a publication co-citation network with more than 1.6 million nodes.

It reveals the volume of information stored in the matching results. We believe that the 6.3 million reference-publication matches will open more research avenues and function as database for various research activities.

From the analysis of the results in Section 7.3, we realize that the selection rules in the match candidate selection phase are limited. We suggest to add more combinations of attributes to include more matches for future improvement. In addition, we suggest to introduce fuzzy selection rules. Different from the exact attribute matching, the fuzzy selection rules would work by calculating the intersection of all the numbers in literature references and all the numeric attributes in publications from WoS. It contributes to a less strict selection of more candidates and also allows more flexible formats in references, enabling the method to handle difficult-to-parse references that previously could not be processed correctly by the parsing.

The genericsness is of concern for future work and we suggest to extend the data. Publications from the non-technology related disciplines and all literature references in PATSTAT can be involved. Furthermore, as the approach is based on identification and matching of publication attributes, it is possible to apply this approach generically on other bibliographic databases such as Scopus which usually contains similar publication attributes.

Increased correctness and genericsness demands a bigger volume of data in future work, increasing requirements for the computational capability. We intend to lower the computing complexity by implementing the program in parallel.

Bibliography

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, P10008.

Borgatti, S. P., Mehra, A., Brass, D. J., and Labianca, G. (2009). Network analysis in the social sciences. *Science*, 323(5916):892 –895.

Brooks, H. (1994). The relationship between science and technology. *Science Policy*, 23(5):477–486.

Brusoni, S., Criscuolo, P., and Geuna, A. (2005). The knowledge bases of the world's largest pharmaceuticals groups : What do patent citations to non-patent literature reveal? *Economics of Innovation and New Technology*, 14(0):395–415.

European Patent Office (2014). Data catalog PATSTAT-EPO worldwide patent statistical database. Technical report, European Patent Office.

Fedoryszak, M., Bolikowski, Ł., Tkaczyk, D., and Wojciechowski, K. (2013). *Methodology for evaluating citation parsing and matching*, pages 145–154. Springer Berlin Heidelberg, Berlin, Heidelberg.

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). Citeseer: an automatic citation indexing system. In *International Conference on Digital Libraries*, pages 89–98. ACM Press.

Gmür, M. (2003). Co-citation analysis and the search for invisible colleges: A methodological evaluation. *Scientometrics*, 57(1):27–57.

Guner, S. (2015). Disambiguation of scientific references in a patent database: a project to facilitate economic research and policy evaluation. Bachelor thesis, Erasmus University Rotterdam.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2008). Laplacian dynamics and multiscale modular structure in networks. e-print, arXiv:0812.1770.

Lammey, R. (2014). CrossRef developments and initiatives: an update on services for the scholarly publishing community from CrossRef. *Science Editing*, 1(1):13–18.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Meyer, M. S. (2001). Patent citation analysis in a novel field of technology: An exploration of nano-science and nanotechnology. *Scientometrics*, 51(1):163–168.

Muchnik, L., Pei, S., Parra, L. C., Reis, S. D. S., Andrade, Jr., J. S., Havlin, S., and Makse, H. A. (2013). Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Scientific Reports*, 3:1783.

Narin, F. and Noma, E. (1985). Is technology becoming science? *Scientometrics*, 7:369–381.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Netherlands Observatory of Science and Technology NOWT (2001). Science and technology indicators 2010. Technical report, Ministry of Education, Culture and Science.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113.

Otte, E. and Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *JOURNAL OF INFORMATION SCIENCE*, 28(6):441–453.

Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science and Technology*, 24(4):265–269.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.

The Science Council (2009). Our definition of science. http://sciencecouncil.org/about-us/our-definition-of-science/. [Online; accessed 18-May-2016].

Thomson Reuters (2013). Web of Science- quick reference guide. Technical report, Thomson Reuters.

Tijssen, R. J. W. (2001). Global and domestic utilization of industrial relevant science: patent citation analysis of science-technology interactions and knowledge flows. *Research Policy*, 30:35–54.

Tijssen, R. J. W., Buter, R., and van Leeuwen, T. (2000). Technological relevance of science: An assessment of citation linkages between patents and researh papers. *Scientometrics*, 47(2):389–412.

Verbeek, A., Debackere, K., Luwel, W., Petra, A., Zimmermann, E., and Deleus, F. (2002). Linking science to technology - bibliographic references in patents to build linkage schemes. *Scientometrics*, 54(3):399–420.

World Intellectual Property Organization (2015). Finding technology using patents. Technical report, World Intellectual Property Organization.

Appendix A

Stop words in publication source name matching

a
about
after
an
and
at
before
for
from
in
into
of
off
on
out
over
the
to
with

Table A.1: Stop words

Appendix B

Discipline distribution in network community

Table B.1 present the disciplines in WoS. We give each discipline an ID to make it easy for display. Table B.2 presents the publication distribution in each discipline for the first 80 biggest network community.

Discipline ID	Discipline
Dis1	LIFE SCIENCES
Dis2	MEDICAL SCIENCES
Dis3	CHEMISTRY, PHYSICS AND ASTRONOMY
Dis4	MULTIDISCIPLINARY JOURNALS
Dis5	EARTH AND ENVIRONMENTAL SCIENCES
Dis6	ENGINEERING SCIENCES
Dis7	MATHEMATICS, STATISTICS AND COMPUTER SCIENCE
Dis8	HEALTH SCIENCES

Table B.1: Disciplines in WoS with discipline ID

Total	Dis1	Dis2	Dis3	Dis4	Dis5	Dis6	Dis7	Dis8
97923	74657.07	8282.15	7625.57	5152.0	984.15	913.75	250.65	57.67
89296	28786.17	51130.42	3714.17	4990.0	118.75	254.08	168.25	134.17
87991	32561.92	27543.08	19956.33	4412.0	499.67	2270.67	599.08	148.25
71415	28514.33	32446.92	5103.83	4840.0	93.58	177.25	128.25	110.83
68297	2926.0	3294.5	52944.33	2011.0	914.5	5489.33	653.67	63.67

61467	20809.5	32202.17	4057.33	3180.0	156.5	569.83	259.5	232.17
60570	15199.33	39812.83	1709.83	3410.0	71.33	139.83	99.67	127.17
60050	8416.23	31988.57	16962.9	1754.0	122.23	237.17	172.57	396.33
56919	2056.67	6274.5	6124.17	428.0	1204.5	26236.33	14385.0	209.83
47707	15037.07	22025.07	7308.4	2448.0	63.07	221.17	509.9	94.33
44083	1227.33	3633.92	26240.83	563.0	187.25	11632.58	544.75	53.33
32865	14988.5	14665.33	1616.17	1202.0	139.83	140.0	81.17	32.0
32390	9752.83	17081.58	2992.17	1915.0	73.42	106.42	61.25	407.33
26334	3931.0	12850.33	8119.0	395.0	148.33	667.0	110.33	113.0
24048	7199.17	12026.5	3296.67	1045.0	115.33	126.5	83.0	155.83
21546	5801.33	12457.42	1564.5	1534.0	41.08	73.42	53.25	21.0
20749	7847.5	5843.67	5919.67	884.0	49.83	143.33	41.5	19.5
19994	4081.67	10096.5	5079.67	499.0	39.5	89.17	72.5	36.0
19608	190.83	354.67	14374.33	186.0	94.0	4144.33	259.0	4.83
17209	6440.17	7209.5	2236.5	984.0	91.5	61.0	33.67	152.67
16906	5003.33	9000.83	2198.5	513.0	21.67	61.67	47.17	59.83
15786	3685.0	9099.25	2126.33	624.0	32.92	87.75	58.42	72.33
15475	6258.7	5164.53	3258.2	420.0	147.37	134.83	40.87	50.5
15238	2218.83	7582.17	3546.0	261.0	126.33	1042.83	426.83	34.0
15104	921.83	445.42	11765.67	235.0	660.58	1026.92	39.08	9.5
14966	4733.08	4673.08	4247.5	319.0	144.33	521.0	287.58	40.42
13087	2985.2	8832.2	641.03	327.0	13.87	97.0	106.53	84.17
12381	2642.67	7676.83	1189.5	492.0	24.5	126.5	51.67	177.33
12215	423.53	412.2	10405.87	212.0	132.03	578.83	50.03	0.5
12008	3949.83	5695.5	1823.17	424.0	15.0	47.67	16.83	36.0
11723	3979.5	5295.25	1934.33	202.0	50.58	58.25	50.25	152.83
11072	3487.33	6203.75	809.83	324.0	24.08	90.08	68.42	64.5
11016	771.83	6915.25	1208.0	107.0	188.25	1139.58	557.25	128.83

10980	4177.33	5381.33	614.83	676.0	33.83	42.0	16.5	38.17
10966	966.0	8004.67	671.5	325.0	53.5	408.83	375.33	161.17
10747	2164.42	6791.25	1315.83	209.0	55.17	73.83	81.92	55.58
10196	1618.67	1953.42	6123.5	213.0	99.25	118.58	58.25	11.33
9950	4913.5	3822.33	864.67	149.0	85.0	61.67	16.5	37.33
9487	3916.67	3913.75	758.33	787.0	14.42	21.42	25.58	49.83
8836	1126.83	2508.42	4115.83	188.0	100.58	642.42	83.08	70.83
8816	2026.83	4493.83	1878.17	266.0	26.17	68.33	29.17	27.5
8538	1609.33	2744.08	1651.17	189.0	961.25	824.58	521.25	37.33
8494	4027.5	2533.0	1471.67	400.0	34.17	6.67	5.5	15.5
8167	2463.03	4485.03	843.2	288.0	27.53	19.83	17.87	22.5
8068	2319.83	4145.08	1148.33	290.0	60.08	49.75	14.42	40.5
8055	3975.17	2690.25	638.67	675.0	22.58	28.08	17.42	7.83
7958	1199.67	5131.17	1139.5	176.0	23.0	62.83	33.0	192.83
7589	567.0	4587.0	1369.83	116.0	76.0	561.33	283.83	28.0
7579	2386.83	3921.33	811.33	376.0	11.67	45.17	12.17	14.5
7450	1922.67	4888.33	361.33	202.0	17.33	33.33	8.5	16.5
6964	1189.0	3683.67	1558.5	195.0	74.17	142.67	79.17	41.83
6704	2111.0	3424.25	704.83	223.0	46.75	46.92	83.08	64.17
6295	2563.83	2483.0	680.0	306.0	24.17	122.33	98.83	16.83
6077	1176.5	1387.0	2711.33	81.0	134.17	392.83	161.0	33.17
6065	1884.17	2651.83	1187.5	242.0	34.5	31.67	5.33	28.0
6060	1844.0	3179.75	695.83	238.0	24.08	26.42	34.75	17.17
4836	787.37	2823.7	951.53	102.0	16.53	58.33	70.53	26.0
4576	1045.17	2872.33	337.67	250.0	6.83	24.83	20.83	18.33
4464	1445.17	2094.5	732.17	131.0	18.0	21.33	7.83	14.0
4108	536.0	2298.67	986.67	66.0	34.5	152.33	19.33	14.5
4105	622.33	2886.75	242.67	120.0	10.58	41.08	12.08	169.5

3688	1105.33	2036.17	209.0	320.0	5.0	2.33	4.33	5.83
3640	1043.87	2055.7	325.2	158.0	10.2	8.33	4.7	34.0
3300	900.67	1619.83	470.33	177.0	51.33	47.33	23.5	10.0
3247	942.5	55.0	1581.17	62.0	411.0	186.5	6.83	2.0
3043	1426.17	1168.17	333.67	55.0	46.0	10.5	0.5	3.0
2983	920.33	788.5	1052.5	49.0	74.33	62.5	7.33	28.5
2859	972.0	1203.83	473.83	76.0	12.5	89.67	13.17	18.0
2536	964.83	1239.5	245.67	52.0	8.0	7.67	1.83	16.5
2245	894.67	1006.92	133.17	112.0	32.25	44.75	13.25	8.0
2034	867.0	849.0	228.67	64.0	7.5	12.33	2.0	3.5
1616	541.7	220.7	786.37	17.0	37.53	6.83	5.87	0
1353	8.5	19.0	1168.17	13.0	4.5	134.5	5.33	0
1318	278.67	749.67	257.33	25.0	0.5	5.33	0.5	1.0
1313	323.33	354.33	533.83	45.0	6.67	44.83	2.5	2.5
1301	40.5	1210.0	13.83	7.0	1.0	4.5	3.17	21.0
1140	2.0	4.08	152.0	10.0	9.58	936.08	25.75	0.5
1112	387.17	72.83	224.5	21.0	301.83	83.5	21.17	0
1089	28.5	856.67	42.67	3.0	4.0	40.5	78.0	35.67
1066	251.83	387.33	354.5	28.0	7.0	35.83	1.0	0.5

Table B.2: Overview of network communities