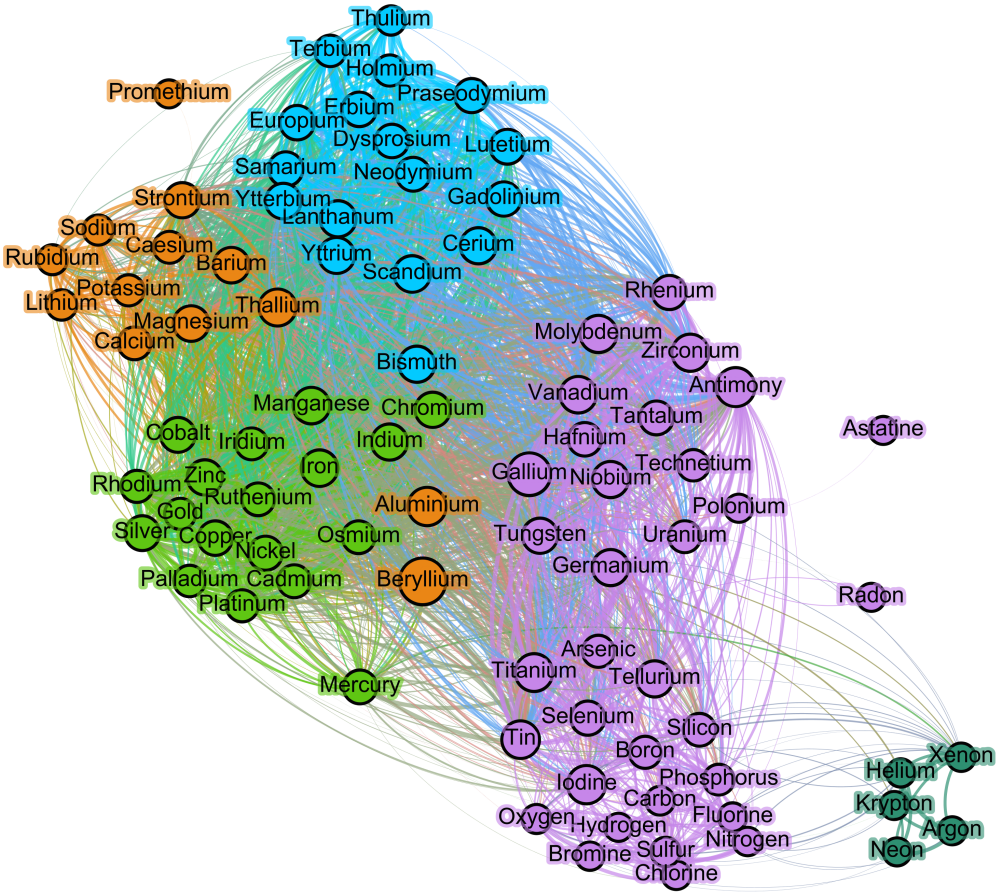
Part III Project

Automatic Chemical Analysis - Big Data and Machine Learning in Online Chemistry Literature



Patrick Lewis

Queens' College

March 2016

This dissertation is submitted in partial fulfilment of the requirements for Part III Chemistry. It describes work carried out in the Department of Chemistry in the Michaelmas Term 2015 and the Lent Term 2016. Unless otherwise indicated, the research described is my own and not the product of collaboration.

Signed:

Date:

Acknowledgements

Acknowledgements will go here

Abstract

A Large Dataset of Chemistry Literature meta-data was built up by automated scraping from freely-available online sources. A UK Chemistry Department dataset of chemical literature meta-data was built up by a similar method. Novel Natural Language Processing algorithms were used to develop powerful models to represent the chemical semantic space. These models were analysed and visualisation techniques were developed. The utility of the models was demonstrated by investigating relationships between researchers at the University of Cambridge Chemistry Department.

Contents

List of Figures

Glossary

$\Delta 1$ Database of article meta-data created from DOIs found on UK Chemistry department websites.

$\Delta 2$ Database of complete article meta-data including abstracts found on UK Chemistry department websites.

$\Delta 3$ Database of article meta-data created from DOIs found in global scraping procedure.

$\Delta 4$ Database of complete article meta-data including abstracts found in global scraping procedure.

$\Delta 5$ Database created by combining databases $\Delta 2$ and $\Delta 4$.

$\Delta 6$ Database comprised of records in database $\Delta 5$ deemed suitable for machine learning, i.e. sufficiently long titles and abstracts and predominantly ascii characters.

$\Delta 7$ Database comprised of records in database $\Delta 6$ which had originated from DOIs found on the University of Cambridge Chemistry Department.

ACS American Chemical Society. Scientific society specialising in chemistry domain with large scientific academic publishing arm. ACS also rusn the Chemical Abstracts Service and SciFinder®.

API Application Programmer Interface. Set of well defined input and output operations to a program or service to enable programmers to easily use the service.

Bag Of Citations Simple Document Representation model which attempts to represent a document based on presence/absence of citations.

Bag Of Words Simple Document Representation model which attempts to represent a document as a vector based on presence/absence of words.

CBOW Continuous Bag Of Words. Learning architecture used by Word2Vec algorithm. Word Vector predictions are made from sum or mean of surrounding context words.

Cluster Map A heat-map of two-dimensional data with axes arranged by a hierarchical clustering algorithm, often overlaid with dendrograms along each axes to illustrate clustering and spatial relationships [1] [2] [3].

Communities Subset of documents in a corpus identified via the Blondel-Guillaume-Lambiotte-Lefebvre modularity algorithm [4][5].

Corpus In the field of language processing, a corpus is a large body of natural language text. In the context of the project, a corpus is the combined titles and abstracts of all the article records in a database.

Cosine Similarity A similarity metric for vectors derived from the angle between them. $S_{cosine} = \cos(\theta)$ for angle θ between two vectors.

Crawling Programming technique to automatically navigate through the online landscape identifying candidate websites for scraping (See Scraping).

Crossref Organisation promoting inter-publisher cooperation with a mission statement to 'support ... persistent, sustainable infrastructure for scholarly communication'[6]. Crossref provides tools for accessing a wide range of publishers' materials.

Dendrogram Tree diagram used to illustrate relationships between clusters produced in hierarchical clustering procedures [1].

Doc2Vec Gensim implementation of Paragraph Vectors algorithm.

DOI Digital Object Identifier. Unique identifier string used to index the vast majority of academic literature articles published since 2000 [7].

Euclidean Similarity A similarity metric for vectors derived from their distance in euclidean space. $S_{euclid} = \sqrt{\sum_{i=1}^{D}(\nu_i^{\alpha} - \nu_i^{\beta})^2}$ for vectors α and β.

Gensim Open-source library for Python programs for use in NLP applications.

Gephi An open-source network visualisation, rendering and analysis application.

Hadamard Division Element-wise division matrix operation defined as $(\mathbf{A} \oslash \mathbf{B})_{i,j} = \mathbf{A}_i / \mathbf{B}_j$ for matrices $\mathbf{A}$ and $\mathbf{B}$.

Hadamard Square Root Element-wise square root matrix operation defined as $\left(\mathbf{A}^{\circ\frac{1}{2}}\right)_{i,j} = \mathbf{A}_{i,j}^{\frac{1}{2}}$ for matrix $\mathbf{A}$.

HTML Hypertext Markup Language. Tag-based language to encode web-pages in a hierarchical structure. Webpages are written as HTML files, which are interpreted by internet browsers to display the page's content to users.

Hyperparameters Adjustable parameters used by a Machine Learning algorithm. Distinct from internal parameters automatically learnt by the algorithm.

IDF International DOI Foundation. Independent not-for-profit body governing use and management of the DOI system. Provide definitive service for resolving DOIs [8].

IP Address Internet Protocol Address. An IP address is the identifier for any computer or device using a network that runs on Internet Protocol.

Lancaster Stemming algorithm [9].

Machine Learning Field of computer science with the aim of developing algorithms that automatically improve performance based on supplied examples [10].

Meta-data Meta-data refers to data about data. In the context of this project, it refers to data describing a chemistry article, i.e. title, abstract, DOI (See DOI), authors, affiliations, journal, publisher and date of publication.

MongoDB Schema-less 'NoSQL' database used for document storage and retrieval.

Neural Net Data structure capable of developing decision pathways using supplied examples.

NLP Natural Language Processing. NLP is the field of linguistics and computer science with the aim of processing human written (natural) language with a computer.

Paragraph Vectors NLP algorithm based on Word2Vec for generating representation vectors for documents.

PCA Principle Component Analysis. Well established technique for reducing dimensionality by a series of orthogonal transformations [11].

PILA Publishers International Linking Association, Inc. . Independent not-for-profit body comprised of scientific prublishing entities. PILA operates Crossref [6].

Porter Early, widely used stemming algorithm [12].

Python Interpreted, dynamically typed programming language. Unless explicitly mentioned, Python was used for all development and analysis in this project.

REGEX REGular EXpression. Text string that is used to inform a programming language of patterns to identify in a body of text.

RSC Royal Society of Chemistry. British learnèd society for chemical sciences with an academic publishing arm.

SciFinder® Bibliographic and citation search engine provided by the American Chemical Society designed for chemical research.

Scraping Programming technique to automatically extract data from online resources.

Skipgram Learning architecture used by the Word2Vec algorithm. Word vector predictions are made from random comparison between a word and nearby context words.

Snowball Recent stemming algorithm [13] (Also known as Porter2). Snowball is also the name of a programming language developed for stemming algorithms.

Stemmer An algorithm used to relate derived words (e.g. plurals, conjugated verbs) to their roots.

Stop Words Words removed from a corpus before being processed. Stopwords are very common and/or do not encode significant information content.

Taylor & Francis Part of the Informa group. An academic publisher covering a range of scientific disciplines.

TF-IDF Term-Frequency Inverse-Document-Frequency. Method for assigning weights to words in a document for how much information the word carries.

Training Epoch A complete iteration over the training data available to a learning algorithm.

TSNE T-distributed Stochastic Network Embedding. State-of-the-art technique for reducing dimensionality. Preserves spatial clusters at high dimensions [14].

Unicode Standard for encoding characters used in worldwide communication. The Unicode character set of 120,000 characters includes mathematical symbols, punctuation, and character languages (Mandarin, Japanese etc.).

UPGMA Unweighted Pair Group Method with Arithmetic Mean. Pairwise Clustering algorithm that partitions a set into hierarchical sub-set clusters using mean distances between pairs of elements [15].

UTF-8 Universal Coded Character Set + Transformation Format 8-bit [16]. An encoding specification for the Unicode character set. Each character is encoded by 8 bits. UTF-8 is the dominant encoding used online [17].

Web Of Science ™ Bibliographic and citation search engine provided by Thomson Reuters.

Word2Vec Sophisticated distributed word vector model utilising a neural net to learn vector representations of component words in a corpus by training sentence by sentence [18][19] .

WordNet Lemmatizing stemming algorithm, based on consulting database of groups of semantically connected word concepts [20] [21] [22].

XML eXtensible Markup Language. Tag-based markup language, closely related to HTML. Enables encoding any type of data in a manner that is machine readable and intelligible by humans.

XPath Query method for extracting data from XML documents. As HTML is closely related to XML, XPath strings can be used to access data in HTML documents.

Zipf's Law States that the relationship of the log of the rank of a word to the log of the frequency of that word in a large corpus of text approximates a directly proportional relationship [23].

1. Introduction

1.1 Modern Scientific Publishing

The widespread adoption of the internet in the late 1990s and 2000s brought fundamental changes to the academic publishing landscape. The information revolution allowed publishers‘ costs to fall, and there was a mood shift in the academic sphere away from subscription-based models, towards giving open and free access to some or all of journal article contents. Simultaneously, learnèd institutions (such as university websites) began to post records of recent publications and other chemical information freely online. Publishers still protect the vast majority of journal article content and some meta-data. Data is valuable, and the insights within, powerful. As such, publishers are unwilling to grant free access to their data, preferring to perform in-house analysis. Article meta-data, such as authors, titles and abstracts may, however, be available, and it is this dataset which the project is focussed on.

1.2 Motivation

By collecting meta-data on papers found on the internet, a large representative dataset of chemical academic writing language can be built. Machine Learning techniques can be applied to find novel connections between articles, research communities, authors, institutions and fields. Machine Learning is a rapidly progressing field and data science can reveal key, non-obvious relationships to aid the scientific process. In an increasingly data-dense world, scientists require smarter tools to streamline research in order to be more productive. Several publishers provide services that perform large-scale analysis and provide literature tools, such as SciFinder® and Web of Knowledge™. The techniques used and motivations behind the corporate bodies that own these services are not necessarily clear, and thus there is much to be gained from independent, original analyses of the online publishing landscape.

1.3 Aims

The aims of the project are set out below:

- Collect large quantities of article meta-data from articles pertaining to chemistry as a general discipline
  - Identify websites that might contain useful chemical information
  - Write web-scraping programs that can scrape to identify and extract chemical information
  - Store information in human readable, computer readable, scalable and stable formats
- Develop novel machine learning techniques to enable meta-data to be interpreted in new ways
  - Sanitise input data effectively
  - Devise machine learning models to interpret article titles and abstracts to attempt to extract their chemical meaning
  - Quantitatively represent an article's content using its collected meta-data
- Validate the model and provide evidence of their efficacy
  - Develop visualisation techniques for interpretation of algorithm output
  - Analyse datasets using the developed model to demonstrate new and useful information
  - Provide usable code with which future analysis may be performed

This project is thus an informatics/data project, which split naturally into two sections. The first half of the project was concerned with acquiring data. This is covered in detail in §2. Programs were written in the Python programming language, and two databases were created (one of UK Department chemistry, and a very large database of unrestricted chemistry-related material).

Once the databases were set up, focus was shifted to how to use the data to find valuable insights. §3 and §4 provide the background of the algorithms used and the process of applying them to create useful models.

Having built the models, it was now necessary to examine their outputs and develop methods to interpret results, covered in §5. Finally, when the models were shown to be performing successfully, they were used in an analytical setting to examine relationships between authors and research communities in the University of Cambridge Chemistry Department, and eventually to recommend specific collaborations between staff.

2. Data Acquisition

2.1 Background

2.1.1 HTML and Xpath

Internet webpages are written in HTML[1]. When a webpage is accessed, the HTML code is sent to the user, and the browser processes and displays the webpage in a human-readable format.

A scraping program must process the raw HTML file and access the useful information on the page in an automated fashion. Information is arranged in an HTML document in a tree-like structure (figure 2.1). This example page would display as a table with three rows, each row containing 'Table Data A/B/C'. This data is accessible programmatically using an 'XPath'.

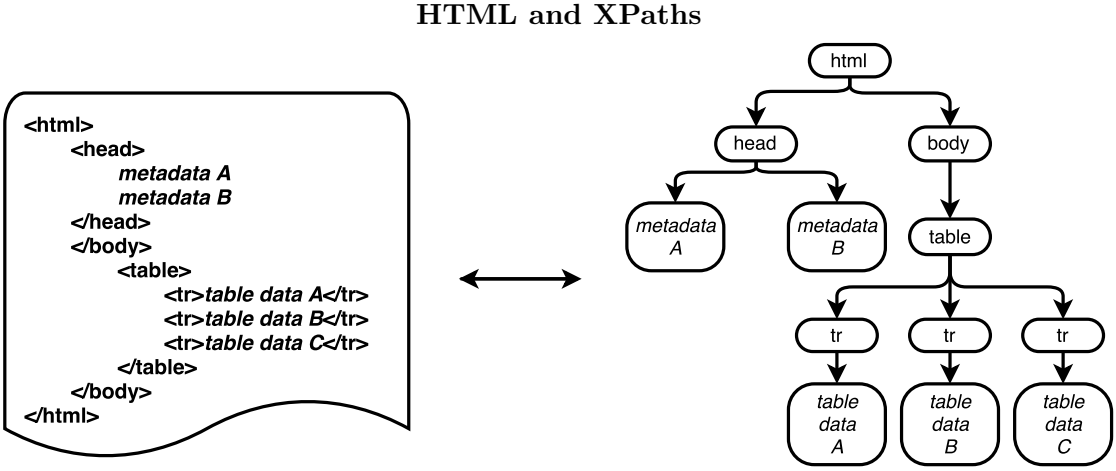


Figure 2.1: Tree representation of HTML code. The html code here displays a table with three rows. The page has two pieces of meta-data associated with it, stored in the 'head'.

[1]See Glossary

XPaths are simply paths through the tree to the desired information. In order the 'scrape' the data in the table, the following XPath could be used:

```
//html/body/tr/*
```

Scraping millions of webpages potentially requires millions of different XPaths. It is impractical to specify them manually, thus the challenge of large-scale scraping is how to identify and collect useful data on webages without manually specifying many XPaths.

2.2 Automatic XPath Generation

The initial approach was to analyse the HTML tree to automatically recognise useful tabulated or listed data. The program started at the tree's root and repeatedly followed the branch with the most 'repeating substructure'. The recursive algorithm is summarised below:

1. `Count # of descendents of each child node`
2. (a) `Calculate the pairwise similarities between all child nodes`
   (b) `Consider two nodes similar if pairwise similarity is above a heuristic threshold`
   (c) `Calculate proportion of nodes that are considered similar`
3. `If proportion calculated in (c) is above a heuristic threshold, this node represents a store of information, and the XPath has been found. Otherwise, move to child node with highest # of descendants, return to step (1)`

The heuristic thresholds are adjustable parameters. The approach was successful for webpages with large numbers of records, formatted in repeating fashion, but performed poorly for smaller collections of data. As such it was not sufficiently flexible for the task of scraping large quantities of chemical data, and was not developed further.

2.3 Collection Strategy

The initial approach was to analyse the HTML tree to automatically recognise useful data generate XPaths[2]. When this strategy proved unsuitable, a new method was required. Chemical information is usually disseminated as journal articles accompanied by a DOI. By programmatically collecting DOIs, (§2.3.1) it was possible to build up a large database of chemical information (see §2.3.2)

[2]This approach is detailed in the appendix, §8.8

2.3.1 DOIs : Document Object Identifiers

DOIs are computer-friendly labels for articles. DOIs are issued by a number of accredited bodies, with the majority issued by Crossref [6]. By pre-pending a DOI with the url stub `http://dx.doi.org/`, The IDF service redirects the request to the publisher's website to display the article the DOI refers to. The DOI structure is shown in Figure 2.2.

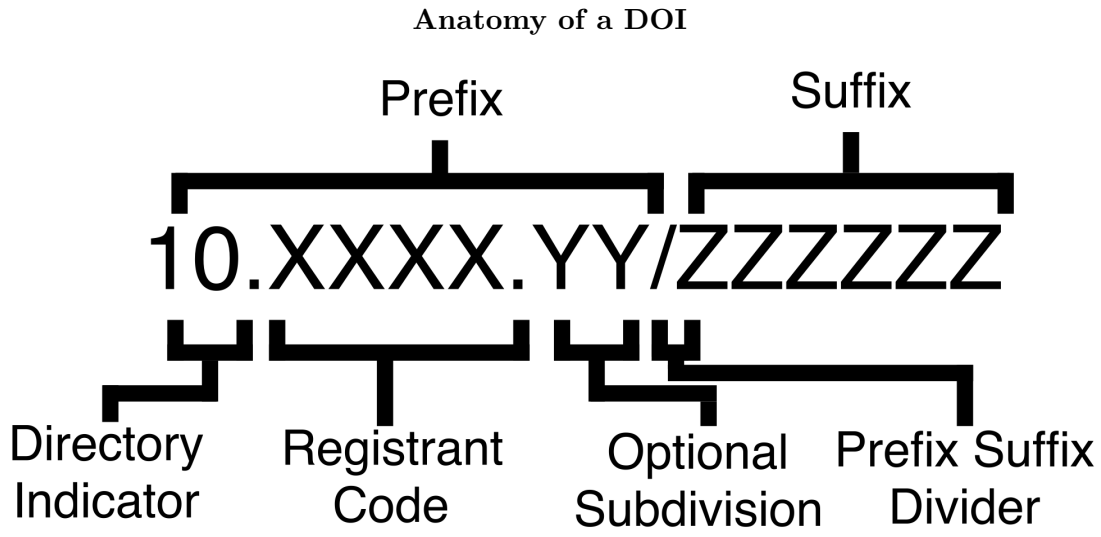


Figure 2.2: DOI structure. The structure consists of a numeric prefix (X and Y must be integers) and alphanumeric suffix (Z can be any Unicode-encoded character)

DOIs consist of a prefix and a suffix. The prefix is subdivided into the Directory Indicator [3] separated from the Registrant Code, assigned by the issuing body [24]. Registrant codes are a minimum of three integers, with further optional subdivisions separated by full stops. The suffix is provided by the registrant and can be any form of unicode-encoded text [24].

It was possible to write a 'Regular Expression' pattern (regex) to automatically recognise DOIs within a body of text (Figure 2.3). The flexibility of the registrant code specification means that DOIs cannot always be unambiguously identified in HTML.

[3] The Directory indicator has always been integer 10 for every DOI issued at time of writing.

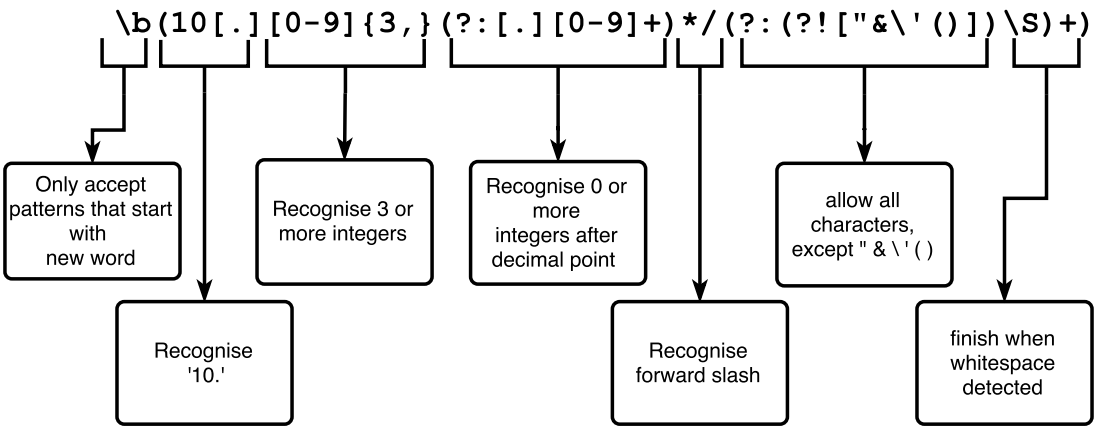


Figure 2.3: Perl Syntax Regex Code that can identify the vast majority of DOIs within free text)

The regex was able to identify 90.4% of the DOIs on `http://www.ch.cam.ac.uk/publications`.

2.3.2 Scraping Program

The regex approach does not require XPaths in order to extract DOIs from a webpage. This facilitates large-scale scraping from many websites. Some meta-data[4] associated with a DOI can be accessed using an online API exposed by Crossref. The remaining meta-data can be accessed by following the `http://dx.doi.org/{DOI}` link to visit publishers' webpages.

With this methodology in place, a scraping program was written to collect DOIs from a list of webpages, collecting meta-data in a two stage process. The Crossref API provides article titles, journals, authors, publisher and publication date meta-data, but not article abstracts. These had to be collected by visiting publisher webpages, and collecting with hand written XPaths[5]. The procedure is summarised in figure 2.4.

[4]See Glossary for project-specific definition of meta-data

[5]Since there are comparatively few publisher websites, only 26 publisher XPaths were required for respectable capture coverage.

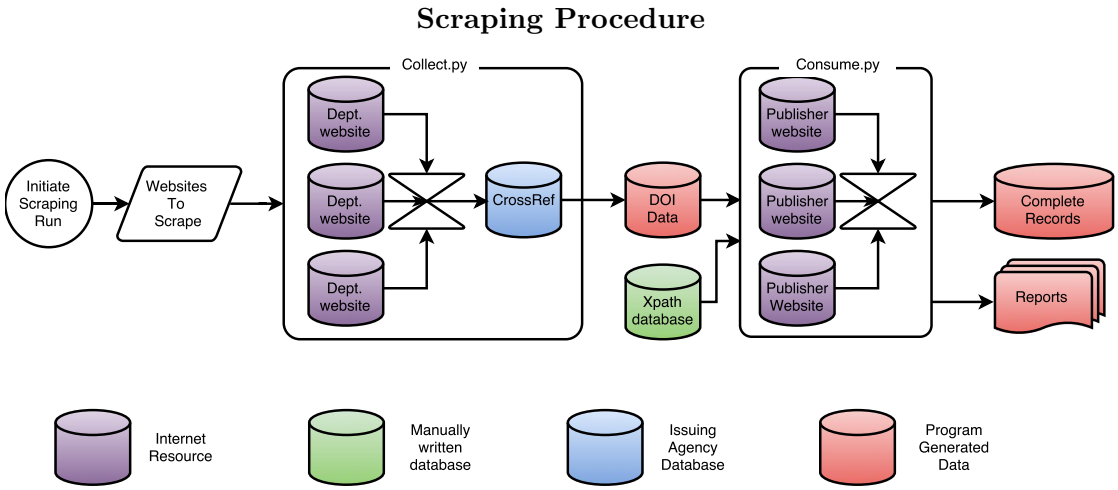


Figure 2.4: The data flow of the scraping program. Websites from an inputted list of websites are visited and DOIs are extracted in the process described in §2.3.1. The Crossref API service is then used to verify the extracted DOIs, and collects available meta-data. The program then accesses publisher webpages and collects abstracts. The program also produces explanation of capture failures and some general statistics

The programmatic steps depicted in figure 2.4 are:

1. `Request the webpage from the inputted list`
2. `Process the html and extract DOIs`
3. `Using the Crossref Online API, verify the extracted DOIs exist`
4. `Crossref yields metadata:`
    - `Title`
    - `Journal`
    - `Publisher`
    - `Authors`
    - `Publication Date`
5. `For each DOI, follow http://dx.doi.org/{DOI} link`
6. `Use XPath to collect article abstracts`

The program exports complete records as .json files, but also feeds to a MongoDB database. Once the program was written, a list of webpages to scrape was required. §2.4.1 and §2.4.2 describe how this was achieved.

2.4 Collection Results

2.4.1 UK University Department scraping

The program was first used to collect the data from the UK. The Goodman group's website hosts a list of UK chemistry departments `http://www-jmg.ch.cam.ac.uk/data/c2k/uk.html`. The list was manually checked and edited to give a list of 68 departments[6]. The program was run using this list, the results of which are detailed in table 2.1. The DOIs collected were stored in database $\Delta 1$ and the complete results were stored in database $\Delta 2$. Conversion losses were due to four components. 45 losses for non-existant

Table 2.1: UK Scraping results

Process	# records	% of maximum yield
Dois collected	22442	N/A%
Dois found with metadata	22397	99.8%
Articles successfully resolved	16363	72.9%
Losses due to failed requests	2753	12.3%
Program errors	133	0.6%
Missing Publication Errors	3148	14.0%

DOIs, 2753 to request errors (404 : not-found errors or permission problems), 133 to the program errors and 3148 to missing publication XPaths. The 26 specified XPaths [7] were sufficient to convert 83.8% of successful requests. This was deemed acceptable, as most major publishers had been covered[8], and the missing publishers each covered a small number of articles[9] The efficiency is depicted in figure 2.5.

[6] Details can be found in the appendix, §8.7

[7] Corresponding to 37 publishers

[8] see appendix for list of covered publishers, §8.7

[9] It would take another 11 XPaths of the missing most popular publishers to increase the conversion rate from 83.8% to 90%.

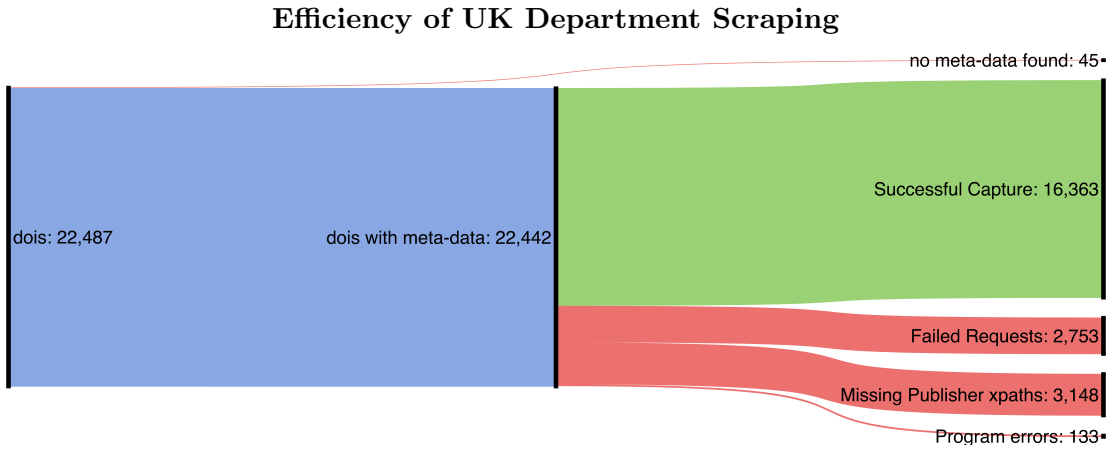


Figure 2.5: The loss processes are coloured red, successfully captured full records in green, and the maximum possible yield in blue.

Interestingly, 9467 out of 16363 successful collections were sourced from `http://www.ch.cam.ac.uk`. This could be because the department at Cambridge has an extensive website and hosts the majority of its information under its own domain name, whereas other departments' data are hosted on central university domains. The program was instructed to only scrape webpages belonging to chemistry department domains, not university websites as a whole. It is noted that the Cambridge chemistry department may be overrepresented in $\Delta 2$.

2.4.2 Global Scale Scraping

Much more data would be required to train a successful machine learning model. One approach would have been to expand to world-wide chemistry departments and other learnèd bodies. However, Crossref also exposes a search service that can be used to query its vast internal database. The program was set up to query the Crossref service for search terms 'Chemistry', 'Chemical', 'Molecule' and 'Molecular' for journal articles and journal titles. This suggested possible yields in the millions of articles.

The program was instructed to scrape the search-result pages of these queries. Because the scraping job was large, it was programmed to 'pause' before publisher abstract collection. The results up to this point were examined before setting off the second stage to collect abstracts.

At the intermediate point, the program had collected 1,267,495 records. This database was labelled $\Delta 3$. Publisher distributions and potential server loads were then carefully considered and capture probabilities were predicted before[10] the second half of the scraping routine was set off to run for three days.

[10]Some of this analysis is presented in §8.7.2.

2.4.3 Problems with ACS and Taylor & Francis

Most publishers track request volumes sent to their servers to discourage automatic downloading. Scraping constitutes fair use and complies to UK copyright law. Despite the university owning a full-access licence to these publishers' publications, the collected material was freely available without licence[25] [26]. During the scraping run, a bug in the randomisation of requests resulted in detection by ACS and Taylor & Francis, which responded by banning the computer's IP address [11]. The department librarians were able to restore access, and it was agreed that no further scraping would be performed.

2.4.4 Analysis of collected data

The yield of the global-scale scraping run was cut significantly by the ACS banning. A summary is tabulated in table 2.2 and shown graphically in figure 2.6. The complete records were stored in database $\Delta 4$.

Table 2.2: Global Scraping Results

Process	# records	% of maximum yield
DOIS collected	1267495	N/A
DOIS collected with meta-data	1267495	100.0%
Predicted maximum capture	1071506	84.5%
Predicted Capture without ACS	581797	45.9%
Articles successfully captured	714370	56.4%
Losses to failed requests (excluding ACS)	53743	4.2%
Losses to ACS banning	303393	23.9%
Missing Publications & Program Errors	195989	15.5%

[11] Explored in §8.7.1

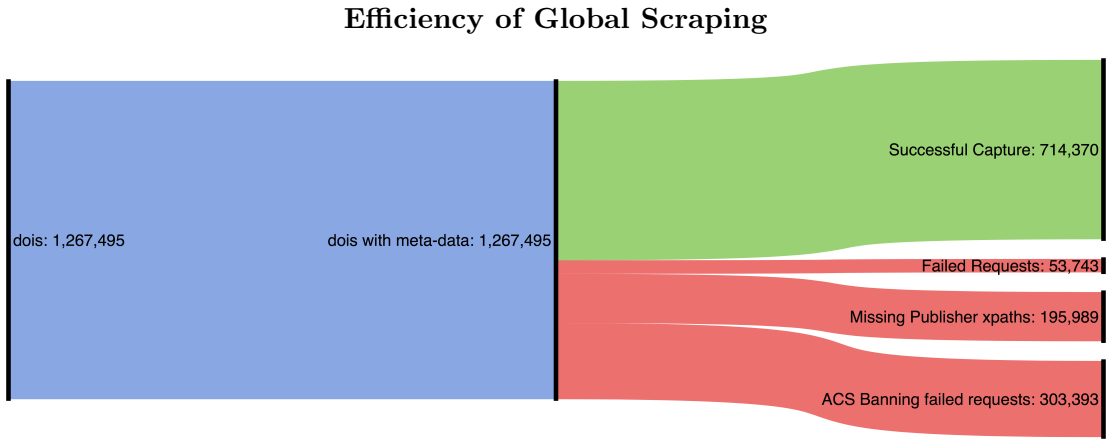


Figure 2.6: The loss processes are coloured red, successfully-captured full records in green, and the maximum possible yield in blue.

The overall efficiency of the process is 56.4%, but excluding lost ACS records, the program's efficiency was 74.0%, similar to the efficiency of the UK scrape (§2.4.1). [12]

The successful 714,370 records were merged with the UK results[13]. Records were rejected with short titles or abstracts, or if the majority of the title and abstract were not written in ascii characters[14]. This was done to provide higher-quality data for algorithm training (see §4). This filtering resulted in a final training database of 464712 articles. This dataset was labelled $\Delta 6$. The database formation process is summarised in figure 2.7 and table 2.3.

[12] Also note that 100% of DOIs were converted to DOIs with meta-data. This was because the format of the webpages scraped was consistent for every DOI collected.

[13] The merged Dataset was labelled $\Delta 5$

[14] likely to be addenda, informal articles, retractions etc, and removing removing majority Japanese and Chinese script

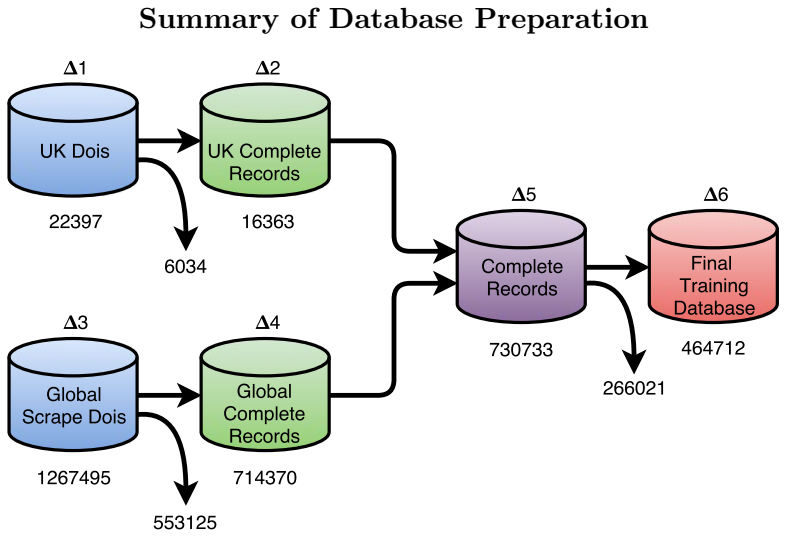


Figure 2.7: Blue databases (Δ1, Δ3) represent data with DOIs and meta-data. Green databases (Δ2, Δ4) represent meta-data, DOIs and abstracts. The purple database (Δ5) is the combined complete records, and the red database is the data deemed suitable for the training algorithm. Database Populations and losses are annotated.

Table 2.3: Databases created in Data Acquisition Process

Database	Contents	# Records
Δ1	Dois found on UK Chemistry Department websites	22,397
Δ2	Complete meta-data obtained from records in Δ1	16,363
Δ3	Dois found in global scraping using crossref API	1,267,495
Δ4	Complete meta-data obtained from records in Δ3	714,370
Δ5	Complete records obtained from combining Δ2 and Δ4	730,733
Δ6	Records appropriate for training taken from Δ5	464,712

It was instructive to examine these databases and derive some simple statistical results, explored in §8.7.2.

3. Techniques for Language Processing

3.1 Background

Natural Language Processing is the application of computer science to study human languages using computers. Machine learning, a class of algorithms for predicting patterns in data, finds many applications in NLP. This section explores approaches to representing journal articles in a quantitative manner using NLP.

3.2 Bag of Words

A simple approach to representing a document is a *bag of words* model. The document is split into component words in an unordered set. The model computes the number of distinct words in a corpus of documents, N. It then assigns each document in the corpus an N dimensional vector $\mathbf{v}$. If document A contains word i 2 times, then $v_{A,i} = 2$. A simple example is given below:

Document A: `A good yield was obtained for a nucleophile`

Document B: `The nucleophile is a good donor`

Document C: `A gaussian basis was used`

Table 3.1: Bag of words

Vocabulary	$\mathbf{v}_A$	$\mathbf{v}_B$	$\mathbf{v}_C$
A	2	1	1
Good	1	1	0
Nucleophile	1	1	0
Yield	1	0	0
For	1	0	0
Is	0	1	0
The	1	0	0
Donor	0	1	0
Was	1	0	1
Gaussian	0	0	1
Basis	0	0	1
Used	0	0	1

Table 3.1 shows vector representations for Documents A, B and C. The higher the scalar product of normalised $\mathbf{v}_A \cdot \mathbf{v}_A$, the more similar the documents are predicted to be:

$$\mathbf{v}_A \cdot \mathbf{v}_A = 0.567 \quad \mathbf{v}_A \cdot \mathbf{v}_C = 0.424 \quad \mathbf{v}_B \cdot \mathbf{v}_C = 0.200$$

and so documents A and B are the more similar. The related *bag of citations* model sets vector components according to the presence of citations. Both models are used by the scientific publishing industry[1] [2].

3.3 Word2Vec

The *Bag of Words* model treats words as atomic units, beneficial for robust computation. However, words have degrees of similarity to each other, which are not captured by bag of words models [27]. Distributed representations have been used to address this for some time [28].

A recent successful approach has been the Word2Vec algorithm [18] [19]. Word2Vec uses a neural net to represent words as vectors in a continuous space. Vectors for words with similar meanings will point in similar directions in this 'semantic space'. Word2Vec is fed a corpus sentence by sentence. The words within the sentences are semantically related, which the algorithm uses to infer word meanings.

This is achieved with two architectures, Continuous Bag of Words (CBOW) and skip-gram. The CBOW architecture uses a shallow neural net to predict a word's vector by

[1] Dr. Colin Batchelor (Senior Data Scientist, Royal Society of Chemistry), personal communication, February 2016

[2] P.E. Peter Murray-Rust, personal communication, February 2016

summing or averaging the vectors of surrounding words in a training sentence. The skip-gram architecture predicts the vectors of words surrounding the current training word. By training with many input sentences, prediction vectors are gradually improved.

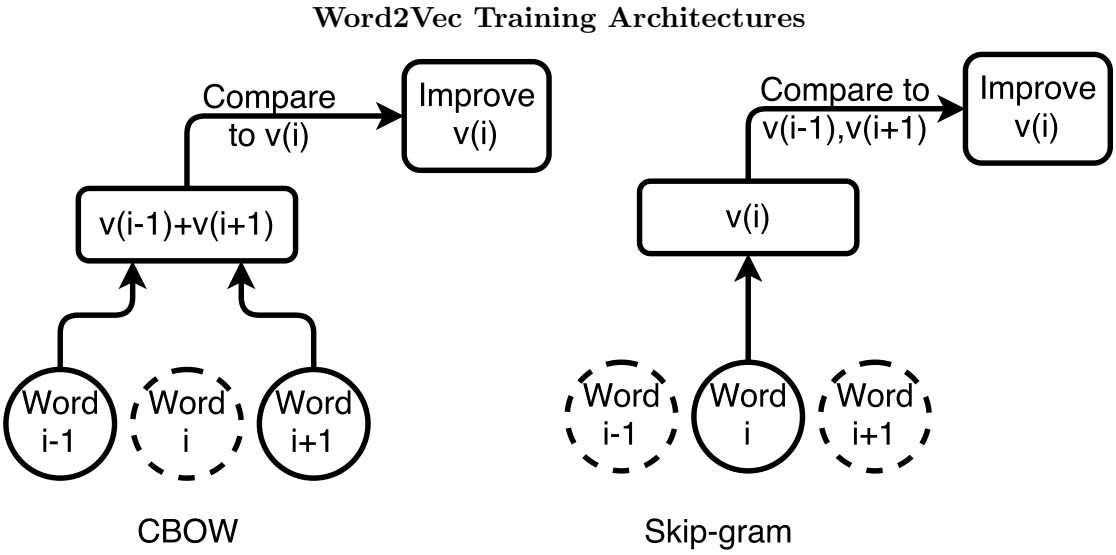


Figure 3.1: The training architectures of the Word2Vec training algorithm. Word vectors are denoted $v(i)$ for word i. In CBOW word i is predicted by the vector found by summing vectors surrounding i, and $v(i)$ is adjusted to be closer to this prediction. In skipgram, word i's vector is pairwise compared to its context words, here i-1 and i+1 as a basis to improve $v(i)$.). CBOW attempts to make words similar the sum of the surrounding words, skipgram attempts to minimise distance to each surrounding word.

The training process is shown in figure 3.1. CBOW uses a fixed window of surrounding words. The order of words within the window does not matter, but because the window 'slides' along as the algorithm considers words i+1, i+2... word ordering is represented in the model . In skipgram, a random number of surrounding words are used for the prediction vectors for word i.

The model has added sophistication to reduce the importance of commonly occurring words, and to identify phrases. The word vectors that are produced encapsulate both semantic and syntactic meanings, and can be manipulated to represent concepts and relationships.

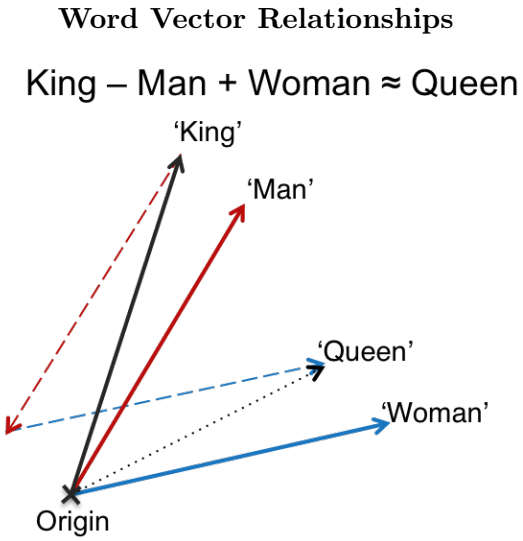


Figure 3.2: Schematic Representation of how concepts can be represented in word vector space. Word2Vec is able to replicate this behaviour. The vector found by vec(King)- vec(Man)+vec(Woman) is approximately equal to vec(Queen). The model has been tested on thousands of similar examples[19][27].

Word2Vec models may represent concepts by vector algebraic operations on their word representations. Figure 3.2 shows one famous example a Word2Vec model trained on the 'Google News' text corpus was able to identify. [3]

3.4 Doc2Vec

The Doc2Vec algorithm[29] (an implementation of Paragraph Vectors [30]) allows the Word2vec process to directly learn vectors representing documents. The CBOW architecture is adapted so that, in addition to word vectors, each document is associated with its own vector that contributes to predictions in training. The result is that documents can be represented by vectors in a document semantic space.

The nature of the collected meta-data detailed in §2 (A large store of natural language) lends itself to the Word2Vec and Doc2Vec algorithms. The focus of the machine learning analysis phase of the project was directed at applying Word2Vec and Doc2Vec to $\Delta 6$ to automatically learn and classify chemical semantic concepts.

[3] It is interesting to consider if chemical examples could be developed. This is considered further in §8.1

4. Algorithm Development - Experimental

4.1 Premise

The aim of the machine learning phase was to apply the Word2Vec and Doc2Vec algorithms to dataset $\Delta 6$ described in §2. An article was considered to be represented by a document consisting of its title and abstract. The aim was to represent these documents as vectors in semantic space, so that advanced computational analyses and statistical methods could be performed. This section constitutes an experimental section.

4.2 Data Sanitisation

The documents (titles and abstracts) in $\Delta 6$ required preprocessing before they could be effectively used in training. The training process requires inputs to be as clean as possible in order to get good results (encapsulated by the well-known computer science idiom 'Garbage in, Garbage out').

The first step was to cast all words to lower case, so that the algorithm did not produce different vectors for e.g. 'Molecule' and 'molecule'.

The raw documents also frequently contained artefacts from the source webpages (unwanted whitspace, vestigial HTML tags, 'newline' characters and carriage returns). The algorithm training word vectors for these symbols is clearly undesired behaviour, so these were removed and whitespace normalised.

It was also observed that, as unicode text scraped from a wide variety of sources, there was varied and redundant punctuation. Punctuation would be treated as separate words by the algorithm, so had to be carefully removed. Unicode has very wide variety of different punctuations. For example, unicode encodes 24 different types of hyphen. Table 4.1 shows the punctuation that was filtered out of the documents. Large sections of unicode script (sections of non-western languages) was also removed as the algorithm works best on a smaller vocabulary.

Filtered Punctuation

"	+	?	–	—	−
#	,	@	–	′	≡
$	.	[	-	‘	⟶
%	/	]	✯	’	➞
&	-	^	▪	•	⇻
\	:	_	”	†	⇒
'	;	`	≈	°	
(	<	{	⇅	“	
)	=	|	~	⊕	
*	>	}	━	−	

Figure 4.1: All the punctuation removed in sanitation. Only these were found in appreciable quantities in the $\Delta 6$.

Removing hyphens and primes also meant chemical names were fragmented. This was considered acceptable as the fragment words had greater freedom than specific (possibly singleton) fully-formed names, e.g. `5-methyl-1-heptanol` is split to `5 methyl 1 heptanol`, this allows the `heptanol` fragment to be associated with other mentions of heptanol in the training set, rather than only associate with mentions of the much less frequent 5-methyl-1-heptanol.

Next, English stopwords were removed [1] (stopwords were taken from the Porter stopwords corpus[31] [12]). From inspection of the zipfian frequency table, (§8.7.2), it was apparent that chemistry literature also generates stopwords. Table 4.1 details ‘Chemistry’ stopwords that were identified and removed[2].

Finally, the processed words were sent through a ‘stemming algorithm’[3]. Several stemming algorithms were assessed (Porter [12], Snowball [13][31], Lancaster [9] and the Wordnet Lemmatizer [20][21][22]). The Snowball [4] stemmer was found to strike a good balance between making an appreciable number of contractions (superior to Wordnet) whilst minimising conflations and over-contraction (superior to Lancaster and Porter). See Table 4.2

[1] Stopwords are commonly occurring words in a corpus that hold little information, e.g. (‘the’, ‘a’, ‘and’...)

[2] The stopwords were chosen from high on the rank frequency table (they appeared extremely commonly) and because they were considered to encode little information; for instance, the digits appeared so frequently and in such an wide set of contexts, no meaningful vector would be trained. The word structure appeared so frequently, so as to encode very little actual information

[3] A stemming algorithm seeks to map derived words onto the same root, such as polymer and polymers, but some also attempt more complex cases such as morphologic and morphology

[4] Also known as Porter2

Table 4.1: Chemistry stopwords

chemistry	containing	7	six	water
structure	novel	8	seven	also
structural	study	9	eight	method
study	studies	0	nine	molecular
new	1	zero	ten	studied
using	2	one	phase	
based	3	two	based	
reaction	4	three	compounds	
reactions	5	four	high	
chemical	6	five	results	

Table 4.2: Stemming Comparisons

Word	Porter Stemmed	Snowball Stemmed	Comment
phyllenthus	phyllenthu	phyllenthus	Overagressive stemming by Porter
angularly	angularli	angular	Adverbs map to root better
infinitly	infinitli	infinit	Snowball maps these to correct root
infinite	infinit	infinit	

Word	Lancaster Stemmed	Snowball Stemmed	Comment
pigment	pig	pigment	Lancaster collapses too far
conductive	conduc	conduct	Lancaster conflates these different words
conducive	conduc	conduct	
scripting	scripting	script	Lancaster doesn't consider present participles
aroma	arom	arom	Preferable not to map aroma and aromatic to same root
aromatic	arom	aromat	

The document preprocessing pipeline is shown diagrammatically in figure 4.2:

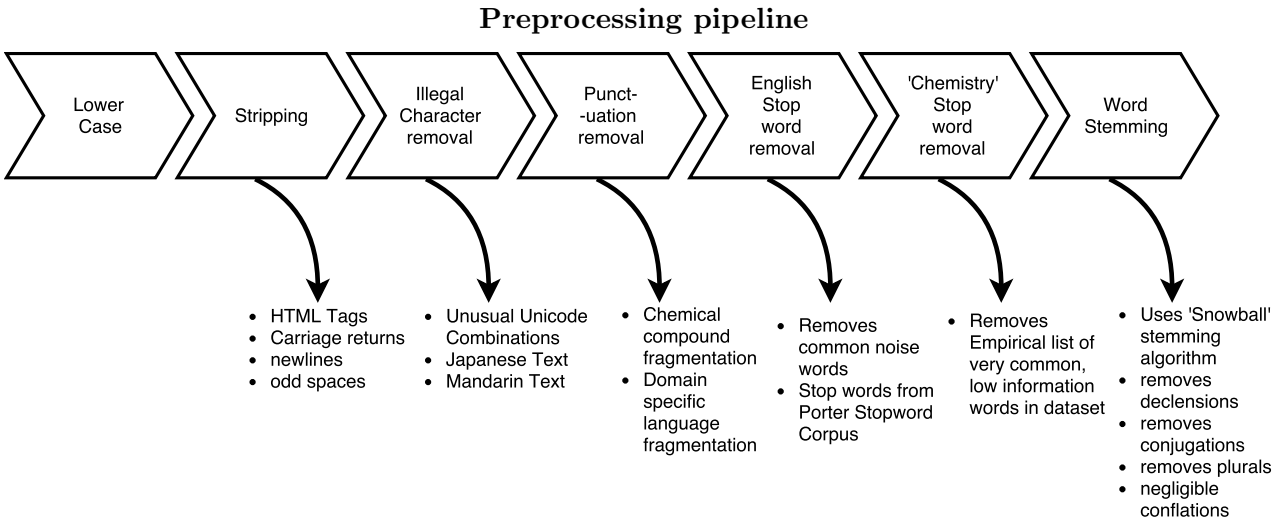


Figure 4.2: All documents in $\Delta 6$ were preprocessed with this pipeline schema before being used in training models

The process is best illustrated by real example from $\Delta 6$:

< p > `n A 9-silafluorene-containing biphenolic monomer, 9,9-bis(4-hydroxyphenyl)-9-silafluorene, was prepared from 9,9-dichloro-9-silafluorene and employed for the synthesis of polyesters using a fluorene-based homoditopic acid chloride.` < \ p >. [32] processed into:

`silafluoren biphenol monom bis hydroxyphenyl silafluoren prepar dichloro silafluoren employ synthesi polyest fluoren homoditop acid chlorid`

Whilst challenging to read, word order is preserved and low information words (or words with complex, diverse meanings such as numbers) have been removed to give good-quality input data. Note how chemical names have been fragmented so that multiple chemical vectors can be learned, rather than the fewer complex vectors (`9,9-dichloro-9-silafluorene` vs `dichloro` and `silafluoren`).

4.3 Word2Vec Models

The processed data was used to train two Word2Vec models (one CBOW, one skipgram) using the gensim implementation [29].The hyperparameters used for training were consistent for the two models. Training was carried out on all documents in $\Delta 6$. The model was trained with sentences formed by simple splitting of documents using full stops[5].

[5]Whilst not perfect, this method was a fair compromise for partitioning on actual sentences and false partitioning.

After examination of different hyperparameters, the models were run using hyperparameters representing good balance of specificity, speed and generality. The hyperparameters used are detailed in table 4.3.

Table 4.3: Word2vec Parameters

Model Parameter	CBOW and skipgram
Vector Dimensionality	100
Minimum word frequency	1 (all words)
Initial learning rate α	0.025
Minimum learning rate α_{min}	0.0001
Epochs of training	24
Sliding word window size	5
Negative sampling	Yes
Downsampling parameter	0.001
Hierarchical Softmax	No
CBOW Mean	Yes (Not applicable for Skipgram)

In order to represent documents as vectors using these models, the component word vectors had to be aggregated into a single vector. There were several possible aggregation techniques, described below.

4.3.1 TF-IDF

TF-IDF [6] is an empirical metric for weighting the importance of words in a sentence. If averaging word vectors, it is intuitive that equal weighting should not be given to information heavy and trivial words. The TF-IDF weight, defined as term frequency: $TF(w,d) = f_{w \in d}$ where $f(w)$ is the raw frequency of a term w in a document d, multiplied by inverse document frequency $IDF(w) = log_2\left(\frac{|D|}{\sum_d df_{w \in d)}}\right)$ where $|D|$ is the number of documents in the corpus, df is 1 if word w is in document d, otherwise otherwise 0 [29]. TF-IDF assigns small weights to words that are common across the corpus. It assigns high weights to words that appear often in a document but rarely in the corpus.

4.3.2 Aggregations

Document vectors could be created by averaging word vectors into sentence vectors, followed by averaging sentence vectors into document vectors, or by simpy averaging word vectors directly into documents. 8 models for document vectors composed of Word2Vec models were constructed, detailed in table 4.4

[6] Term - frequency Inverse - Document - Frequency

Table 4.4: Word2vec Document Vector Models

Model Name	Description	Model Name	Description
CBOW-W	Simple average of CBOW word vectors	SG-W	Simple average of skip-gram word vectors
CBOW-S	CBOW word vectors averaged to sentence vectors, then sentence vectors averaged	SG-S	SG word vectors averaged to sentence vectors, then sentence vectors averaged
CBOW-TFIDF-W	CBOW-W with TFIDF weighting on word vectors	SG-TFIDF-W	SG-W with TF-IDF weighting on word vectors.
CBOW-TFIDF-S	CBOW-S with TFIDF weighting on word vectors	SG-TFIDF-S	SG-S with TF-IDF weighting on word vectors

4.4 Doc2Vec Models

A Doc2Vec model was trained with a *distributed memory* architecture[7] using the gensim framework in python [29] using $\Delta 6$ with the same sanitiation pipeline as for the Word2Vec models. The training sentences were labelled with the document (journal article DOI) they came from. 100 dimensional vectors were chosen as a compromise of training speed and specificity[8], and also so that dimensions were consistent across all models. The Doc2Vec model was trained for 24 epochs, with hyperparameters detailed in table 4.5

Table 4.5: Word2vec Parameters

Model Parameter	value
Vector Dimensionality	100
Minimum word frequency	1 (all words)
Initial learning rate α	0.025
Minimum learning rate α_{min}	0.0001
Epochs of training	24
Sliding word window size	8
Negative sampling	No
Hierarchical Softmax	Yes
CBOW Mean	Yes (Not applicable for Skipgram)

The model took considerably longer to train than Word2Vec, as there were appreciably

[7]Distributed memory is the Paragraph Vector algorithm equivalent of CBOW, the performance of this architecture is optimal[30]

[8]as well as for computational considerations of analytical techniques, see §5,§6

more work required per document. Negative sampling was disabled as per recommendations in the literature. [29] [30]. The Doc2Vec and Word2Vec models are assessed in §5.

5. Model Examination

The models created in §4 were then examined and assessed. As an unsupervised learning algorithm, it is challenging to assess model quality, due to a lack of concrete metrics for comparisons[1]. The Word2Vec development team tested against approximately 10,000 semantic and syntactic relationships (See Figure 3.2)[18] [19] [27]. The scope of this project does not extend to such elaborate tests. In the section, some examples of model strengths are given and techniques for using word vectors and visualisation are presented.

5.1 Word Similarities

Word similarities can be be obtained by direct comparison of their word vectors. A possible metric is to compute euclidean distance. For words α and β, with vectors ν^α and ν^β,

$$S_{euclid} = \sqrt{\sum_{i=1}^{D}(\nu_i^\alpha - \nu_i^\beta)^2}$$

where D is the dimensionality (D=100). S_{euclid} simply describes the distance between the end points of ν^α and ν^β. A larger S_{euclid} indicates weaker similarity. A second similarity metric is *cosine similarity*, a measure of the directionality. A value close to 1 corresponds to high similarity of α and β. Cosine similarity is computed as:

$$S_{cos} = \frac{\nu^\alpha \cdot \nu^\beta}{|\nu^\alpha||\nu^\beta|} = \frac{\sum_{i=1}^{D}\nu_i^{(\alpha)}\nu_i^{(\beta)}}{\left(\sum_{i=1}^{D}(\nu_i^{(\alpha)})^2\right)^{1/2}\left(\sum_{i=1}^{D}(\nu_i^{(\beta)})^2\right)^{1/2}}$$

[1]This is to say, there is no objective 'similarity' relationship value between two words to compare models to.

The CBOW and skipgram models were examined using these metrics. For a given word, they were requested to return the three words in the corpus with highest similarity. Some examples are given in tables 5.1 and 5.2.

Table 5.1: Closest words to test words using cosine similarity

Model	Test Word	Most Similar	2nd	3rd
CBOW	Iron	Chromium	Manganes	Nickel
Skip-gram		Chromium	Manganes	Nickel
CBOW	Colloid	Nanoparticl	Nanos	Monodispers
Skip-gram		Particl	Spheric	Suspens
CBOW	Statistical	Varianc	Bayesian	Multivari
Skip-gram		Nonparametr	Varianc	Bootstrap
CBOW	Plastic	Thermoplast	Elastomer	Nonwoven
Skip-gram		Nonwoven	Thermoplast	Textolit
CBOW	Catalyst	Cocatalyst	Nanocatalyst	Precatalyst
Skip-gram		Catalyt	Nanocatalyst	Polystyrylbipyridin

Table 5.2: Closest words to test words using Euclidean similarity

Model	Test Word	Most Similar	2nd	3rd
CBOW	Iron	Chromium	Managanes	Nickel
Skip-gram		Chromium	Manganes	Nickel
CBOW	Colloid	Nanos	Ultrafin	Agglomer
Skip-gram		Particl	Suspens	Spheric
CBOW	Statistical	Varianc	Phenomenolog	Bayesian
Skip-gram		Nonparametr	Bivari	Multigrid
CBOW	Plastic	Thermoplast	Elastomer	Mold
Skip-gram		NRL	Prepreg	Sealant
CBOW	Catalyst	Nanocatalyst	Cocatalyst	Precatalyst
Skip-gram		Catalyt	Molybdena	Nimo

As shown above, the models perform well, returning intuitively similar words to the test word.[2]. In most cases, chemical inference is represented in some way [3].

It was observed that the skipgram model gave misleading positives more frequently. [4]. CBOW was considered to be superior for word-word comparisons. It was also noted that

[2]Note that returned words are 'stemmed' from use of stemming algorithm before training. It is not difficult to interpret derived words from their stems

[3]e.g. the models understood that catalysts and nanocatalysts are closely-connected concepts

[4]in the catalyst case above, skipsgram associated a stemming false negative and gave a specific catalysed compound to the 'catalyst' test word than more obvious, fundamental associations

CBOW had closer agreement between S_{euclid} and S_{cosine}, however, euclidean similarity gave poorer general performance.[5]. It was noted that S_{Cosine} is the accepted similarity metric in the literature [18] [19] [30].

5.2 Document Similarities

The models detailed in §4 were then tested for document vector similarity. A document was chosen from the corpus, the three most similar articles were computed for each model and results assessed. One test document was `DOI: 10.1134/s0036024412120266` [33]:

`'Photochemical transformations of anthraquinone in polymeric alcohols'`.

The TF-IDF models (CBOW-TFIDF-S, CBOW-TFIDF-W, SG-TFIDF-S, SG-TFIDF-W) suffered from mathematical conditioning problems, giving poor predictions. The remaining models' most similar documents [6] for this test document are shown in table 5.3:

[5] 'NRL' in the 'Plastic' category of skip-gram is a typical example of poorer euclidean performance. NRL appears to be a reference to the Navy Reseach Laboratory.

[6] Cosine similarity was used, as it performed better than euclidean similarity for document vectors.

Table 5.3: Document Vector Similarities to [33]

Model	DOI	Title	DOI	Title
CBOW-W	10.1080/ 00222338 208074396	Oxidation of Poly(dimethylbutadiene) Popcorn Polymer	10.1246/ cl.1974.133	photochemical reaction of 2-cyanoquinoline 1-oxides in an acidic alcohol. synthesis of 6-alkoxy-2-cyanoquinolines
CBOW-S	10.1002/ pol.1985 .170230401	Polymerization of glycidol and its derivatives: A new rearrangement polymerization	10.1080/ 002223381 08074381	Cyclic Acetal-Photosensitized Polymerization. 9. Photopolymerization of Triallylidene Sorbitol
SG-S	10.1002/ pola.1991. 080290207	Photochemical synthesis of nitroxyl free radicals in the presence of vinyl monomers	10.1002/ pola.10311	Benzyl alcohols as accelerators in the photoinitiated cationic polymerization of epoxide monomers
SG-W	10.1080/ 00222338 208074396	Oxidation of Poly(dimethylbutadiene) Popcorn Polymer	10.1080/ 00222338 408077237	Photopolymerization of Acrylonitrile: Benzophenone-Isopropanol System as Initiator
doc2vec	10.1002/ pola.10059	Aqueous photopolymerization with visible-light photoinitiators: Acrylamide polymerization photoinitiated with a phenoxazine dye/amine system	10.1080/ 10587259 408029732	An Investigation into the Solid-State Behaviour of Anthraquinone and Its Derivatives

The document vectors generated by the Doc2Vec model had considerably better general performance, and were selected as the model of choice for further analysis.

5.3 Visualisation Techniques

5.3.1 Network Visualisation

High Dimensional systems are difficult to visualise but there are several methods available to visualise high-dimensional data. PCA[7] [11] and TSNE[8][14][34] techniques allow 100-dimensional document vectors to be collapsed to points on an arbitrary 2D plane, to give a visual 'snapshot' of the semantic space. Figures 5.1 and 5.2 show PCA and TSNE reductions[9] on 10,000 document vectors randomly selected from the Doc2Vec model representation of $\Delta 6$. [35].

[7] See Glossary
[8] See Glossary
[9] performed using scikitlearn and python

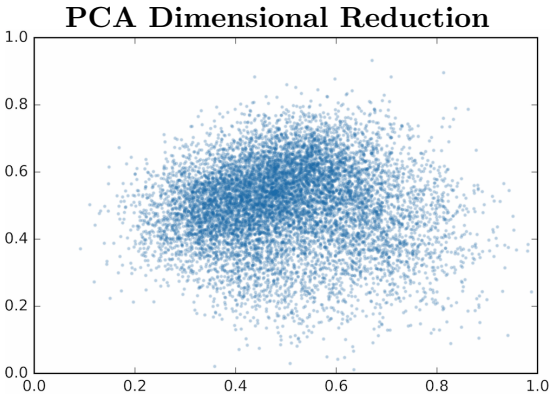


Figure 5.1: PCA map of 10,000 documents in the corpus. PCA has not resolved any particular structure. The dimensional reduction task is probably too challenging for PCA.

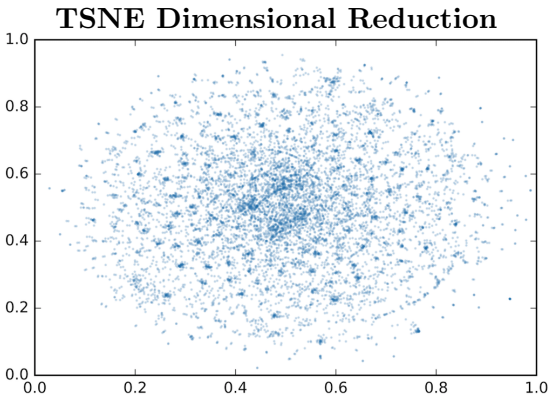


Figure 5.2: TSNE map of the same 10,000 documents. TSNE is a much more sophisticated, state of the art technique, that seeks to preserve high dimensional clusters. Document vectors have gathered into noticeable clusters, with non negligible outlier documents between clusters. Performed using Barnes-Hut TSNE implementation as sample size is large.

The PCA reduction shows a dark central area, suggesting most vectors are 'smeared' about a common direction. The map is not symmetric which is what would be expected for random vectors. It was expected that document vectors would be distributed in clusters representing particular research fields within the literature. This is indeed seen in the TSNE reduction, which resolved many clusters. There are document vectors scattered between dark cluster spots, which may be could interpreted as 'interdisciplinary'[10]. TSNE is based upon euclidean distance, which is noted not to be the best similarity measure. Whilst qualitatively useful, TSNE maps were interpreted cautiously.

5.3.2 Networks and Network Visualisation

A S_{cosine} matrix $\mathbf{C}$ was defined between sets of documents. For a set of documents A (with a documents) and B (with b documents) document matrices of document vectors were defined, $\mathbf{A}$ and $\mathbf{B}$, such that

$$\mathbf{A} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \mathbf{w}_1 & \mathbf{w}_1 & \cdots & \mathbf{w}_a \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_1 & \cdots & \mathbf{v}_b \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

where w and v are document vectors. $\mathbf{C}$ was then defined as $\mathbf{C}_{i,j} = cosine\,(\theta_{i,j})$ where

[10] More in-depth analysis of TSNE maps forms part of the recommended work section, §8.1

element i j contains the cosine between ith vector in A and jth vector in B:

$$\mathbf{C} = \mathbf{A}^T\mathbf{B} \oslash \left(diag(\mathbf{A}^T\mathbf{A})^T diag(\mathbf{B}^T\mathbf{B})\right)^{\circ\frac{1}{2}}$$

where $\oslash$ and $^{\circ\frac{1}{2}}$ indicate Hadamard division and Hadamard square root, $diag(Q)$ the $1\times n$ matrix formed from the diagonal of Q. $\mathbf{C}$ represents a network where each document in A is a node with an edge to every document in B with weights equal to the cosine. If A=B, then the matrix is a fully connected network[11]. This network can be visualised using specialist software[12] [36]. Figure 5.3 visualises the same 10,000 document sample as a network graph.

[11]That is to say, if A and B are the same set of documents, every document node has an edge to every other document in the set

[12]Gephi, a powerful, open source network visualisation and analysis suite, was used for this purpose.

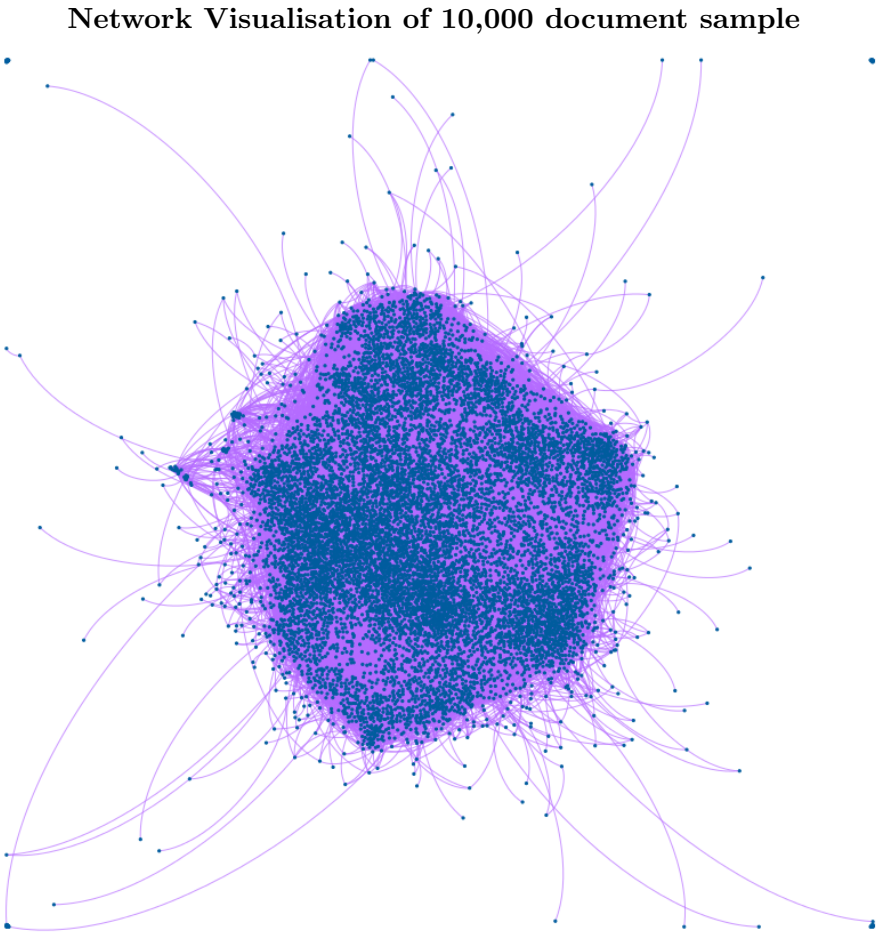


Figure 5.3: A Network visualisation of the 10,000 document sample. Nodes (blue) are spatially distributed by modelling the edges (purple) as springs connecting nodes with spring constants equal to cosine similarity, then allowing the system to approach equilibrium. Edges were only placed between nodes with cosine similarity greater than 0.35 for computational tractability. The edges have been curved to aid visualisation.

Concentrations of documents also form in the network visualisation. There are noticeable outlier documents far from the central clusters [13]. Also note that the network visualisation technique is dependent only on cosine similarity, so was considered a more reliable analytical tool than TSNE. Treating the system as a network graph also enables powerful network analysis algorithms to be applied.

[13]These articles are predicted to be short, or qualitatively different from, the majority, e.g. addenda or retraction notices, rather than proper articles. See §8.6 in appendix for examination of some of these singletons

6. Analysis with Sample Dataset

Having developed a framework to examine the models, attention was turned to some analyses that could be carried out within the time frame and scope of the project[1]. With this in mind, it was decided to focus analysis on a smaller subset of $\Delta 6$, documents from the University of Cambridge Chemistry Department. This dataset was labelled $\Delta 7$.

6.1 Cambridge Chemistry research clusters

$\Delta 7$ contained 9467 documents. The cosine matrix was calculated and a network was constructed from the matrix. *Communities* within the network (clusters of strongly-connected nodes) were identified by applying a modularity algorithm[4][5]. The result is shown in figure 6.1.

[1] Please refer to §8.1 for recommendations for further work

Figure 6.1: A Network visualisation of $\Delta 7$*. Edges were placed between nodes with weights corresponding to cosine similarity if* S_{cosine}*. Nodes are coloured by their detected communities, and node size is proportional to the number of connections a node has. Nodes are arranged by modelling edges as springs.*

It was apparent that $\Delta 7$ contained clear communities. This corresponds to different fields of research within the department. Some communities were small, but some most large (green, orange, etc...). The algorithm was then re-applied only to the 'green' community, which revealed subcommunities. A program was then written to recursively find subcommunities in $\Delta 7$. This resulted in $\Delta 7$ being divided into 300 communities of comparable size. The smallest communities were singleton documents, the largest was

434 documents, and the mean population was 34.5. The community-finding subdivision process is shown in figure 6.2

Recursion Tree for Community Generation Process

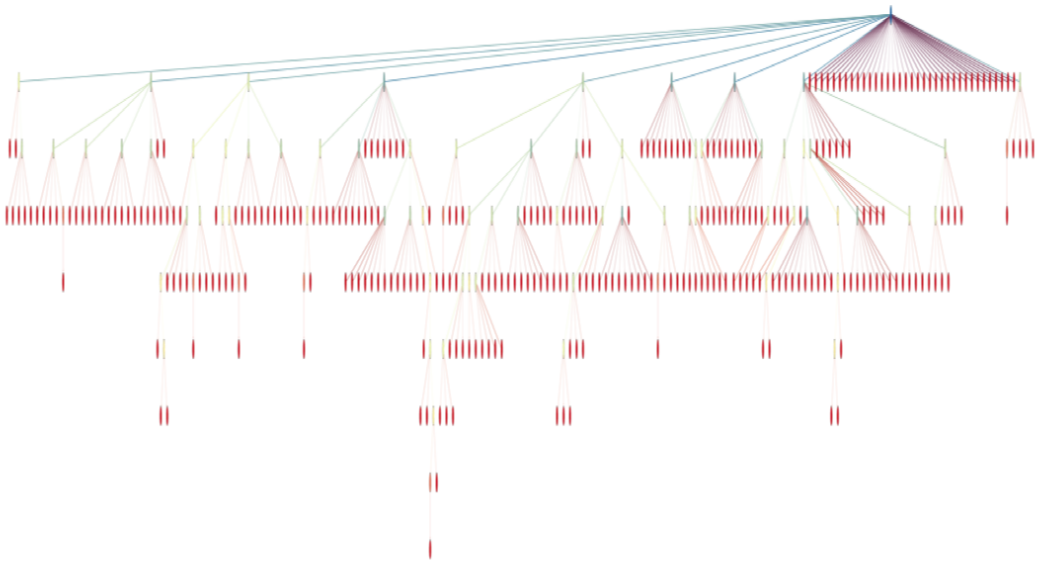


Figure 6.2: Recursion Tree for how communities were derived. The dataset was partitioned using the modularity algorithm. Sets with more than 100 documents were then repartitioned recursively. Sets of less than 100 documents were considered to be communities (red nodes in the diagram). If the algorithm could not partition a set any further, the recursion was stopped and the set was considered a community, even if it was larger than 100 documents. The figure shows the maximum depth of partitioning required was eight, and most communities were found after three partitions.

Figure 6.2 can be interpreted as showing *relationships* between different fields of research within the department. The tree is shallow with highly branched nodes, suggesting wide research fields, and much qualitative overlap between fields. The process constitutes an unsupervised categorisation algorithm[2]. It was instructive to examine what the algorithm defined as communities. Communities were examined and community clustering made intuitive sense in the majority of cases. Community 275 is typical:

[2]The entire process, from model training to finding communities has been performed without human labelling or intuition

Table 6.1: Community 275

Community Size	15
Depth down Recursion Tree	2
Contents	Bees, Neonicotinoids, toxicology, pollen.
Article closest to Mean Vector	10.1021/es2035152: Assessment of the Environmental Exposure of Honeybees to Particulate Matter Containing Neonicotinoid Insecticides Coming from Corn Coated Seeds
Community members	**(Some omitted for brevity)**
10.1007/s00216-012-6338-3	UHPLC-DAD method for the determination of neonicotinoid insecticides in single bees and its relevance in honeybee colony loss investigations
10.1021/es2035152	Assessment of the Environmental Exposure of Honeybees to Particulate Matter Containing Neonicotinoid Insecticides Coming from Corn Coated Seeds
10.1007/s11356-014-3470-y	Systemic insecticides (neonicotinoids and fipronil): trends uses mode of action and metabolites
10.1111/j.1439-0418.2012.01718.x	Aerial powdering of bees inside mobile cages and the extent of neonicotinoid cloud surrounding corn drillers
10.1098/rsif.2013.0394	Analysing photonic structures in plants
10.1007/s00114-013-1020-y	The influence of pigmentation patterning on bumblebee foraging from flowers of Antirrhinum majus
10.1111/ics.12035	Keratins and lipids in ethnic hair
10.1021/ja047905n	Photoluminescent Layered Lanthanide Silicates

Table 6.1 shows that this particular research community refers mainly to toxicology studies of neonicotinoids, bees and flowers[3]. The connections mostly make sense. Note the surprising inclusion of the cosmetics and silicate studies. Upon investigation, both studies used very similar analytical techniques used elsewhere in the community, and both examined intercalation[4]. [5]

Note also that the mean vector for the community was closest to a paper in $\Delta 6$ that summarised the community extremely well[6]. This paper could be considered as a *Summary paper.* The uses of this kind of analysis include:

- Analysis of literature field - trees such as figure 6.2 can give an understanding of how facets of a field link up.
- Research tool: If researching a paper, identifying its community immediately pro-

[3]Note only some members of the community are shown above. Care was taken to give a representative sample of all 15 articles. The rest refer to Neonicotinoid insecticide studies with honey bees, and honey bee affinity to corn and pollen

[4]Some further discussion of community 275 can be found in §8.4

[5]Both used made use of powder X-ray diffraction, and the silicates paper used thermogravimetry, the cosmetics study uses FID and several types of liquid chromatography, all methods used in the bee/neonicotinoid studies.

[6]The paper happened to be in the community itself, but was not restricted to be so

vides the researcher with related papers. This is done *without following citations*, so that interesting, perhaps overlooked, links can be found.

- Summarising: If a researcher is required to read many papers from a field, they could find the communities involved and begin by reading 'summary' papers.

6.2 Cambridge Staff Member Similarities

It is not only articles themselves that can be grouped and analysed. Articles can be aggregated together to represent higher concepts, such as staff members[7]. To investigate this further, `http://www.ch.cam.ac.uk/publications/authors` was scraped in order to associate the documents in $\Delta 7$ with particular staff members.

A cosine matrix was created for each pair of authors A and B, authoring α and β documents respectively, $\mathbf{C}^{(\mathbf{A}),(\mathbf{B})}$ (see §5.3.2). The similarity between the author pair was defined as

$$S_{A,B} = \sum_{i}^{\alpha} \sum_{j}^{\beta} C_{i,j}^{(A),(B)}$$

An *author similarity matrix* can then be built up, $\mathbf{M}^{Auth.Sim.}$, with elements $\mathbf{M}_{A,B}^{Auth.Sim.} = S_{A,B}$. A similar technique to that described in §6.1 could have been used to create clusters of authors. Since the sample size was now much smaller (47 authors compared to 9467 papers) a more appropriate technique, Dedicated Hierarchical Clustering, specifically UPGMA was applied [15] [8]. This method clusters the authors pairwise in a hierarchical fashion. An effective visualisation of the similarities between staff was to plot a *clustermap* [2] [3].

[7] or research groups, or potentially even departments.

[8] See glossary

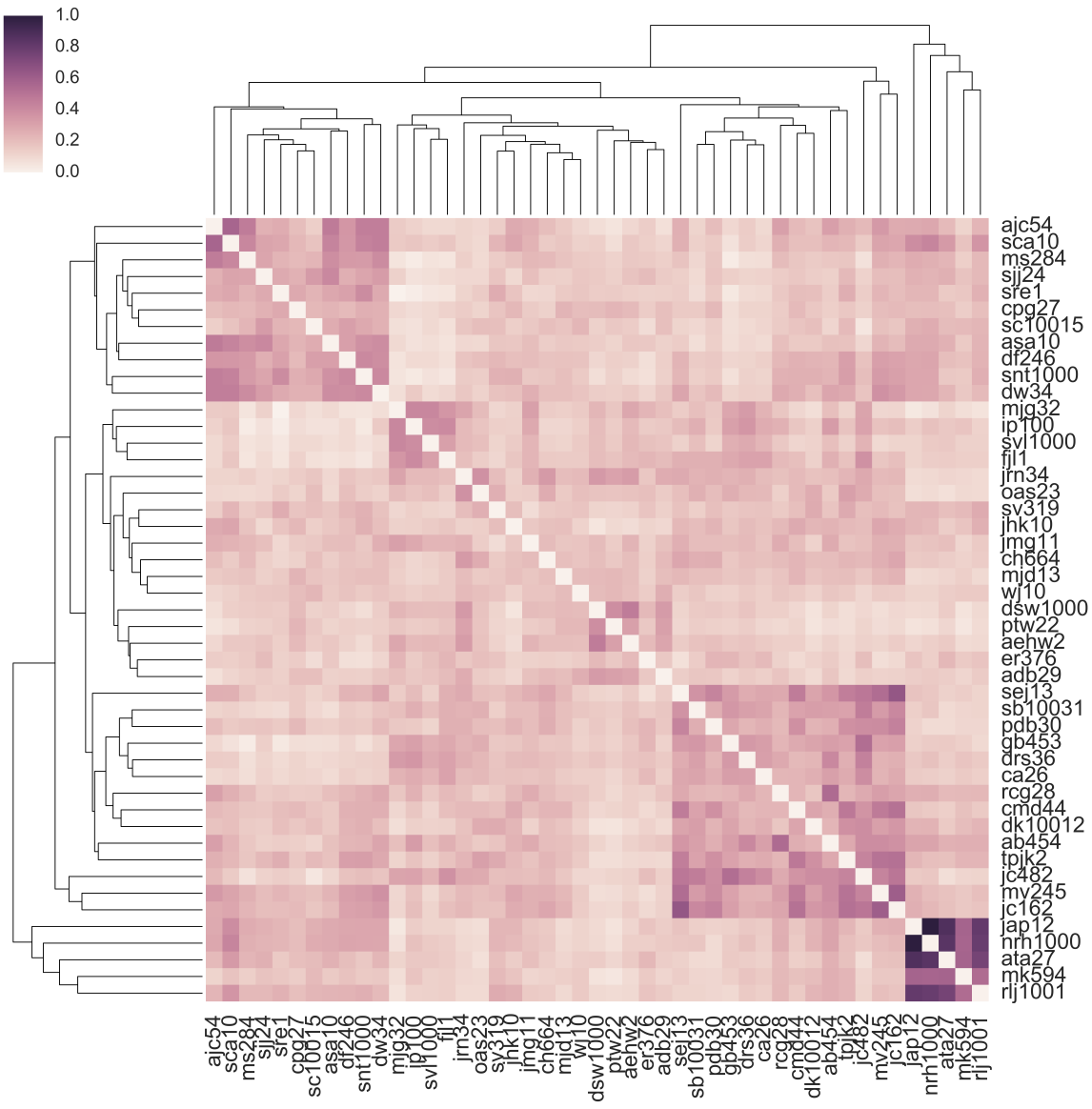


Figure 6.3: This figure shows a heatmap of author similarity. Dark pixels correspond to the author in the pixel's row having similar research interests to the author in the pixel's column. The authors are labelled by crsid. The matrix has been scaled to the range $(0 \to 1)$. The authors are arranged by clusters found in UPGMA. The hierarchical clustering structure is demonstrated by the dendrogram tree connecting author pairs together.

Figure 6.3 shows the result of generating $\mathbf{M}^{Auth.Sim.}$ and performing UPGMA hierarchical clustering. The dendrogram tree links authors pair-by-pair, illustrating how closely related clusters are. An enlarged dendrogram is shown below:

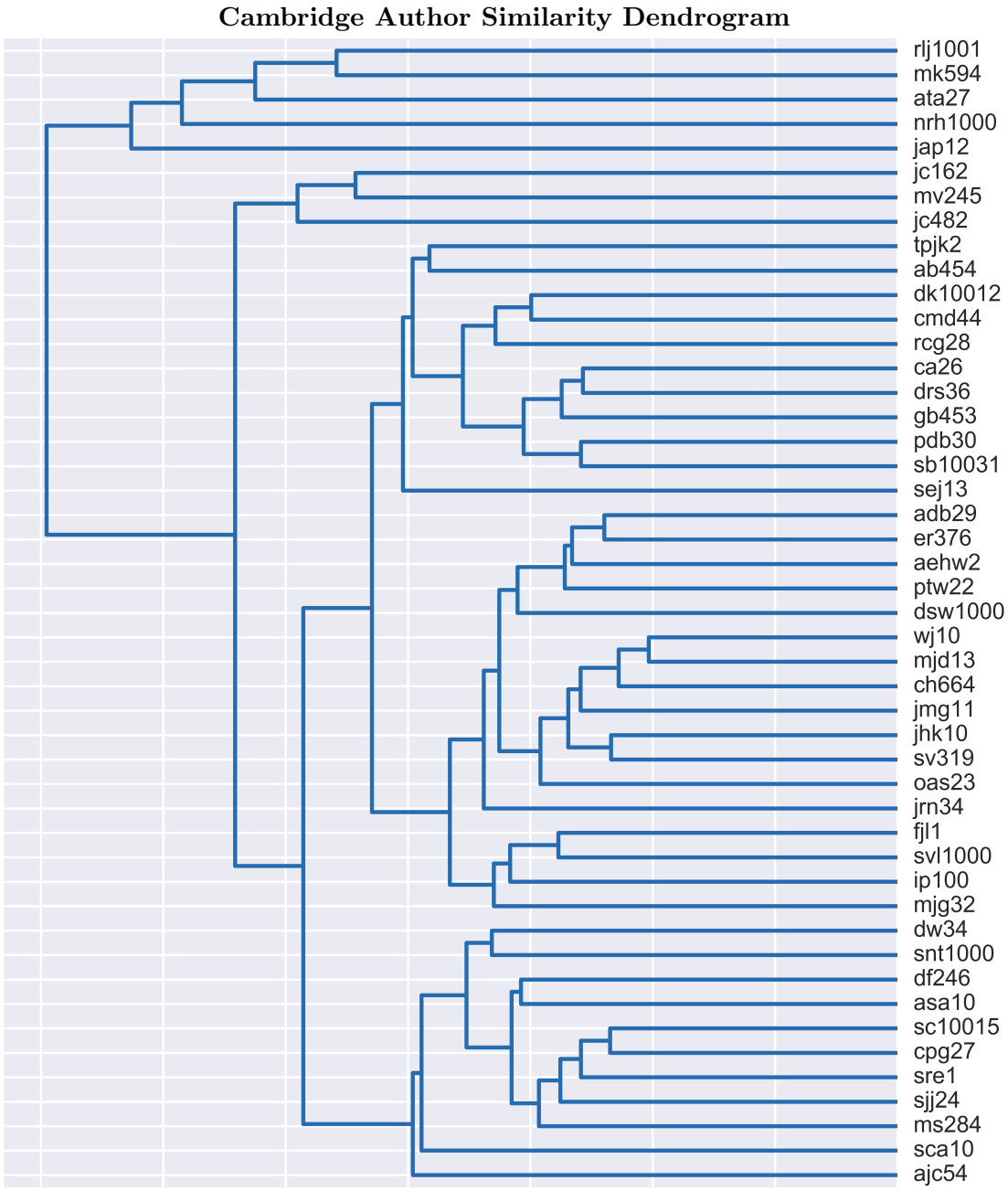


Figure 6.4: The dendrogram of figure 6.3
plotted for clarity

A striking feature of figure 6.3 is the cluster in the bottom-right corner. The dendrogram shows the members of this cluster occupy a separate branch of research space than the

rest of the department. The staff members involved[9] are all members of the Centre for Atmospheric Science. The unsupervised model thus successfully 'predicted' their department, and indicated that their work is separate from most of the Chemistry Department. This is a real success for the model. The dendrogram was then further examined and broken into distinct branches. Each branch was examined and manually labelled (see figure 6.5). Most clusters make intuitive sense, but there is a core of well-connected, more disparate members (wj10 to jrn34). These members could be interpreted as forming an interdisciplinary cluster.

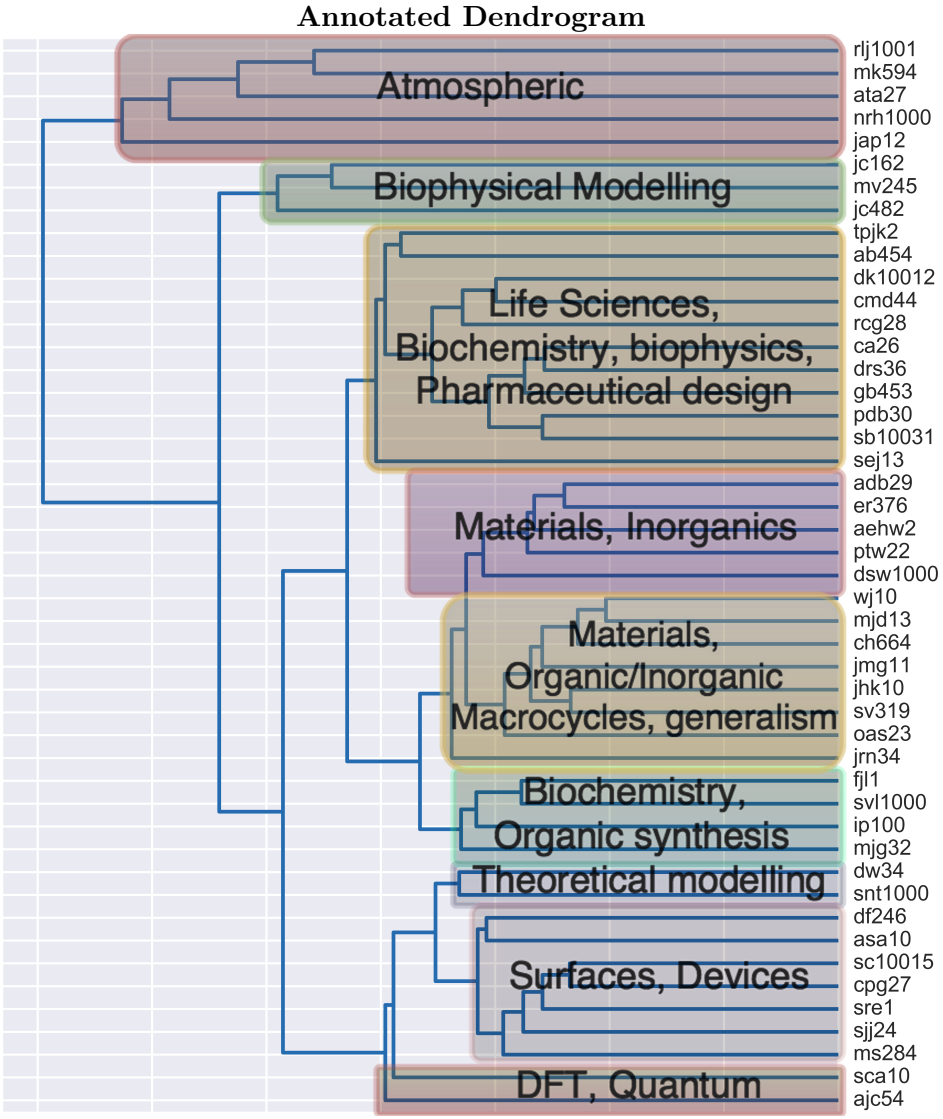


Figure 6.5: Cluster labels overlayed over the distinct branches of the dendrogram.

[9]Professors Jones and Pyle, Drs. Harris, Archibald and Kalberer

The value of this method is self-evident. Clustering staff members informs the department about the width of research (number of clusters), and how resources are partitioned (size of clusters). It should also be stressed that authors are associated without any human preconceptions/bias. Perhaps the most valuable author associations are the unexpected ones, and authors should be encouraged to examine their cluster and consider their 'neighbours'.

6.3 Combining research clusters and authors

As a final data examination, the topic communities found in §6.1 were linked to the staff members. Different metrics for author similarity were developed to investigate if they correlated with the maps produced in §6.2. Firstly, for a topic community $\mathfrak{C}$, with documents $d \in \mathfrak{C}$, and an author $\mathfrak{A}$ with documents $\delta \in \mathfrak{A}$, we can associate the author with the community if $\mathfrak{C} \cap \mathfrak{A} \neq \{\}$. The function f_{assoc} was defined as

$$f_{assoc}\left(\mathfrak{C}, \mathfrak{A}\right) = \begin{cases} 0 & \mathfrak{C} \cap \mathfrak{A} = \{\} \\ 1 & \mathfrak{C} \cap \mathfrak{A} \neq \{\} \end{cases}$$

It was noted that there was significant variation in the number of communities that researchers were associated with. A plot of $\sum_i f_{assoc}\left(\mathfrak{C}_i, \mathfrak{A}\right)$ for each author is shown below:

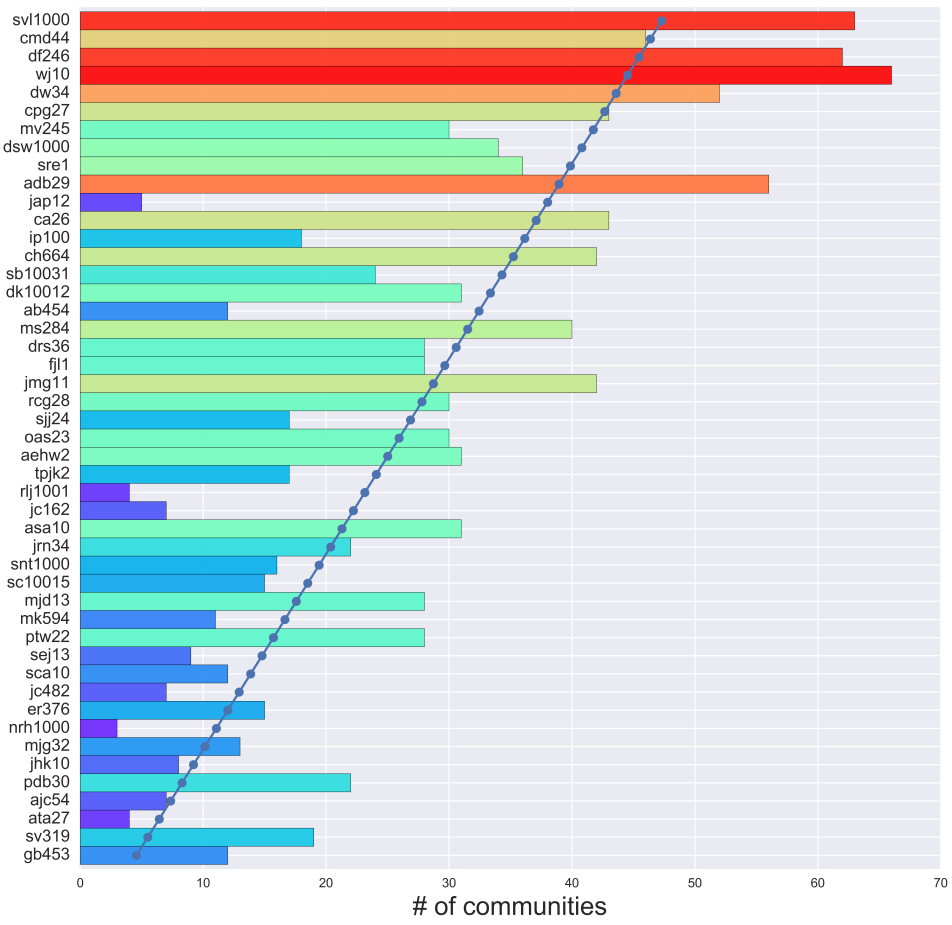


Figure 6.6: Number of research communities authors are associated with. High values (towards red) indicate an author publishing across many communities and suggest more interdisciplinary work, but also higher publication count per author. The authors are ordered by publication count (highest at top). As expected, there is a correlation between publication count and number of communities an author appears in. The blue dotted line-of-best-fit divided the authors into those that publish more widely given their publication count (bar to the right of the line) form those who publish more narrowly for their publication count (bar to the left of the line)

It can be seen that some authors were widely distributed between communities, whereas others were concentrated. It was noted that communities were not uniformly distributed.

For example, there were many communities in 'Life Sciences' but few in Atmospheric Chemistry, as such, interpretation of high values in Figure 6.6 directly corresponding to wide research interests should be tentative[10].

An association metric $S_{coincidence}$ between authors $\mathfrak{A}$ and $\mathfrak{B}$ was then defined as

$$S_{coincidence}\left(\mathfrak{A},\mathfrak{B}\right)=\sum_{c}^{C}\left(f_{assoc}\left(\mathfrak{C}_{c},\mathfrak{A}\right)\times f_{assoc}\left(\mathfrak{C}_{c},\mathfrak{B}\right)\right)$$

Where C is the total number of communities. An author association matrix was created, $\mathbf{M}_{\mathfrak{A},\mathfrak{B}}^{Auth.Coinc.}=S_{coincidence}\left(\mathfrak{A},\mathfrak{B}\right)$, where high values for author pair $\mathfrak{A},\mathfrak{B}$ indicate they appear in many research communities together. The matrix was then scaled such that: $\mathbf{M}_{\mathfrak{A},\mathfrak{B}}^{Auth.Coinc.scaled}=\mathbf{M}_{\mathfrak{A},\mathfrak{B}}^{Auth.Coinc}/\left(\mathbf{M}_{\mathfrak{A},\mathfrak{A}}^{Auth.Coinc}+\mathbf{M}_{\mathfrak{B},\mathfrak{B}}^{Auth.Coinc}\right)$, and normalised from $0\rightarrow1$. This was a measure of how often authors published in the same communities. The matrix is shown below:

[10]It should also be noted that this method (shown in figure 6.6) treats strong connects equally to weak connections, i.e. 100 papers published in one community is just as strong a connection to a single paper published in a community

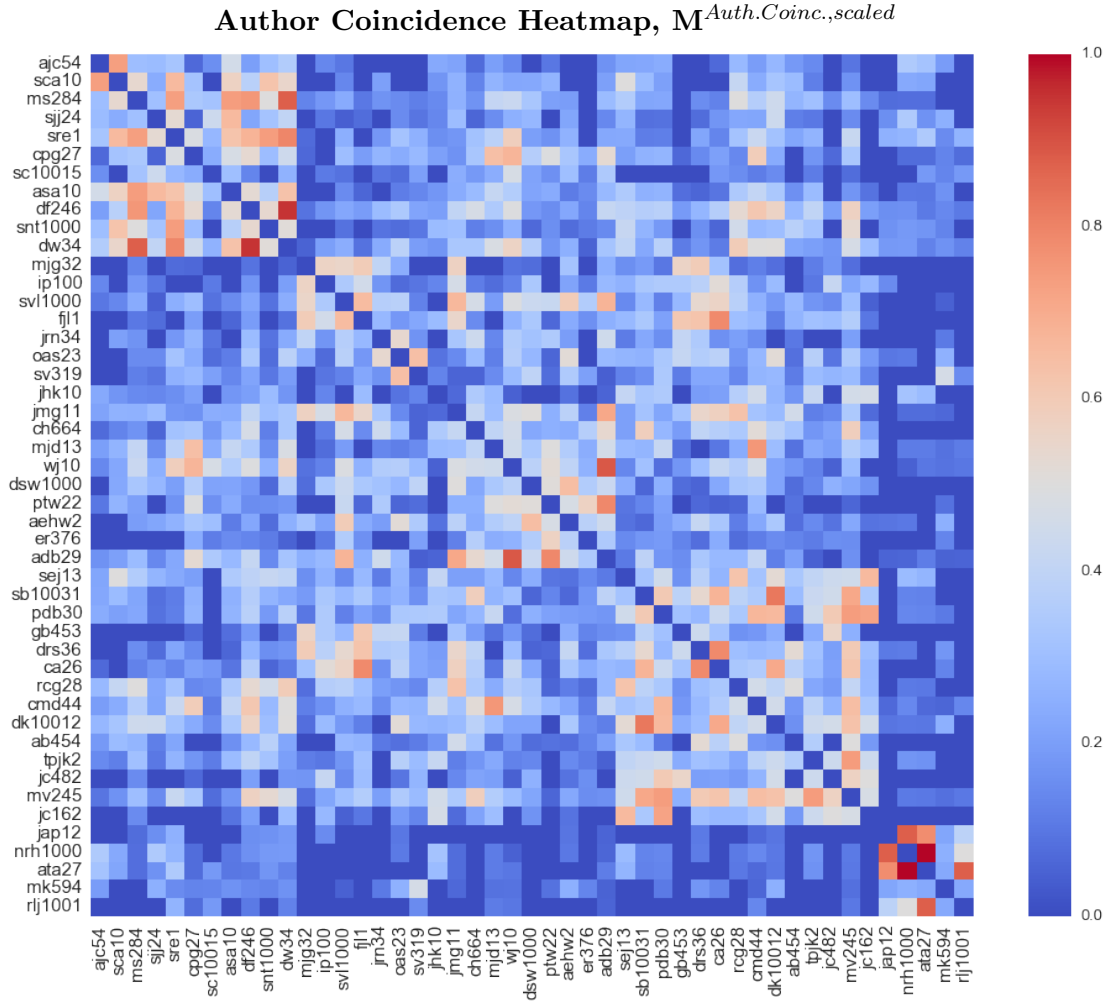


Figure 6.7: Heatmap showing author-author pair values for how often author pairs publish works in the same communities. High values indicated that authors are predicted to have similar publication profiles. Note the authors are arranged with the ordering from figure 6.3.

Figure 6.7 displays where authors have similar research community occupations. High values should indicate that authors should ideally collaborate/communicate because they publish in the same research communities. Note also the square patterns of high values close to the diagonal of the map reproduce the clustering in figure 6.3, lending weight to the validity of both analyses.[11].

[11]This is because the heatmap has been arranged according to the clustering found in §6.2, but the matrix in figure 6.7 is derived with a completely different method (without applying any clustering algorithm to authors). As clustering is qualitatively visible in figure 6.7, there is a correlation between the two methods, i.e. they are consistent

Having defined a framework for finding shared research interests, the next step was to find where authors were *actually* collaborating. It was possible to identify approximately 700 documents in $\Delta 7$ that were co-authored by staff members. A heatmap for co-authorship between authors is shown below, $M^{Raw\ Collab.}$[12], as well as a metric equivalent to the $\mathbf{M}^{Auth.Coinc.scaled}$ with elements as the sum of the number of communities in which both staff members have co-authored, $M^{Community\ Collab.}$.

Author Collaboration Heatmap, $M^{Raw\ Collab.}$

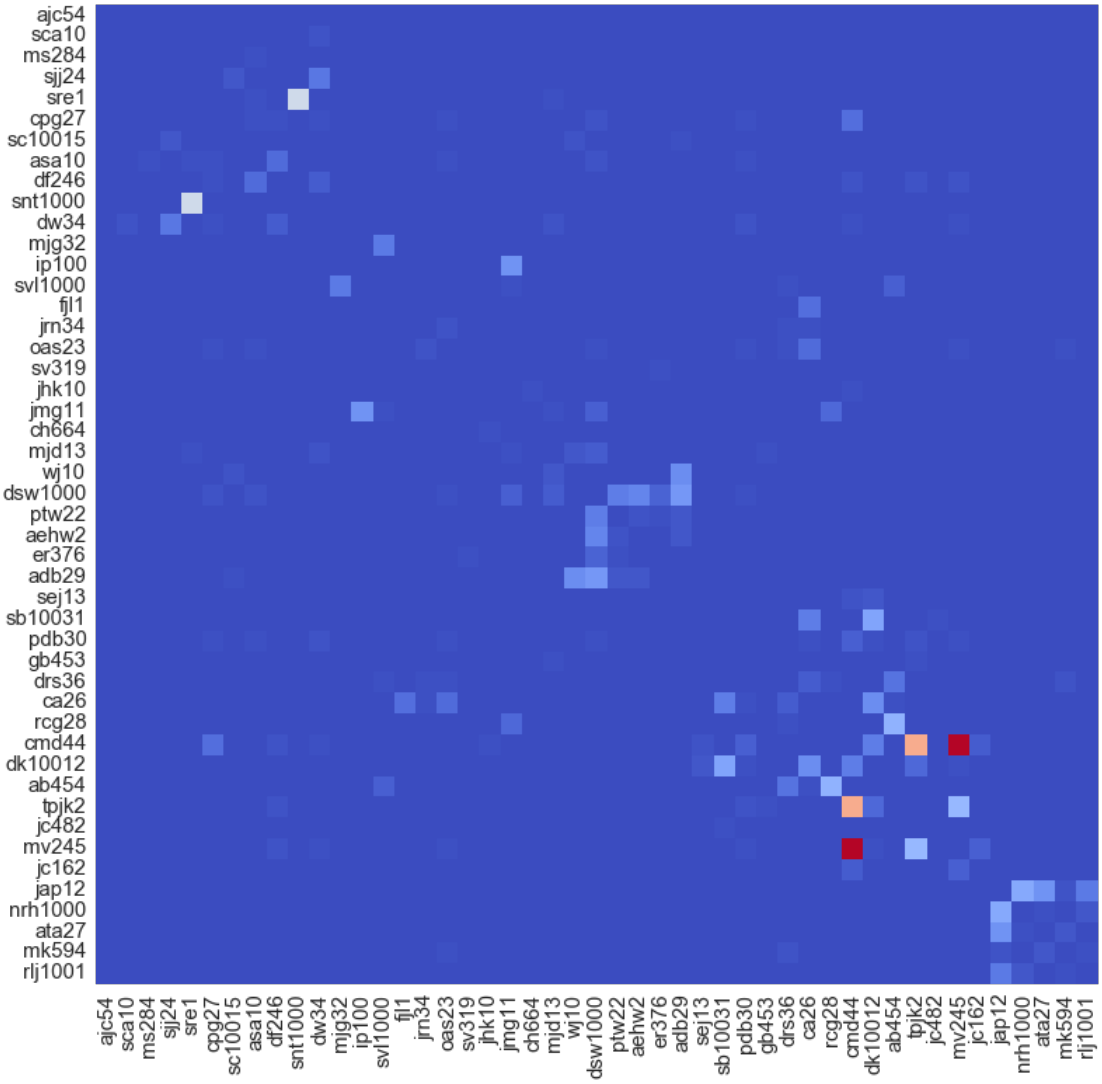


Figure 6.8: Raw collaboration matrix (values scaled to range $0 \rightarrow 1$). Note the general lack of co-publishing between staff members. Again staff are ordered by clustering described in §6.2, but no actual clustering has been performed. Hot spots near the diagonal suggest that author pairs clustered close together in §6.2 generally collaborate more than distant author pairs.

[12]Elements of $M_i^{Raw\ Collab.}.j$ are set to the number of times the authors have co-authored

Community-summed Author Collaboration Heatmap, $M^{Community\ Collab.}$

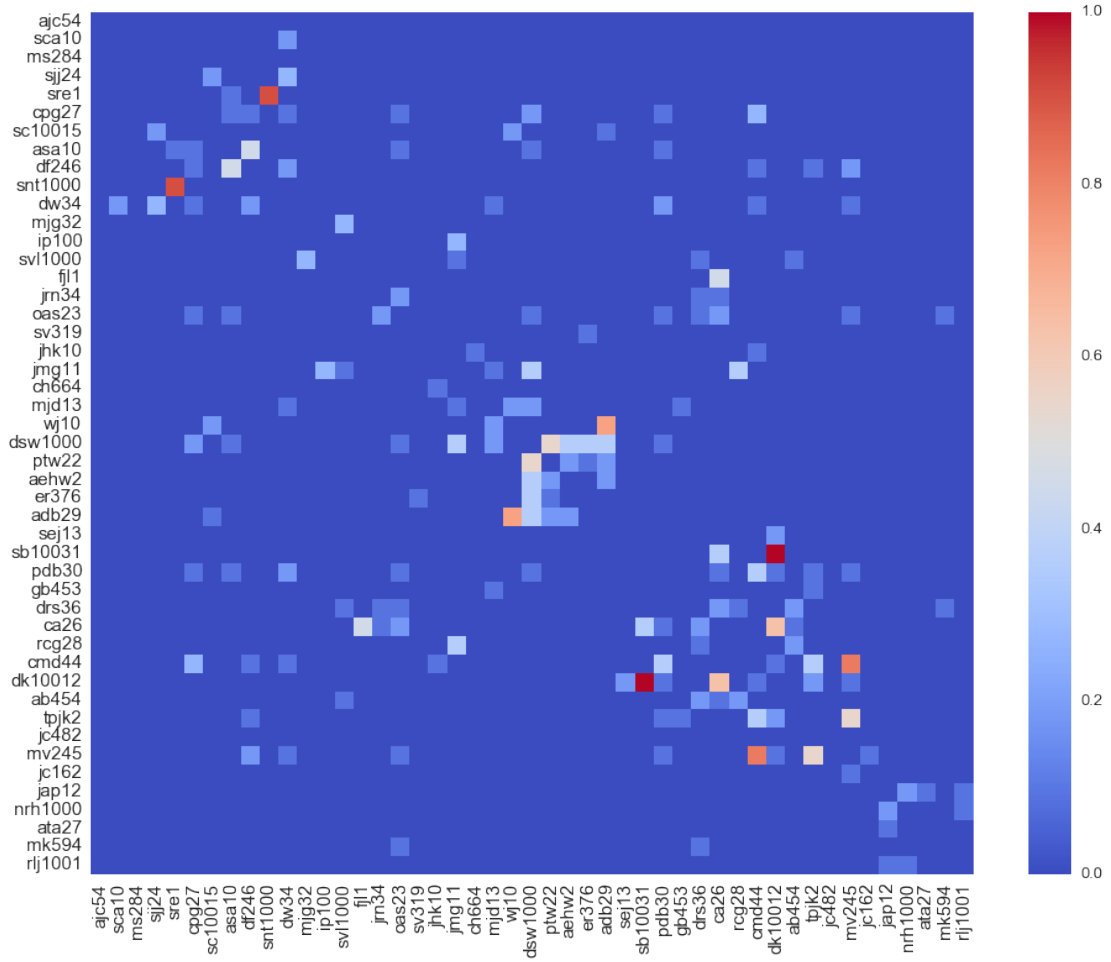


Figure 6.9: Matrix formed by summing collaboration of author pairs over research communities (values scaled to range $0 \rightarrow 1$). Qualitatively similar to figure 6.8. Hot spots near diagonal again suggest authors closely clustered in §6.2 collaborate more frequently.

Both maps show similar qualitative pictures. Similar author pairs (close to diagonal) are more likely to collaborate.

As a final data step, a matrix defined as the difference between an author similarity matrix e.g figures 6.3 ($M^{Auth.Sim}$), 6.7 ($M^{Auth.Coinc.}$) and an author collaboration matrix e.g. figures 6.8 ($M^{Raw\ Collab.}$), 6.9($M^{Community\ Collab.}$) could be interpreted as a *recommended collaboration matrix*, [13] Author Pairs with values to 1 should be encouraged to consider working together. Matrix $M^{Recommend}(=M^{Auth.Coinc.} - M^{RawCollab.})$ is one

[13] i.e. where values close to 1 indicate high similarity but low evidence of collaboration, values close to 0 indicate effective collaboration and values close to -1 indicate high collaboration but low author similarity.

possible example, shown below:

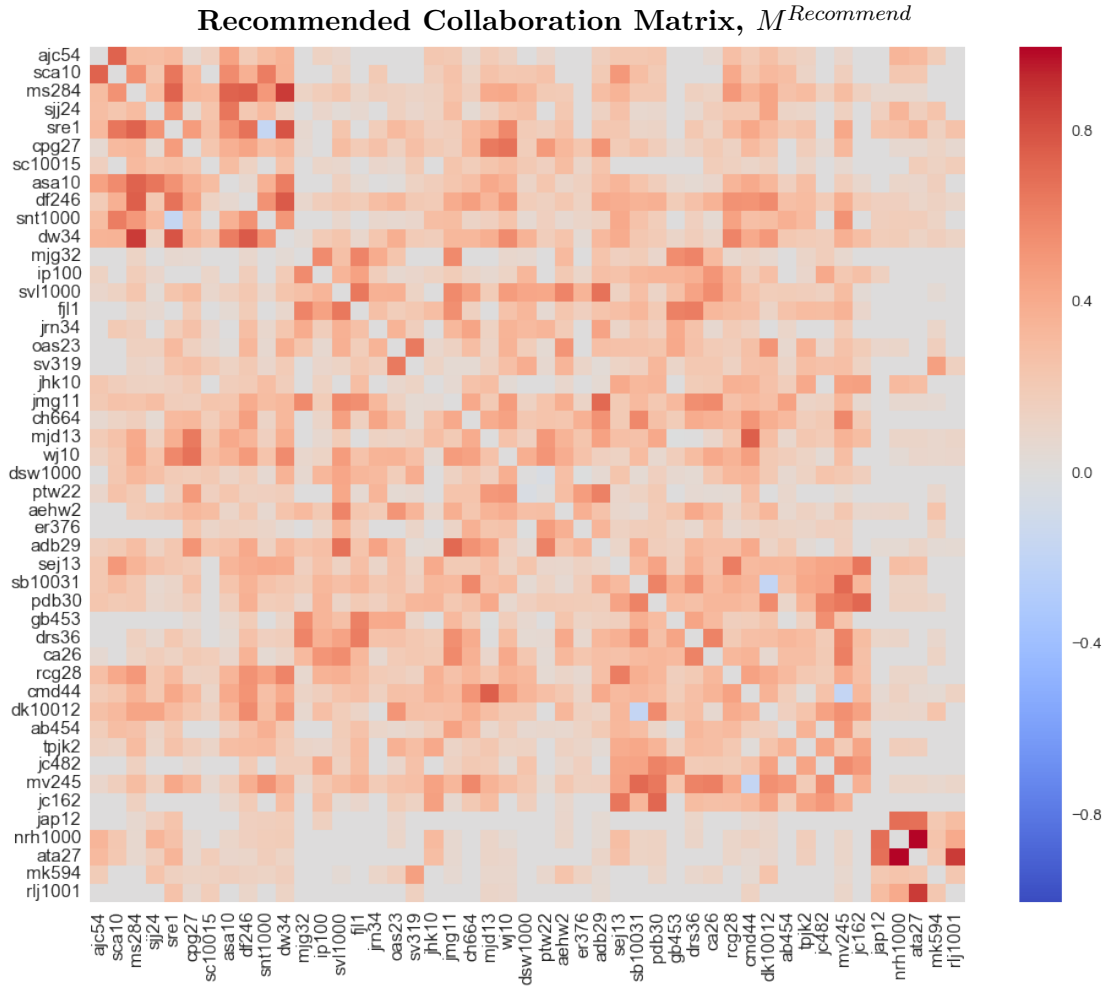


*Figure 6.10: High values (Deep red) indicate authors that have similar research but for which there is little evidence of collaboration on published works. Values near 0 (grey/white) are where authors are **neither** similar **nor** collaborate, or **are** similar **and** collaborate closely. Values towards -1, (Blue) indicate authors that are collaborate but do share similar research (not strongly observed, as expected. High negative values would be somewhat paradoxical.)*

This final piece of the analysis section illustrates how the framework developed over the research project reveals where it might be profitable for authors to collaborate. Table 6.2 shows the top 20 scores in $M^{Recommend}$, where there is stronger evidence to suggest these author pairs *should* collaborate but little evidence was found that they *are* collaborating[14].

[14]Please see §8.5 for a brief exploration of the the table

Table 6.2: Community 275

Rank	Author CRSID	Author CRSID	Recommended Collaboration Matrix Score
1	ata27	nrh1000	1.000
2	dw34	df246	0.916
3	dw34	ms284	0.875
4	rlj1001	ata27	0.875
5	dw34	sre1	0.795
6	adb29	ptw22	0.765
7	cmd44	mjd13	0.757
8	df246	ms284	0.755
9	ca26	drs36	0.753
10	sca10	ajc54	0.737
11	sre1	ms284	0.737
12	adb29	wj10	0.736
13	mv245	pdb30	0.731
14	asa10	ms284	0.730
15	jc162	pdb30	0.724
16	adb29	jmg11	0.714
17	mv245	sb10031	0.713
18	ca26	fjl1	0.708
19	df246	sre1	0.679
20	adb29	svl1000	0.676

The matrix row of $M^{Recommend}$ for a particular staff member (Professor Goodman) is plotted below by way of example of what the model considers a staff member's recommendations to be.

Recommended Collaboration Scores for Particular Staff Member, Professor Goodman

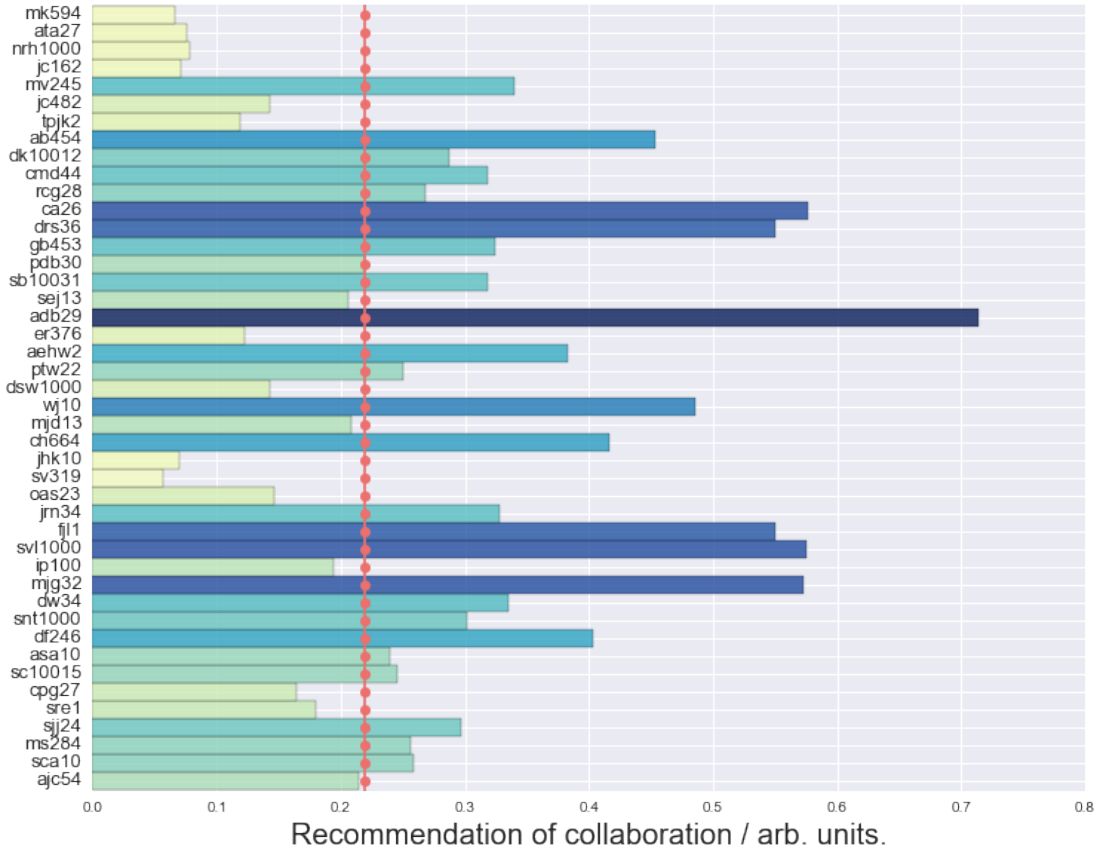


Figure 6.11: Recommendations for a particular staff member from the recommendation matrix, plotted in bar form, (Professor Goodman). (Bars very close to zero have been removed). Colour is a guide for the eye. Large values (Long bars, deep blue) indicate the author publishes in many similar communities to Professor Goodman, but little evidence was found of collaboration, thus the recommendation to collaborate. Low values (short bars, towards yellow) indicate 'appropriate' collaboration (either similar work and collaboration, or dissimilar work without collaboration). The red vertical line indicates the mean *value in the Recommended Collaboration Matrix. Bars breaking through the line to the right are higher than the mean, and thus are where new collaborations should be considered.*

The aim is that these maps and plots may trigger constructive debate, and promote effective collaboration in the department. [15] It should also be noted that the evidence for collaboration is from quite a small sample, and the collaboration metric could be improved by considering other factors than just co-authorship [16].

[15] The analyses presented in this section are not exhaustive, and there is potential for more fruitful insights to be found. Please see §8.1

[16] Please see appendix §8.5 for further exploration of collaboration metrics

7. Conclusions

Focussing first on the data acquisition phase, the scraping procedure was regarded as a modest success. The volumes of data collected from the UK chemistry departments was respectable, as was conversion rate from the potential results to fully-resolved records (72.9% to give 16363 records). The actual number of articles from UK chemistry departments can be confidently predicted to be considerably larger. The limited harvest could be down to the input list of scraping websites being too small. The procedure to identify webpages for scraping was limited where the departments did not host their own website, precluding large parts of many important departments. However, the data that was successfully resolved was of high relevance, with few false-positive inclusions. The scraping program was robust and efficient.

The data collected in global scraping was sufficiently populous and chemistry-specific to enable effective models to be trained. It should be highlighted all the datasets were created from freely-available sources, requiring no subscription. This said, it must be acknowledged that the publisher banning was considered as a major failure in the project. However, it was dealt with swiftly, and did not present a lasting issue. [1].

It should be mentioned that there are existing meta-data stores available (such as PubMed). Whilst using one of these datasets would certainly have been easier, there was no real available chemistry dataset with enough *breadth* of data. $\Delta 6$, whilst taking considerable time and effort to create, was heterogeneous and thus a more suitable tool.

The algorithmic development section can be regarded as successful. The premise of quantitative vectorial representation of articles was realised, especially by the Doc2Vec model. It should be mentioned the TF-IDF models failed to produce effective vectors, which is not well understood. The power of the model can begin to be seen in §6, where clustering performances were intuitive and instructive. Some model design choices may have limited specificity, such as the decision to use 100 dimensional vectors.[2].

The analysis that was performed is most interesting, but the usage to chemists is some-

[1] The author wishes to thank the librarians and Professor Goodman addressing this problem so efficiently

[2] Higher dimensional vectors have been shown to perform better [18]

what limited. As a chemical project, it should have been a strong focus to produce results directly useful to chemistry. This was achieved to some extent towards the end of the project, but this point was reached probably slightly too late.

Some further useful applications of the methodologies have been alluded to, but most of these take the form of a *service* rather than concrete universal insight. Whilst the author would be enthusiastic to implement some of these services (on-demand similarities, clustering, recommendations of articles to read, research profiling etc.), the project scope had to be limited at some point[3].

It is concluded that the aims set out in this project have been addressed, and there were no major barriers preventing the fulfilment of the project brief.

[3]It is the author's opinion that another project could be filled developing further uses of the dataset and extending the methodologies presented

8. Appendix

8.1 Recommendations for Further Work

As alluded to in the text, there are several recommendations for further work. The code and data will be improved and amended over time, and is freely available under MIT licence on request [1]. If attempting to carry out further work on this project, it is recommended to contact the author for in-depth explanations. This list is by no means exhaustive, and it is the author's belief that literature semantic analysis should be considered an important analytical chemical tool.

8.1.1 Greater Dimensionality and Training Improvements

The principles behind the methods discussed in the project have been shown to be sound. Models should now be improved. Computing resources should be obtained to train higher dimensional vectors [2]. The models should also be trained for longer (> 24) epochs on more data (> 460000 documents). These steps will lead to more expressive models.

8.1.2 Greater use of word vectors

This project focussed mainly on document vectors. However, word vectors may be very useful. A method for testing the quality of improved models should be developed. This could take the form of expected relationships to test the model: e.g. Fluorine is to Fluoride as Chlorine is to ... . Many hundreds of these relationships should be systematically built up to test model intuition.[3] This follows the methodologies set out in the literature [18] [19]. Furthermore, is it possible to predict chemical properties using semantic relationships found in the literature? Vec(Compound A) + Vec(Compound B) + Vec(Lab Technique) may give vec(Product C). If so, it may be possible to find

[1] A digital copy is included with this dissertation.

[2] The author recommends 400 dimensional vectors

[3] This would probably require much larger, more descriptive training sets, e.g. textbook transcipts etc.

unexpected reactions. This could be coupled with the RInChI database to form a new type of data-driven cheminformatics.

8.1.3 Time resolution in clustering

Methods have been described for clustering documents. The cluster centres represent the content of the cluster effectively. By finding early papers in the cluster, is it possible to identify influential papers or authors? By clustering on documents from particular years, is it possible to identify a path for the evolving cluster centre vector? If so, it should be possible to extrapolate to *predict* near future research directions.

8.1.4 Open Source Chemistry Vectors

With the increase in open source papers, it should be possible to build up a vast dataset of chemical language for training, using the bodies of articles published on open source platforms, and even to use supplied supporting information.

8.1.5 Structure stemming

Chemical names could be smartly preprocessed to classes of chemicals, for example by identifying a compound from its name and mapping to InChI key, then to a chemical class. This would allow better association of chemical fragments in training.

8.1.6 Multiply labelled Documents

In Training Doc2Vec, by specifying document with more than just their unique identifiers allows more vectors to be associated. By identifying and labelling all documents with a particular concept, e.g. 'palladium-catalysed', and then training Doc2Vec, one defines an 'palladium-catalysed' vector, specifically trained for the concept. These concept vectors would be robust and information-rich[4]

8.1.7 TSNE Maps

There was not sufficient time to explore the clustering found by TSNE reductions. TSNE is a very popular technique in the current machine learning Literature, and should be investigated more thoroughly. Clusterings found by TSNE should be subject to similar methods of study to those performed in this project. K-Means clustering performed on the TSNE maps was briefly investigated before more rapid progress could be made by

[4]e.g. which documents are close to the indium-catalysed vector but do not contain the word indium...

other techniques. There was evidence to suggest that TNSE K-Means clustering was potentially useful, but time did not permit investigation of this technique.

8.2 Technical Details

In the interest of future work, this section details the technical details of artefacts provided with this project.

8.2.1 Code Artefacts

The python code used in this project was written in a largely self-documenting style. The time limits did not permit for professional doc-strings to be produced, or for anaconda packages to be provided, but the code is well commented. There is also a comprehensive set of `Jupyter Notebooks` as tutorial guides for using the code[37]. The core code has been presented in a 'package' style. The module was named `fruitbowl` with five submodules,

- `Cherry` for operations concerning scraping and data collection.
- `Orange` for operations concerning NLP corpus creation and big data memory-friendly streaming
- `Strawberry` for operations concerning Word2Vec and Doc2Vec model Training
- `Apple` for operations concerning analysis of trained models (visualisation, export management etc.)
- `Pomegranate` for operations interfacing with Gephi and community generation.

There are approximately 30 python source files included in the module. If using the code it is recommended to read and adapt the jupyter notebooks `Fruitbowl Example 1.ipynb` to `Fruitbowl Example 3.ipynb`. It is recommended to write code in a directory that contains the `fruitbowl` module. The Module is free to distribute and adapt under the MIT licence, which must be included in any copy. The list of dependencies required for fully functional behaviour for the `fruitbowl` suite is as follows:

- `Python 2.7` Developed on Python 2.7.11 (recommended version)
- Python 2 external modules required:
  - `matplotlib 1.5.1` Plotting modules [38]
  - `Seaborn 0.7.0` Extension to plotting modules and data analysis [2]
  - `numpy 1.10.4` Computational Library [39]
  - `Scikit-Learn 0.17` Machine learning library [35]
  - `Scrapy 1.0.3` Scraping framework
  - `Gensim 0.12.2` Natural Langauge Processing library [29]
  - `nltk 3.1` Natural Language ToolKit library [31]
  - `pandas 0.17.1` Data analysis and management library [40]
  - `pymongo 3.0.3` Python driver for MongoDB database
  - `requests 2.9.1` Web scraping library
  - `scipy 0.17.0` Scientific computing library [3]

  - `jupyter 1.0.0` Jupyter notebooks will be required to use the tutorial notebooks.
- `JDK` Java Development Kit - for Gephi graph analysis via gephi api
- `apache-maven-3.3.9` Java dependency manager - for Gephi graph analysis via gephi api
- `C Compiler` for use in BHTSNE reductions[34].
- `mongoDB` The program was built around use of MongoDB. Not strictly necessary but strongly recommended. Recommended versions >3.2.

8.2.2 Data Artefacts

Data used in the project was dumped from their mongo databases is also supplied in .json format. The data provided is as follows:

- `Delta1.json` : These are the DOIs found in the UK scrape
- `Delta2.json` : These are the complete meta-data results found in the UK scrape
- `Delta3.json` : These are the DOIs found in the global scrape
- `Delta4.json` : These are the complete meta-data results found in the global scrape
- `Delta6.json` : This is the data used for training and analytical purposes in the project
- `Delta7.json` : The subset of $\Delta 6$ from Cambridge used in §6
- `cbow_model` : Gensim binary saved model for final cbow Word2Vec model used in the project
- `sg_model` : Gensim binary saved model for final skipgram Word2Vec model used in the project
- `FULL_DOC2VEC` : Gensim binary saved model for final Doc2Vec model used in the project

Note $\Delta 5$ is not provided to save disk space (It is simply $\Delta 2$ combined with $\Delta 4$).

8.3 Word Vector Analysis of Chemical Elements

As an investigation into the utility of word vectors trained, it was decided to briefly investigate the word similarities between chemical elements. This analysis is included as an appendix as there was not sufficient space for it to be included in the main body, and due to its self-contained nature. It was hoped that this short investigation would provided evidence that methods sketched in §8.1.2 could work.

The similarity matrix was produced for chemical elements mentioned in the text corpus (115 out of 118 known chemical elements). This required mapping both chemical names and symbols together (e.g. `florin`[5] and `F` for fluorine) to represent the concept vector for the elements in question.

A modified data sanitation pipeline was created to substitute the chemical symbol for the chemical name. This was only done for chemical symbols longer than 1 letter to dissuade conflating different concepts to the same word vector (`S` could represent Sulfur or a stereochemical label.)

A CBOW model was trained using this modified input data with the same presets as the main CBOW model, detailed in table 4.3. The Cosine Similarity matrix was produced for the 115 elements found in the corpus. UPGMA clustering was performed[35], as well as graph visualisation with modularity clustering [4],[5]. The dendrogram of the UPGMA clustering is shown in figure 8.1. The process identified 5 main branches:

- The gold region includes a sub-branch of noble gases, the other branch mainly actinoids.
- The magenta region contains non-metals mostly associated with organic compounds
- The cyan region contains mainly metalloids, actinoids and alkali metals
- the red region contains mainly transition metals
- The green region contains almost exclusively lanthanoid metals

[5]Fluorine is stemmed to fluorin by the stemming process (§4)

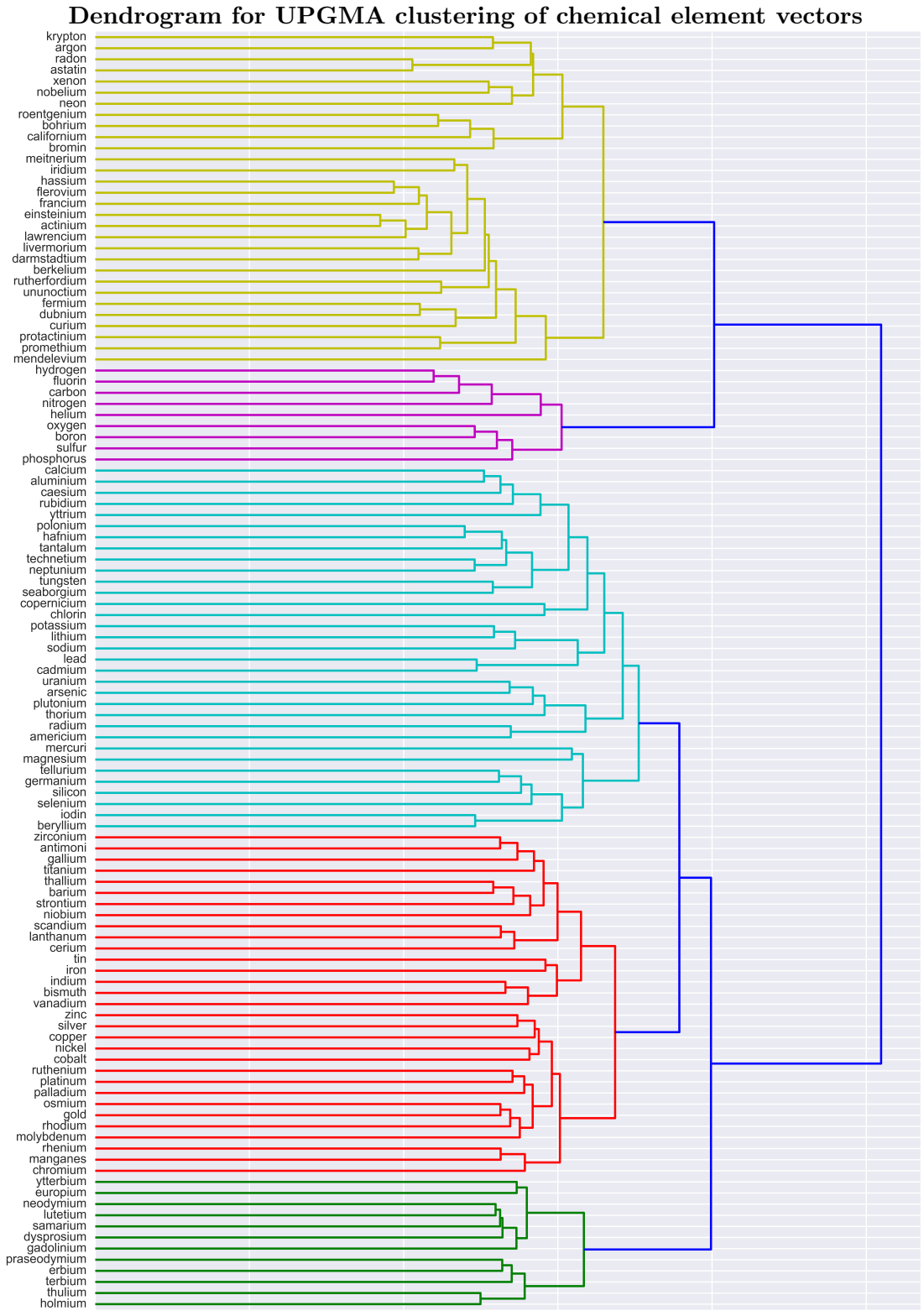


Figure 8.1: Dendrogram for UPGMA clustering of chemical element vectors. Colours indicate distinct branches

The dendrogram shows that the classifications broadly fall into intelligible categories within the periodic table. There are, however, some surprises, especially the halogens, with bromine in the actinoid subbranch, and chlorine associating with copernicium. This may be because the symbols Cl and Cn occur together often in the literature due to mentions of carbon and nitrogen, not copernicium. Similar reasoning can be used for bromine (`cf br` could refer to a CFC rather than californium and bromine) This exposes a flaw in the symbol/name association process that could be tackled in further work.

The graph visualisation is shown in figure 8.2 (Also the front cover of this dissertation). Period 7 was removed from this graph as there were too few mentions in the corpus for reliable vectors[6], and to remove cluttering nodes.

[6]Uranium was kept as it had non-negligible corpus mentions

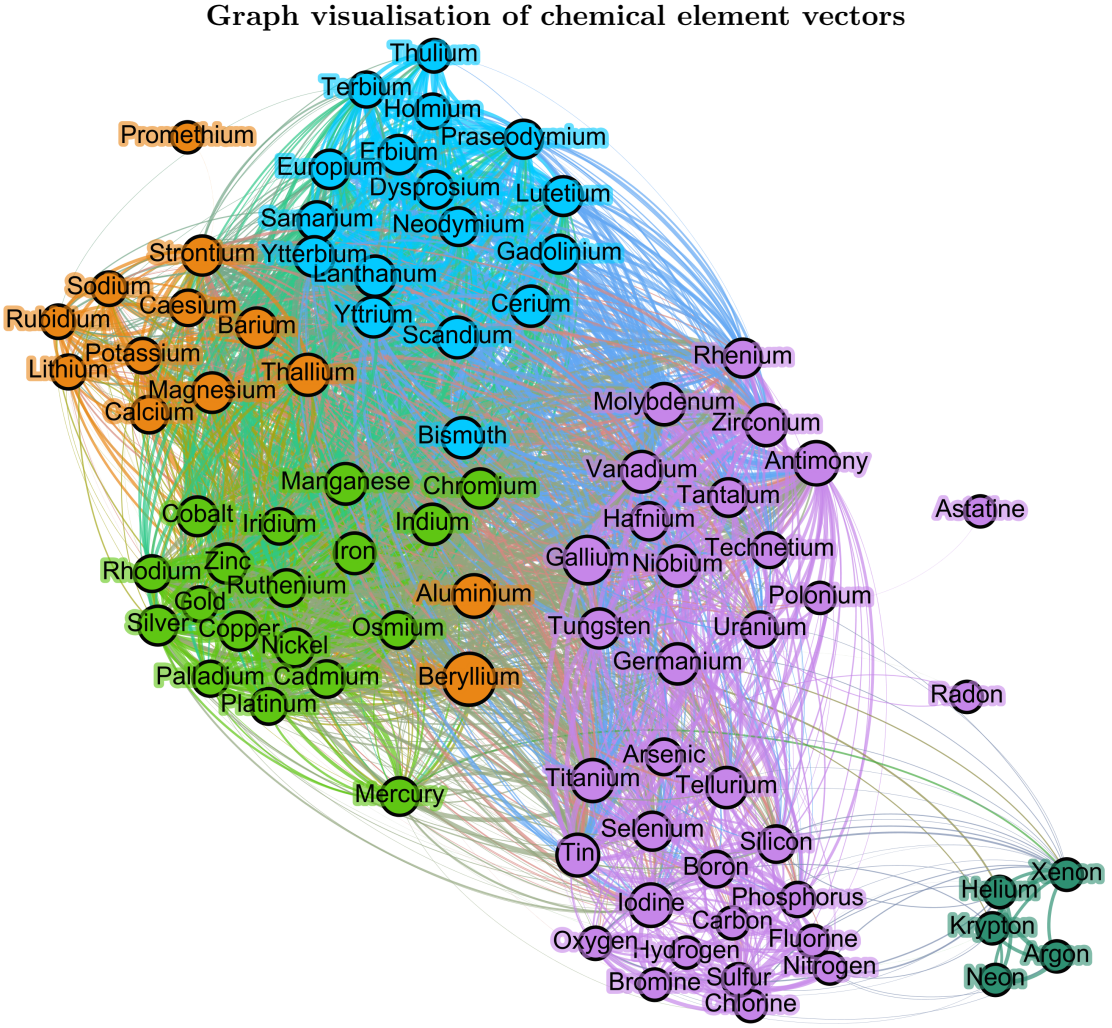


Figure 8.2: Nodes are coloured by their communities, and are spatially arranged by modelling edge weights as springs. Node sizes are proportional to their connectivity.

5 distinct communities were identified:

- The Orange community contained mainly alkali metals and alkali earth metals
- The Blue community contained mainly lanthanoids
- The Light Green community contained mainly transition metals
- The Purple community separated into two spatially distinct regions. The northern region generally contained transition metals and metalloid, and the southern region contained organic non-metals.

- The Dark Green community contained noble gases.

The community finding process reflects a similar situation to the UPGMA process, but is perhaps more successful. The removal of the actinoids appears to have improved community finding. The community finding process was repeated with period 7 included, resulting in broadly the same communities, but with bromine and chlorine leaving the purple community to join a a loose community of actinoids, however they remained strongly associated with each other. The degree of connectivity between nodes is similar for most nodes, but larger for some nodes, e.g. beryllium, which is difficult to interpret.

Attention was turned to a practical example. Palladium is used widely in catalysis but is rare and expensive, and alternatives would be economically and environmentally beneficial[41]. With this in mind, the cosine similarities of all the elements to palladium vector were computed, and a selection of metallic elements with high similarity is shown in figure 8.3.

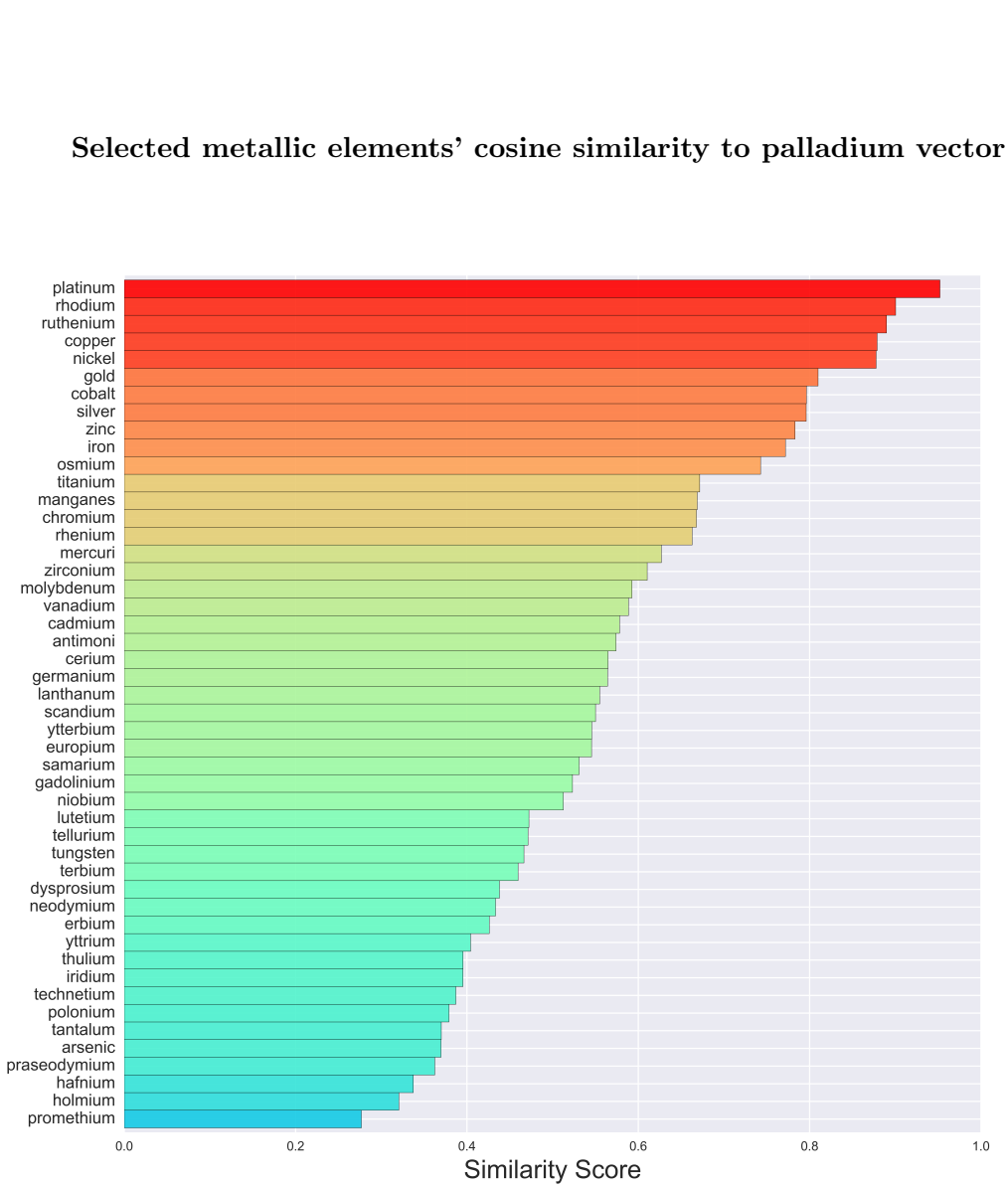


Figure 8.3: The metals are ordered to to bottom from most similar (long bars, red) to low similarity (short bars, blue). The colours are a guide to the eye.

Platinum, rhodium, ruthenium, copper and nickel all had very high scores. The models could be interpreted as suggesting that these metals have similar properties to palladium. This is very much the case for platinum and rhodium (pd, pt and rd are all platinium group metals)[42]. Nickel and copper are predicted to be similar to palladium, and there is evidence that nickel could be used for some palladium-catalysed reactions[43], whereas copper is often combined with palladium to form more effective catalysts[44]. Thus it

could be argued the models suggest that more attention should be focussed to nickel catalysis.

This analysis, whilst brief, is promising. This lends weight that more in-depth considerations of word vectors and concept vectors would be fruitful.

8.4 Finding Unexpected Links

This section briefly re-examines community 275, discussed in §6.1. There was not room for this section to be included in the main body, so is included as an appendix.

Table 6.1 details some of the contents of community 275. Most of the articles in community 275 were published by members of staff who are no longer in the department. The only authors currently in the department are Dr. Kalberer (1 out of 15 articles) and Dr. Vignolini (3 out of 15 articles). These two authors work in very different fields, but are connected here. Dr. Vignolini has worked on plant microsctructures (including pollen) and Dr. Kalberer has worked on atmospheric affects of pollen particles. It could be argued that these two researchers could benefit from discussing each others' work. The program has thus found an unexpected, non-obvious link between these researchers. These unexpected links can be extracted as follows:

The co-occurance of authors is represented in $\mathbf{M}^{Auth.Coinc.}$(figure 6.7 in §6). To emphasise where authors often appear together in communities but do not collaborate, a more extreme recommendation matrix could be defined, by setting values in $\mathbf{M}^{Auth.Coinc.}$ to zero if the author pair have ever collaborated.

Because of the UPGMA clustering, the ordering of the authors in the matrix reflects their similarity (authors are next to similar authors), so that pairs near the diagonal are close. If we select the high values that are distant from the diagonal, these are the more 'unexpected' pairings. Setting a minimum distance from diagonal of 10 authors (11 was the population of the largest dendrogram branch identified in 6.5, a distance of 10 ensured author pairs in the same branch were not included), the following heatmap of 'unexpected' links is created, $\mathbf{M}^{Unexpected}$.

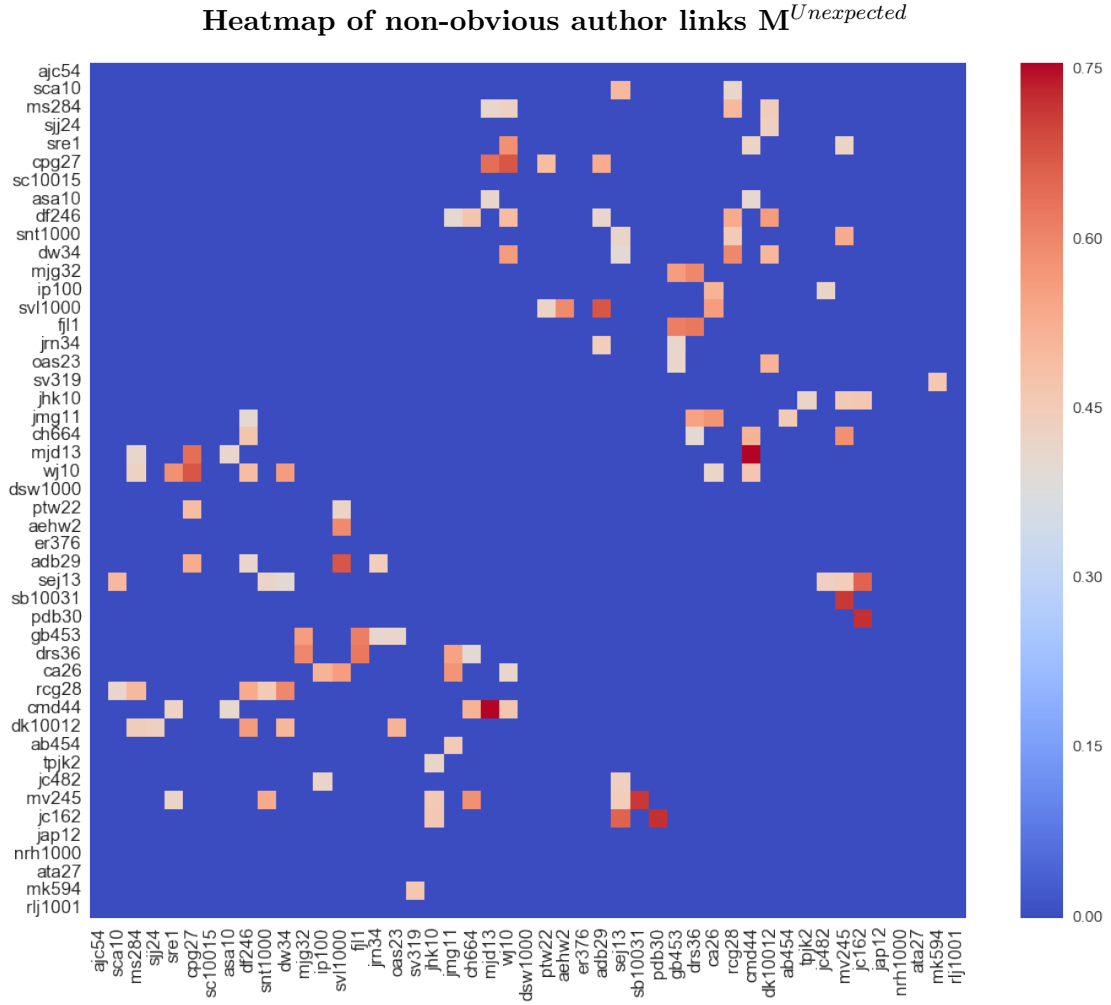


Figure 8.4: High Values (close to 1, red) indicate that the author pair publish in the same communities but there is no evidence of collaboration (no co-authorship detected). The author pair must also be a distance of 10 authors from the diagonal. Low values indicate low similarity and low evidence of links, or links are 'obvious' and therefore not important

The top 20 'unexpected' results are shown in table 8.1. They are ordered by their distance from the diagonal, to attempt to highlight high scores but also the mosts 'unexpected' links.

Table 8.1: Top 20 results from $\mathbf{M}^{Unexpected}$

Author 1	Author 2	Link Score	Distance from Diagonal
dk10012	df246	0.565	28
rcg28	dw34	0.598	24
drs36	mjg32	0.598	21
mv245	ch664	0.583	20
ca26	svl1000	0.561	20
gb453	mjg32	0.560	20
drs36	fjl1	0.625	18
wj10	sre1	0.583	18
wj10	cpg27	0.674	17
gb453	fjl1	0.613	17
mjd13	cpg27	0.641	16
cmd44	mjd13	0.757	14
adb29	svl1000	0.676	14
ca26	jmg11	0.576	14
jc162	sej13	0.656	13
drs36	jmg11	0.550	13
aehw2	svl1000	0.596	12
wj10	dw34	0.564	12
jc162	pdb30	0.724	11
mv245	sb10031	0.713	11

Due to the nature of research in the department, even these 'unexpected' pairings can be rationalised. This analysis could be taken further to find the most surprising results using more sophisticated or different techniques. This analysis could be generalised to larger sets of researchers, not just those in the department. The analyses presented in this section and §6 are intended to be useful in their own right, but mainly to serve as the basis for further sophistication.

8.5 Comments on Recommended Collaboration Table

In this section, the suggested collaboration table 6.2 is briefly explored to rationalise some of the suggested pairings. Limitations should be explained. The recommended collaborations arise from a balance of two factors: how similar the pair are, and how often they have collaborated in the past.The evidence that members are collaborating is taken from `http://www.ch.cam.ac.uk/publications/authors`, finding articles that are in $\Delta 7$ and considering authors to be collaborating if they co-author papers. This is not a particularly robust metric, as there were only about 700 co-authored paper found in this process, a small sample. This is why the main body of §6 refers to 'evidence of collaboration' rather than concretely stating that authors are collaborating. In order to build better metrics, citations and a wider body of co-authorship data would be required, which is beyond the scope of this project.

This said, the table is still useful as a guide to who these staff members should focus future collaboration with. It *does not* assert that these authors are not already collaborating, only that they should treat the collaboration recommendation as a useful guide to who would be fruitful to work with, as their work is quite strongly related.

The top suggestion is for Dr. Archibald to work with Prof. Harris. They both work in the Centre of Atmospheric Science. There is one instance of collaboration on `http://www.ch.cam.ac.uk/publications/authors` but this article was not successfully collected to $\Delta 7$. If this evidence was represented in the dataset, it is likely their recommendation score would drop. It is so high is because their work was considered very similar by the models, pushing their recommended collaboration score high.

David Wales and Daan Frenkel were second highest, and there is also evidence to suggest they have collaborated in the past on `http://www.ch.cam.ac.uk/publications/authors`. Some of this evidence was represented in the collaboration matrix, but perhaps with more data, the association would have been stronger. Their research was considered very similar and this outweighed the evidence of collaboration to give a high score.

This trend (some evidence of collaboration, but strong similarity) is present in most of the top 20.

It should be noted that date of publications was not a factor considered in the analysis. This goes some way to explaining why authors who were not simultaneously at the department for long periods have high recommended collaboration scores, as they would not have collaborated. For example, this is probably the case for Dr. Andrew Bond, who features several times in the table. As a recent staff member who specialises in crystallographic techniques, there has not been a great deal of time for collaboration and co-authorship. Many authors publish articles mentioning crystallography, and so Dr. Bond's work will have high similarity to several members in the department.

8.6 Comments on Singletons

It was mentioned in §5.3.2 that singletons (articles with few to no connections to others or those that form their own community)in the graph were predicted to be significantly different to most articles. The singletons produced analysing $\Delta 7$ in §6.1 were examined to test this hypothesis.

It was certainly the case that some of the 33 singletons produced in the community finding process were poorly formatted or different. There was one article that detailed RSC award winners, one symposium handout, and 14 did not have well resolved abstracts. However the majority of singletons appeared to be mostly normal (if not quite narrow-scope) articles. The average number of words in a singleton document was lower than the average for $\Delta 7$ (65.7 words vs 125.7). It appears that in the majority of cases, singletons lie 'just outside' a threshold for inclusion, rather than being freak anomalies.

8.7 Data Acquisition Supplementary Information

8.7.1 Publisher Denial of Service

As mentioned in §2.4.3, Taylor & Francis and ACS banned the scraping computer's IP address during the second stage of global scraping. This section explores why this occurred.

Taylor & Francis banned the IP address after it detected over 100 requests were made within five minutes. This corresponds to a request every three seconds. This was a modest server load compared to other publishers, and was not foreseen to cause problems.

The ACS banning occurred because of a bug in the randomisation of requests. The program was instructed to take a DOI from a random publisher every time it made a request, rather than just a random DOI. Since the largest publisher was ACS, the program eventually exhausted DOIs from the other publishers, until there were only ACS DOIs to 'randomly' draw requests from. This meant the request frequency to the ACS server went up dramatically. This increase broke the threshold of allowed requests at the ACS server which then banned the IP (approximately 10 requests a second).

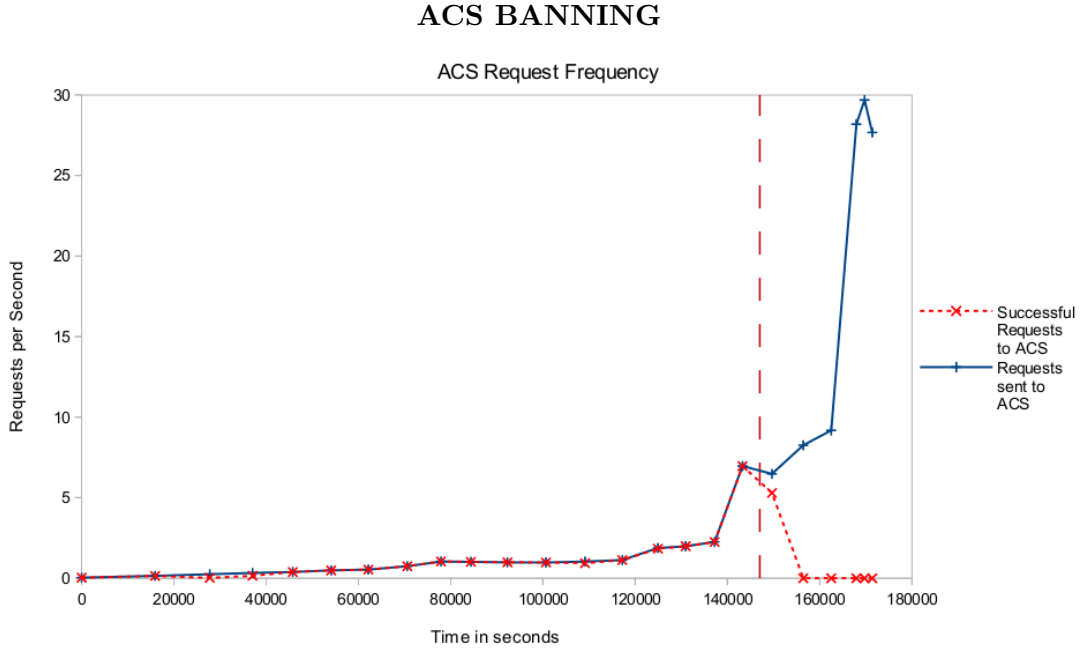


Figure 8.5: The request frequency is plotted in blue, the received pages frequency in red. The vertical dashed line shows where the server detected the scape and banned the IP.

The program was capable of making a total number of approximately 30 requests per

second. As can be seen in figure 8.5, the program began to run out of requests to other publishers after approximately 140,000 seconds, resulting in an increase in the proportion of total requests per second to ACS. The ban occurred after approximately 150,000 seconds, after which there were no more responses received.

8.7.2 Some Observations on $\Delta 1$ through $\Delta 6$

There is much to be learnt by examination and simple statistics of the collected data. This section details some of this analysis which was used in development of the scraping program and to inform upon algorithm and processing design choices.

When deciding how many XPaths were required, it was necessary to examine publication profiles. The publisher 'market share' can be approximated from examining $\Delta 3$.

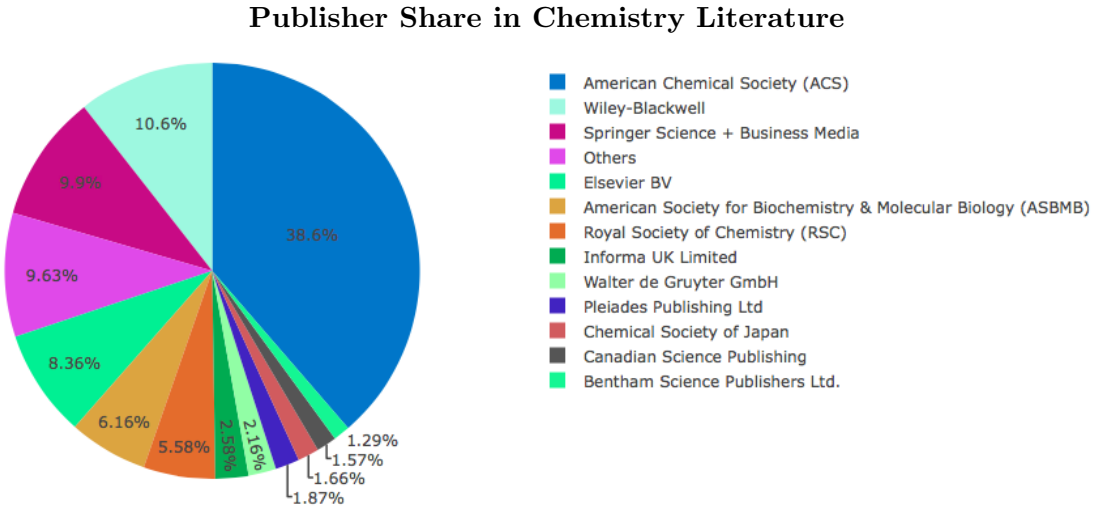


Figure 8.6: Articles grouped by publisher in $\Delta 3$. Only the top 12 publishers are shown.

As shown in shown in figure 8.6, it can be seen that 90% of all the chemistry literature collected was published by just 12 publishers, the majority from ACS, Wiley-Blackwell, Springer and Elsevier BV. Looking at the UK scraping DOI dataset (Figure 8.7), the same large publishers are represented, but the Royal Society of Chemistry has a much larger share. This is to be expected, as the RSC is a UK based body. In the UK, there is a more even distribution between the large publishers.

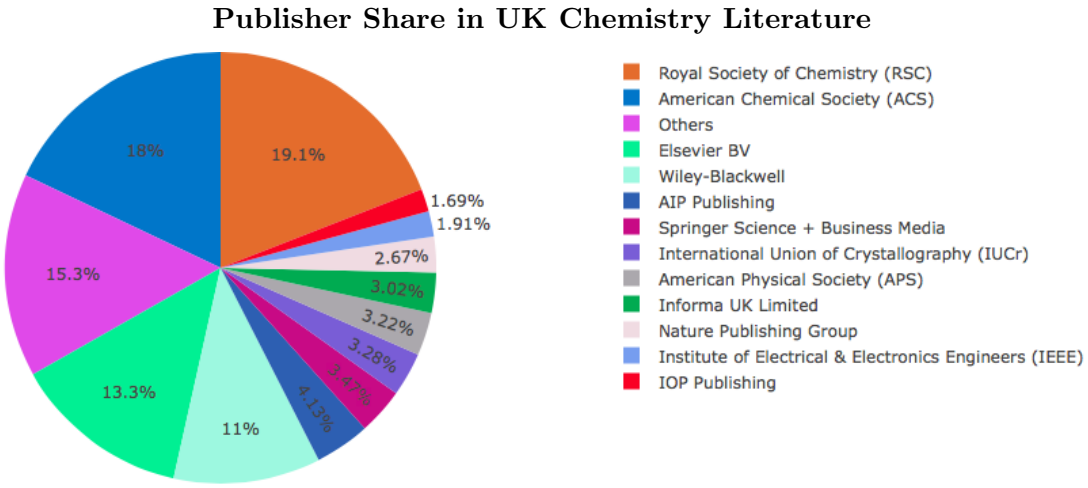


Figure 8.7: Articles grouped by publisher in $\Delta 1$. *Only the top 12 publishers are shown.*

The corpus of combined titles and abstracts in $\Delta 6$ was then examined. An understanding of word distributions would inform data sanitiation practices. It is is included here for interes and completeness. The word frequencies across all the data were found to be approximately Zipfian, with a gradient of -1.11[7] See figure 8.8

[7] A Zipfian distribution is a subset of the Pareto distribution, stating that the frequency of a word is proportional to its ranking in the word frequencies table. Ideally, the gradient of a log(frequency) vs log(rank) should be -1.0 [23]

Approximate Zipfian Distribution of Collected Corpus

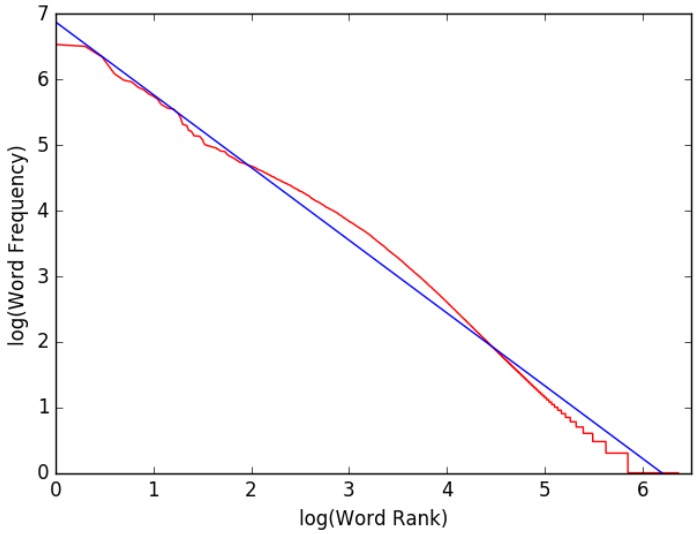


Figure 8.8: The log Frequency of words vs the log of their position in the rank in the word frequency table in blue. Best fit line in red, gradient = -1.11, intercept 6.3.

A summary of the corpus statistics are shown below: Note that the mean scores are

Table 8.2: Titles and Abstracts in Databases

	Δ4 (Global)	Δ2 (UK)
Total Word Count	61,296,410	2,256,722
Total Unique Words	2,326,725	60,166
Total Document Count	464,712	16,363
Mode Words per Title	11	11
Mean Words per Title	12.2	14.0
Mode Words per Abstract	156	52
Mean Words per Abstract	119.7	158.4
Mode Sentences per Abstract	4	4
Mean Sentences per Abstract	5.4	6.0

slightly higher for the UK, suggesting UK universities tend to publish slightly more verbose publications. The mode abstract length for the UK abstracts is significantly below the mean. This is indicative of a skewed, 'noisy distribution', which is indeed found when the distribution is plotted (figure 8.9).

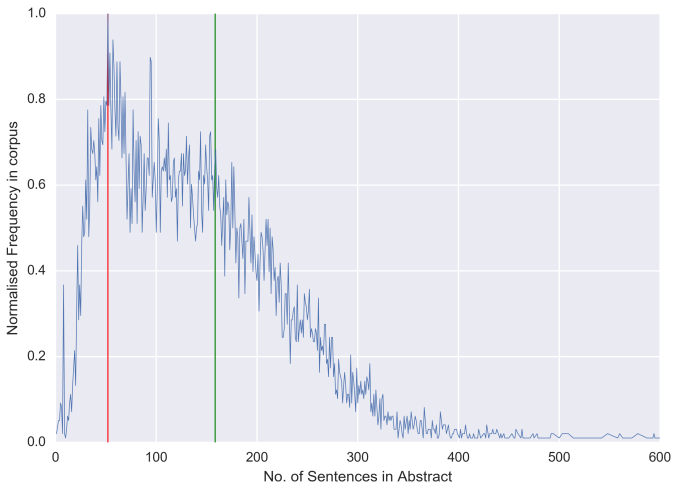


Figure 8.9: Skewed distribution for number of words in abstracts. The mode is marked in red, the mean in green.

As can be seen in the plot, there is significant variation in abstract lengths, with anything from 25 to 200 words commonly observed.

8.7.3 UK Departments

University Chemistry departments with suitable websites were considered when building the input list for the scraping program. Table 8.3 details all the departments that were included. A crawler program was written to navigate through these websites and store urls which had DOIs in them for the main program to scrape.

Table 8.3: UK Chemistry Departments considered in Scraping

Department	URL
Aberdeen	http://www.abdn.ac.uk/chemistry/
Aston	http://www.aston.ac.uk/eas/about-eas/academic-groups/ceac/
Bangor	http://www.bangor.ac.uk/chemistry/index.php
Bath	http://www.bath.ac.uk/chemistry/
Belfast (Queen's)	http://www.qub.ac.uk/schools/SchoolofChemistryandChemicalEngineering/
Birmingham	http://www.birmingham.ac.uk/schools/chemistry/index.aspx
Bradford	http://www.brad.ac.uk/acad/chemistry/
Brighton	http://about.brighton.ac.uk/pharmacy/
Bristol	http://www.bris.ac.uk/Depts/Chemistry/Bristol_Chemistry.html
Cambridge	http://www.ch.cam.ac.uk/
Cardiff	http://www.cardiff.ac.uk/chemistry
Dundee	http://www.lifesci.dundee.ac.uk
Durham	http://www.dur.ac.uk/chemistry/
Edinburgh	http://www.chem.ed.ac.uk/
Essex	http://www.essex.ac.uk/bs/
Glasgow	http://www.chem.gla.ac.uk/
Greenwich	http://www.gre.ac.uk/engsci/study/pharchemenv
Heriot-Watt	http://www.eps.hw.ac.uk/institutes/chemical-sciences.htm
Hertfordshire	http://www.herts.ac.uk/research/hhsri/research-areas-hhsri/pharmacy-and-pharmacology/pharmaceutical-chemistry
Huddersfield	http://www.hud.ac.uk/sas/chemistry/
Hull	http://www2.hull.ac.uk/science/chemistry.aspx
Keele	http://www.keele.ac.uk/chemistry/
Kent at Canterbury	http://www.kent.ac.uk/bio/
Kingston	http://sec.kingston.ac.uk/research/research-centres/
Lancaster	http://www.lancaster.ac.uk/chemistry/
Leeds	http://www.chem.leeds.ac.uk/
Leicester	http://www.le.ac.uk/chemistry/

Department	URL
Lincoln	https://www.lincoln.ac.uk/home/chemistry/
Liverpool	http://www.liv.ac.uk/chemistry/
Liverpool John Moores	https://www.ljmu.ac.uk/about-us/faculties/faculty-of-science/school-of-pharmacy-and-biomolecular-sciences
London Met.	http://www.londonmet.ac.uk/faculties/faculty-of-life-sciences-and-computing/school-of-human-sciences/
Loughborough	http://www.lboro.ac.uk/departments/chemistry
Manchester	http://www.manchester.ac.uk/chemistry/
Manchester Met.	http://www.sste.mmu.ac.uk
Newcastle	http://www.ncl.ac.uk/chemistry/
Northumbria	https://www.northumbria.ac.uk/about-us/academic-departments/applied-sciences/
Nottingham	http://www.nottingham.ac.uk/chemistry/
Nottingham Trent	http://www.ntu.ac.uk/sat/about/academic_teams/chemistry.html
Open University	http://www.open.ac.uk/science/chemistry/
Oxford	http://www.chem.ox.ac.uk/
Univ. West Scotland	http://www.uws.ac.uk/schools/school-of-science/departments/chemistry-and-chemical-engineering/
Plymouth	https://www.plymouth.ac.uk/schools/school-of-geography-earth-and-environmental-sciences/chemistry
Reading	http://www.reading.ac.uk/chemistry/
Robert Gordon	http://www.rgu.ac.uk/about/faculties-schools-and-departments/faculty-of-health-and-social-care/school-of-pharmacy-and-life-sciences1
St Andrews	http://ch-www.st-and.ac.uk/
Salford	http://www.salford.ac.uk/environment-life-sciences/research/biomedical
Sheffield	http://www.sheffield.ac.uk/chemistry
Sheffield Hallam	http://www.shu.ac.uk/schools/sci/chem/
South Wales	http://www.southwales.ac.uk/chemistry/
Southampton	http://www.soton.ac.uk/chemistry/
Strathclyde	http://www.strath.ac.uk/chemistry/
Sunderland	http://www.sunderland.ac.uk/ug/subjectareas/pharmacychemistrybiomedicalsciences/
Surrey	http://www.surrey.ac.uk/chemistry/
Sussex	http://www.sussex.ac.uk/chemistry/
Teesside	http://www.tees.ac.uk/schools/sst/
UEA	http://www.uea.ac.uk/chemistry
Warwick	http://www2.warwick.ac.uk/fac/sci/chemistry/
York	http://www.york.ac.uk/depts/chem/

Department	URL
Bradford Ploymer IRC	http://www.brad.ac.uk/acad/irc/
Cardiff Pharmacy	http://www.cardiff.ac.uk/pharmacy-pharmaceutical-sciences
Burbeck Chemistry	http://www.bbk.ac.uk/bcs/
Burbeck Crystallography	http://www.cryst.bbk.ac.uk/
Imperial College London	http://www.imperial.ac.uk/chemistry/
King's College London	http://www.kcl.ac.uk/nms/depts/chemistry/index.aspx
Queen Mary London	http://www.sbcs.qmul.ac.uk/
UCL School of Pharmacy	http://www.ucl.ac.uk/pharmacy
University College London	http://www.ucl.ac.uk/chemistry/
Sheffield Comput. Chem.	http://www.sheffield.ac.uk/is/research/groups/chemoinformatics

8.7.4 Publishers Considered in UK scraping

The UK scraping run found articles published by 36 different publishers. These are detailed below in table 8.4.

Table 8.4: All publishers found in the UK scraping run

IBM
Pleiades Publishing Ltd
Informa Healthcare
Informa UK Limited
Royal Society of Chemistry (RSC)
Vilnius Gediminas Technical University
Technical Association of Photopolymers, Japan
Springer US
Trans Tech Publications
Thieme Publishing Group
Nature Publishing Group
American Physical Society (APS)
IOP Publishing
Institute of Electrical & Electronics Engineers (IEEE)
American Chemical Society (ACS)
Walter de Gruyter GmbH
Pharmaceutical Society of Japan
American Association of Physics Teachers (AAPT)
AIP Publishing
Japan Society of Applied Physics
American Vacuum Society
Wiley-Blackwell
Springer Berlin Heidelberg
Springer New York
Royal Society of Chemistry
Public Library of Science (PLoS)
Surface Science Society Japan
Springer Science + Business Media
The Royal Society
Society of Rheology
Acoustical Society of America (ASA)
Springer International Publishing
Proceedings of the National Academy of Sciences
Japan Society for Analytical Chemistry
International Union of Crystallography (IUCr)
Chemical Society of Japan
EDP Sciences

8.8 Automatic XPath Generation

The initial approach was to analyse the HTML tree to automatically recognise useful tabulated or listed data. The program started at the tree's root and repeatedly followed the branch with the most 'repeating substructure'. The recursive algorithm is summarised below:

1. `Count # of descendents of each child node`
2. (a) `Calculate the pairwise similarities between all child nodes`
   (b) `Consider two nodes similar if pairwise similarity is above a heuristic threshold`
   (c) `Calculate proportion of nodes that are considered similar`
3. `If proportion calculated in (c) is above a heuristic threshold, this node represents a store of information, and the XPath has been found. Otherwise, move to child node with highest # of descendants, return to step (1)`

The heuristic thresholds are adjustable parameters. The approach was successful for webpages with large numbers of records, formatted in repeating fashion (such as long tables, lists etc.), but performed poorly for smaller or unstructured collections of data. As such it was not sufficiently flexible for the task of scraping large quantities of chemical data, and was not developed further.

Bibliography

[1] Brian S. Everitt. *Cambridge Dictionary of Statistics*. Cambridge University Press, 1998. ISBN 0521593468.

[2] Michael Waskom et al. seaborn: v0.7.0 (january 2016). 2016. doi: 10.5281/zenodo.45133. URL `http://dx.doi.org/10.5281/zenodo.45133`.

[3] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed 2016-03-30].

[4] Vincent D Blondel, Jean-Loup Guillaume, and Etienne Lefebvre Renaud Lambiotte. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Thory and Experiment*, 10:1000, 2008.

[5] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *IEEE Transactions on Network Science and Engineering*, 1:76–90, 2015.

[6] Crossref Foundation. *The Formation of CrossRef: A Short History*. 2009. URL `http://www.crossref.org/08downloads/CrossRef10Years.pdf`. [Online; accessed 2016-03-10].

[7] Norman Paskin. Digital object identifier (doi®) system. *Encyclopedia of Libary and Information Sciences*, pages 1586–1592, 2010.

[8] The doi handbook - international doi foundation (2016). 2014. URL `http://www.doi.org/doi_handbook/7_IDF.html#7.2.1`. [online; Accessed 2016-03-25].

[9] Chris D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, November 1990. ISSN 0163-5840. doi: 10.1145/101306.101310. URL `http://doi.acm.org/10.1145/101306.101310`.

[10] "machine learning". encyclopdia britannica. encyclopdia britannica online. *Encyclopdia Britannica Inc.*, 2016. URL `http://www.britannica.com/technology/machine-learning`. [Online; Accessed 2016-03-25].

[11] Karl Pearson. LIII. on lines and planes of closest fit to systems of points in

space. *Philosophical Magazine Series 6*, 2(11):559–572, nov 1901. doi: 10.1080/14786440109462720. URL http://dx.doi.org/10.1080/14786440109462720.

[12] M. F. Porter. *An Algorithm for Suffix Stripping*, volume 14. 1980.

[13] M. F. Porter. The Porter2 stemming algorithm, 2002.

[14] GE Hinton LJP van der Maaten. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[15] Sokal R and Michener C. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[16] The Unicode Consortium. *Unicode® 6.0.0*. 2010. ISBN 978-1-936213-01-6. URL http://www.unicode.org/versions/Unicode6.0.0/. [online; Accessed 2016-03-25].

[17] Mark Davis. Unicode nearing 50% of the web. *Official Google Blog*, 2010. [online; Accessed 2016-03-25].

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. URL http://arxiv.org/abs/1310.4546.

[20] Ingo Feinerer and Kurt Hornik. *wordnet: WordNet Interface*, 2016. URL https://CRAN.R-project.org/package=wordnet. R package version 0.1-11.

[21] Mike Wallace. *Jawbone Java WordNet API*, 2007. URL http://mfwallace.googlepages.com/jawbone.

[22] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[23] A. Ullah and D.E.A. Giles. *Handbook of Empirical Economics and Finance*. Statistics: A Series of Textbooks and Monographs. CRC Press, 2010. ISBN 9781420070361. URL https://books.google.co.uk/books?id=QAUv9R6bJzwC.

[24] The doi handbook - numbering (2014). 2016. URL https://www.doi.org/doi_handbook/2_Numbering.html#2.2.2. [online; Accessed 2016-03-25].

[25] The copyright and rights in performances (research, education, libraries and archives) regulations, no. 1372 regulation 3. 2014. URL http://www.legislation.gov.uk/uksi/2014/1372/regulation/3/made. [Online;accessed 2016-03-22].

[26] Responsible content mining. 2015. URL http://contentmine.org/wp-content/uploads/2015/06/responsible-content-mining-1.pdf. [Online; accessed 2016-03-22].

[27] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013. URL http://research.microsoft.com/apps/pubs/default.aspx?id=189726.

[28] David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X.

[29] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[30] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014. URL http://arxiv.org/abs/1405.4053.

[31] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009. ISBN 0596516495, 9780596516499.

[32] Surasak Seesukphronrarak and Toshikazu Takata. Novel fluorene-based biphenolic monomer: 9, 9-bis(4-hydroxyphenyl)-9-silafluorene. *Chem. Lett.*, 36(9):1138–1139, 2007. doi: 10.1246/cl.2007.1138. URL http://dx.doi.org/10.1246/cl.2007.1138.

[33] Yu. B. Tsaplev. Photochemical transformations of anthraquinone in polymeric alcohols. *Russian Journal of Physical Chemistry A*, 86(12):1909–1914, oct 2012. doi: 10.1134/s0036024412120266. URL http://dx.doi.org/10.1134/s0036024412120266.

[34] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014. URL http://jmlr.org/papers/v15/vandermaaten14a.html.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[36] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. 2009. URL http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

[37] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific

computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL http://ipython.org.

[38] John D. Hunter. Matplotlib: A 2d graphics environment, 2007.

[39] S. Chris Colbert Stfan van der Walt and Gal Varoquaux. The numpy array: A structure for efficient numerical computation, 2011.

[40] Wes McKinney. Data structures of statistical computing in python, 2010.

[41] Jiro Tsuji. *Palladium Reagents and Catalysts: New Perspectives for the 21st Century*. Wiley, 2004. ISBN 0470850329.

[42] *Ullmann's Encyclopedia of Industrial Chemistry*. Wiley-Blackwell, jun 2000. doi: 10.1002/14356007. URL http://dx.doi.org/10.1002/14356007.

[43] Hervé Bricout, Jean-François Carpentier, and André Mortreux. Nickel vs. palladium catalysts for coupling reactions of allyl alcohol with soft nucleophiles: activities and deactivation processes. *Journal of Molecular Catalysis A: Chemical*, 136(3):243–251, dec 1998. doi: 10.1016/s1381-1169(98)00067-3. URL http://dx.doi.org/10.1016/S1381-1169(98)00067-3.

[44] M. Bonarowska, O. Machynskyy, D. Łomot, E. Kemnitz, and Z. Karpiński. Supported palladium–copper catalysts: Preparation and catalytic behavior in hydrogen-related reactions. *Catalysis Today*, 235:144–151, oct 2014. doi: 10.1016/j.cattod.2014.01.029. URL http://dx.doi.org/10.1016/j.cattod.2014.01.029.