

Part III Project
Automatic Chemical Analysis - Big Data and Machine
Learning in 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 metadata 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 finding relationships between researchers at the University of Cambridge Chemistry Department.

Contents

List of Figures	3
Glossary	5
1 Introduction	8
1.1 Modern Scientific Publishing	8
1.2 Motivation	8
1.3 Aims	9
2 Data Acquisition	10
2.1 Background	10
2.1.1 HTML and Xpath	10
2.2 Automatic Xpath Generation	11
2.3 Collection Strategy	11
2.3.1 DOIS : Document Object Identifiers	12
2.3.2 Scraping Program	13
2.4 Collection Results	15
2.4.1 UK University Department scraping	15
2.4.2 Global Scale Scraping	16
2.4.3 Problems with ACS and Taylor & Francis	17
2.4.4 Analysis of Collected data	18
Observations	20
3 Techniques for Language Processing	24
3.1 Background	24
3.2 Bag of Words	24
3.3 Word2Vec	25
3.4 Doc2Vec	27
4 Algorithm Development	28
4.1 Premise	28
4.2 Data Sanitisation	28
4.3 Word2Vec Models	31

4.3.1	TF-IDF	32
4.3.2	Aggregations	32
4.4	Doc2Vec Models	33
5	Model Examination	35
5.1	Word Similarities	35
5.2	Document Similarities	37
5.3	Visualisation Techniques	38
5.3.1	Network Visualisation	38
5.3.2	Networks and Network Visualisation	39
6	Analysis with Sample Dataset	42
6.1	Cambridge Chemistry research clusters	42
6.2	Cambridge Staff Member Similarities	46
6.3	Combining research clusters and authors	51
7	Conclusions	61
8	Recommendations for Further Work	63
8.1	Greater Dimensionality and Training Improvements	63
8.2	Greater use of word vectors	63
8.3	Time resolution in clustering	64
8.4	Open Source Chemistry Vectors	64
8.5	Structure stemming	64
8.6	Multiply labelled Documents	64
Bibliography		65
9	Appendix	66
9.1	UK Departments scraped	67
9.2	Publishers Considered in UK scraping	70
9.3	Scaled Communities for staff members in Cambridge	71

List of Figures

2.1	Tree representation of HTML Code	10
2.2	Anatomy of a DOI	12
2.3	Pattern Matching Procedure for DOIs	13
2.4	Data Flow in Scraping Procedure	14
2.5	Efficiency of UK Department Scraping	16
2.6	Request Frequency Leading to ACS Ban	18
2.7	Efficiency of Large Scale Scraping	19
2.8	Summary of Database Preparation	20
2.9	Publisher Share in Chemistry Literature	21
2.10	Publisher Share in UK Chemistry Literature	22
2.11	Zipfian Plot of Collected Corpus	23
3.1	Word2Vec Training Architectures	26
3.2	Word Vector Relationships	27
4.1	Punctuation removed in sanitation processing	29
4.2	Preprocessing Pipeline	31
5.1	PCA Dimensional Reduction	39
5.2	TSNE Dimensional Reduction	39
5.3	Network Visualisation of 10,000 document random sample	41
6.1	Network Visualisation of University of Cambridge Chemistry Department Documents	43
6.2	Recursion Tree for Document Community Generation	44
6.3	Cambrdige Author Similarity Clustermap	47
6.4	Cambridge Author Similarity Dendrogram	49
6.5	Dendrogram annotated with labelled fields	50
6.6	Author community Spread	52
6.7	Author Coincidence Matrix Heatmap	54
6.8	Author Collaboration Heatmap	56
6.9	Community-summed Author Collaboration Heatmap	57
6.10	Recommended Collaboration Matrix	58

6.11 Recommended Collaboration Scores for Particular Staff Member	60
9.1 Number of research communities authors are associated with. High values suggest more interdisciplinary work. The bars have been scaled relative to the author's total publication count.	71

Glossary

ACS blah.

API blah.

Bag Of Words blah.

BagOfCitations blah.

CBOW blah.

CCD blah.

Cluster Map blah.

Communities blah.

Corpus blah.

Cosine Similarity blah.

Crawling blah.

Crossref blah.

Dendrogram blah.

Doc2Vec blah.

DOI blah.

EuclideanSimilarity blah.

Gensim blah.

Gephi blah.

Hadamard Division blah.

Hadamard Square Root blah.

HTML blah.

Hyperparameters blah.

IDF blah.

IP Address blah.

Lancaster blah.

Machine Learning blah.

Meta-data blah.

MongoDB blah.

Neural Net blah.

NLP blah.

Paragraph Vectors blah.

PCA blah.

Perl blah.

PILA blah.

Porter blah.

Python blah.

Regex blah.

RSC blah.

SciFinder® blah.

Scraping blah.

Skipgram blah.

Snowball blah.

Stemmer blah.

Stop Words blah.

Taylor & Francis blah.

TF-IDF blah.

Training Dataset blah.

Training Epoch blah.

TSNE blah.

Unicode blah.

UPGMA blah.

UTF-8 blah.

Web Of Science TM blah.

Word2Vec blah.

WordNet blah.

XML blah.

XPath blah.

Zipf's Law blah.

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, learned 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 metadata. 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 metadata on papers found on the internet, a large, representative dataset of chemical academic writing language can be built up. Machine Learning techniques can then 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 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 for future analyses to be performed with

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 selected 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, which is covered in §5. Finally, when the models were shown to be performing successfully, they were used in an analytical setting; To examine the relationships between authors and research communities in the University of Cambridge Chemistry Department and eventually to recommend specific collaborations between staff that were predicted to be fruitful.

2. Data Acquisition

2.1 Background

2.1.1 HTML and Xpath

Internet webpages are written in a markup language called HTML, (HyperText Markup Language). 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 program written to automatically interpret webpages to extract information, is known as a ‘scraping’ program. The 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 in a browser as a table with 3 rows, each row containing ‘Table Data A/B/C’. The method of tree traversal is by specifying a path through the document tree on the right, using an ‘xpath’.

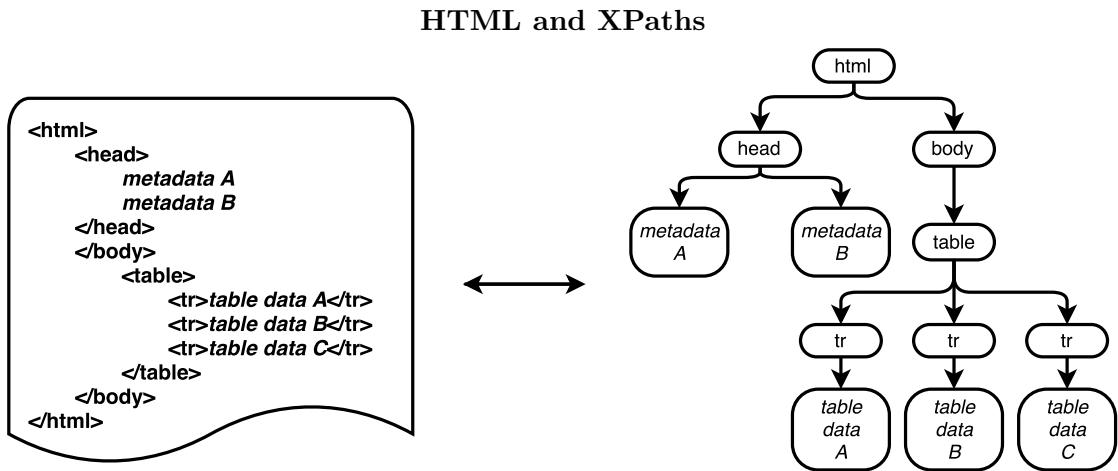


Figure 2.1: Tree representation of HTML code. The html code here displays a table with 3 rows. The page has two pieces of metadata associated with it, stored in the ‘head’.

Xpaths are just paths through the document tree to the desired information. In order to ‘scrape’ the data in the table, the following xpath could be used:

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

This presents an immediate problem, as scraping millions of webpages requires millions of potentially different xpaths to be known. 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 where useful tabulated or listed data was. The program started at the root of the tree and repeatedly followed the branch with the most ‘repeating substructure’. The recursive algorithm is summarised below:

1. Count # of descendants 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 flexible enough for the task of scraping large for chemical data, and was not implemented in final solution.

2.3 Collection Strategy

As generating xpaths proved unsuitable, a new strategy was required. Chemical information is usually disseminated as journal articles, mostly 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)

2.3.1 DOIS : Document Object Identifiers

DOIs are computer-friendly labels for articles. DOIs are issued by a number of accredited bodies, with the vast majority of chemistry related articles issued by Crossref.¹ [?]. By pre-pending a DOI string with the url stub <http://dx.doi.org/>, the International DOI foundation (IDF) service redirects the request to the publisher's website to display the article the DOI refers to. The structure of a DOI is shown in Figure 2.2.

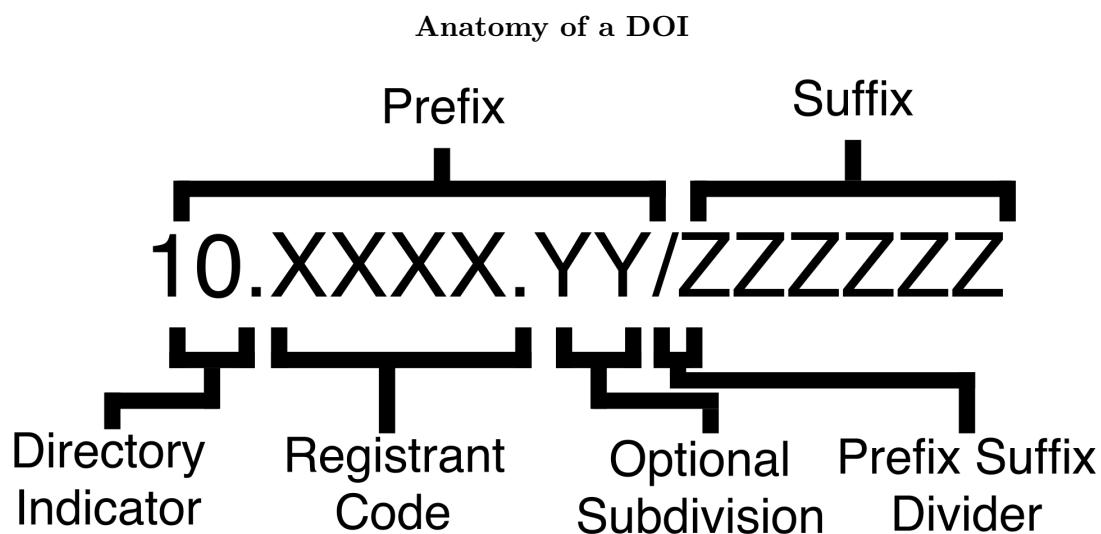


Figure 2.2: Doi structure. The structure consists of a numeric prefix (X and Y must be integers) and alphanumeric suffix (Z can be any Unicode encoded character)

DOIs consist of a prefix and suffix. The prefix is subdivided into the Directory Indicator (always integer 10) separated from the Registrant Code, assigned by the issuing body. Registrant codes are numeric and can be a minimum of 3 integers, with further optional subdivisions separated by full stops. The suffix is provided by the registrant themselves. It can take any form as long as it is unicode encoded.

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

¹Crossref is a not-for-profit body comprised from Publishers International Linking Association (PILA), an association of many academic publishers

Pattern Matching Procedure for DOIs

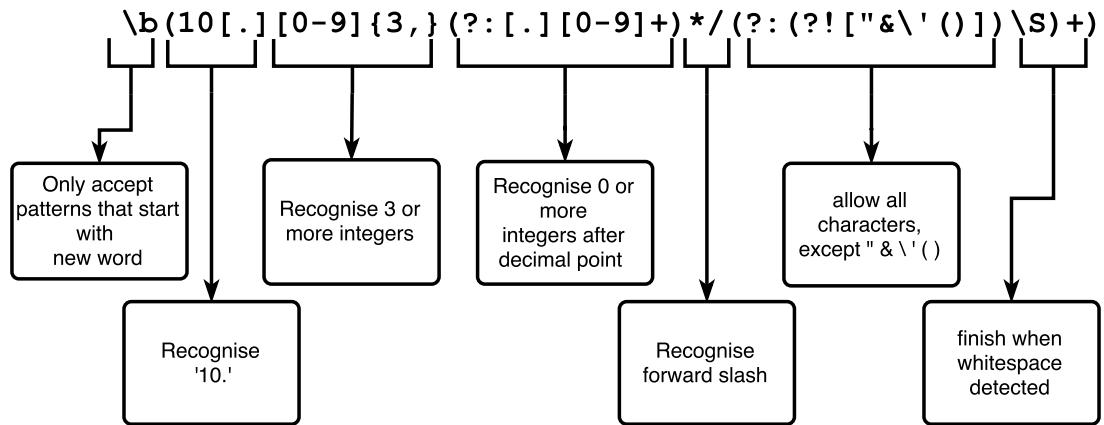


Figure 2.3: Perl Syntax Regex Code that can identify the vast majority of DOIs within free text)

Despite this, the regex was able to identify 90.4% of the dois on the Cambridge University Chemistry Department website <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 a large set of websites. Some meta-data associated with a DOI can be accessed using an online API exposed by Crossref. Further metadata can be accessed by following the <http://dx.doi.org/{DOI}> redirecting service by DOI[®].org. to visit publishers' websites to collect remaining metadata.

With this methodology in place, a scraping program was written in python to collect DOIs from a list of webpages and collect metadata in a 2 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.² The procedure is summarised in figure 2.4 below:

²Since there are comparatively few publisher websites, only 26 publisher xpaths were required for decent capture coverage.

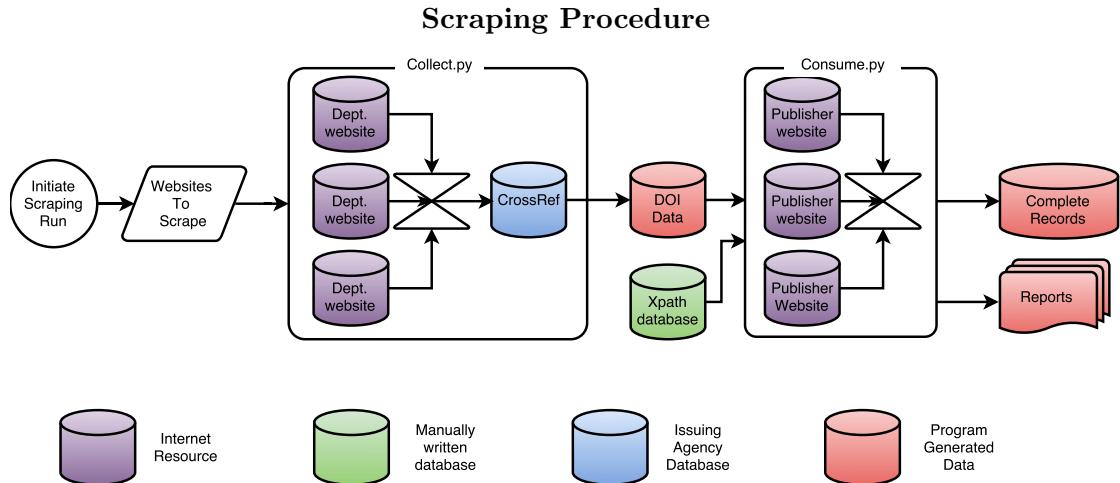


Figure 2.4: The data flow of the scraping program. Websites from an inputted list of websites are visited and dois are extracted in the process described in §2.3.1. The Crossref API service is then used to verify the extracted dois, and collects available meta-data. The program then accesses publisher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

The programmatic steps depicted in 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 the doi using `http://dx.doi.org/{doi}`
6. use xpath to collect article abstracts.

The program exports complete records as .json files, but also feeds directly to a MongoDB database. Once the program was written, the next priority was to obtain a list of webpages to scrape. This is described in §2.4.1 and §2.4.2

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 some urls were changed to give a list of 68 departments³. The program was run with this list as an input, the results of which are detailed in table 2.1. Conversion losses were due to 4 components. 45 losses

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%

for non-existent dois, 2753 losses to request errors (404 : not-found errors or permission problems), 133 to the program throwing internal errors and 3148 conversions were lost due missing publication xpaths. The 26 specified xpaths ⁴ were sufficient to convert 83.8% of successful requests. This was deemed acceptable, as most major publishers had been covered ⁵, and the missing publishers each covered a small number of articles it would take another 11 xpaths of the missing most popular publishers to increase the conversion rate from 83.8% to 90%. The efficiency is depicted in figure 2.5

³Details can be found in the appendix

⁴Corresponding to 37 publishers

⁵see appendix for list of covered publishers

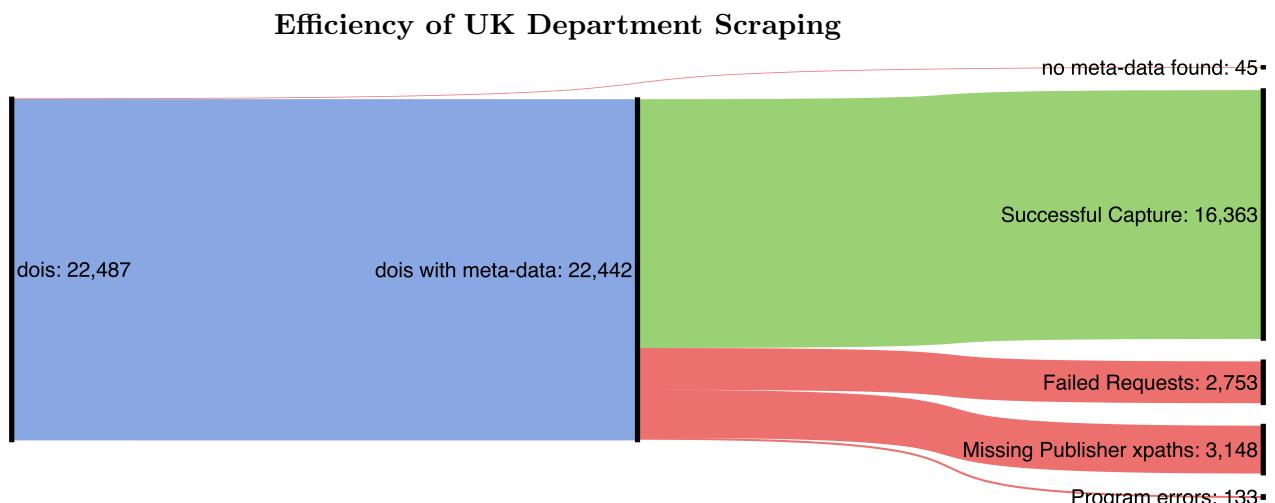


Figure 2.5: The loss processes are coloured red, successfully captured full records in green, and the maximum possible yield in blue.

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 also hosts the majority of its information under its own domain name, whereas other departments’ data are hosted on central university domains. The scraping program was confined to scrape only webpages belonging directly to chemistry department domains, not the university website as a whole. As a result, it is worth bearing in mind that the Cambridge chemistry department may be overrepresented in the UK chemistry data set.

2.4.2 Global Scale Scraping

The UK scrape was a respectable start, but much more data would be required to train a machine learning model successfully. One approach would have been to expand the scrape to world-wide chemistry departments, and other learned bodies. However, Crossref also exposes a search service that can be used to query its vast internal database. The program was then 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 results pages of these queries. Because the scraping job was so large, the program was set up to ‘pause’ before the publisher abstract collection stage. The results of 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, which was deemed successful, and would provide enough data to train a powerful machine learning

algorithm.

The records were then inspected and publisher distributions were considered. Some of this analysis is presented in §2.4.4. After careful considerations of request server loads and predicting capture probabilities, the second half of the scraping routine was set off to run for 3 days.

2.4.3 Problems with ACS and Taylor & Francis

Some publishers automatically track request volumes sent to their site as they wish to discourage automatic scraping of their data. Scraping their websites constitutes fair use and is legal in the UK , and despite the university owning a Licence to full access of these publishers' publications, the material collected was freely available without licence[?] [?]. However, during the scraping run, a bug in the randomisation of request frequencies resulted in detection by the ACS⁶ and Taylor & Francis. Both publishers responded by banning the IP address of the computer running the program. The department librarians were able to restore access, and it was agreed that no further large scale scraping runs would be performed.

Taylor & Francis banned the IP address after it detected over 100 requests were made within 5 minutes. This corresponds to a request every 3 seconds. This is 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 uptick broke the threshold of allowed requests at the ACS server which then banned the IP (approximately 10 requests a second).

⁶American Chemical Society

ACS BANNING

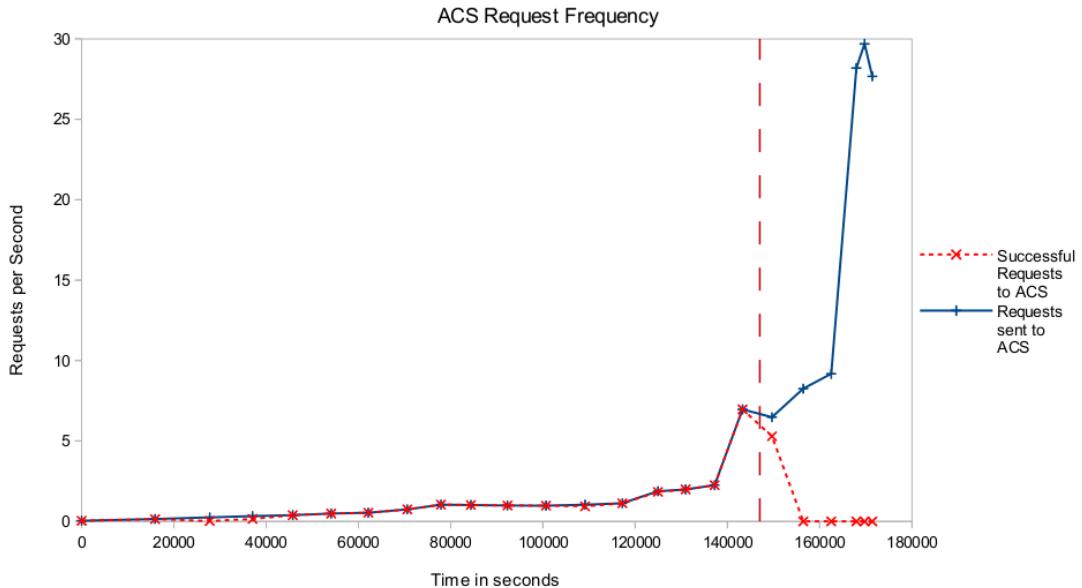


Figure 2.6: The request frequency is plotted in blue, the received pages frequency in red. The vertical dashed line shows where the server detected the scape and banned the IP.

The program was capable of making a total number of approximately 30 requests per seconds. As canbe seen in figure 2.6, 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.

2.4.4 Analysis of Collected data

The yield of the global scale scraping run was cut significantly by the ACS banning event. A summary is tabulated in 2.2, and shown graphically in figure 2.7.

Table 2.2: Large Scale 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%

Efficiency of Large Scale Scraping

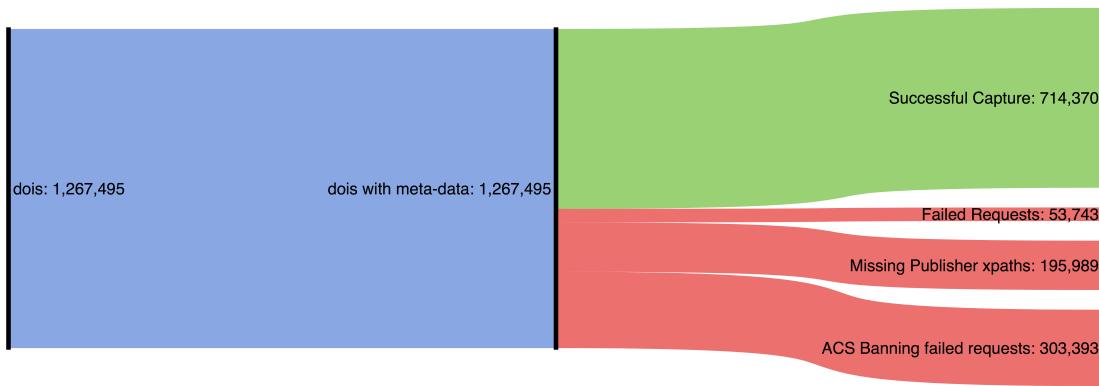


Figure 2.7: The loss processes are coloured red, successfully captured full records in green, and the maximum possible yield in blue.

The overall efficiency of the process is 56.4%, but excluding the acs records lost in the ban, the program's efficiency jumps to 74.0%, similar to the efficiency of the UK scraping run (§2.4.1). ⁷

The successfully captured 714,370 records were inspected and merged with the UK results. Records were rejected with short titles or abstracts (likely to be addenda, informal articles, retractions etc.) Records were also removed if the majority of the title and abstract were not written in ascii characters ⁸ (removing majority Japanese and Chinese script). This was done to provide better quality data for training the algorithm described in §4. This filtering resulted in a final training database of 464712 articles. This dataset is henceforth referred to as *the training dataset*. The entire database formation process is shown in figure 2.8.

⁷Also note that 100% of dois were converted to dois with metadata. This is because the format of the webpages scraped was consistent for every doi collected.

⁸ascii is an encoding for English characters a-z, A-Z, some punctuation and 1-9.

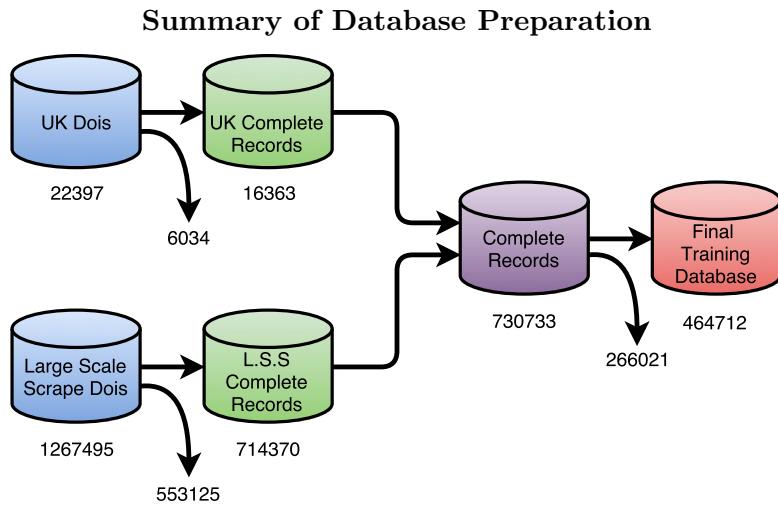


Figure 2.8: Blue databases represent data with dois and metadata. Green databases represent meta-data, dois and abstracts. The purple database is the combined complete records, and the red database is the data deemed suitable for the training algorithm. database sizes and losses are annotated.

It was instructive to examine these databases and derive some simple statistical results. The following section briefly explores some of these.

Observations

The publisher ‘market share’ can be approximated from examining the database⁹.

⁹Data was taken from ‘Large Scale Scrape Dois’ database, shown in figure 2.8

Publisher Share in Chemistry Literature

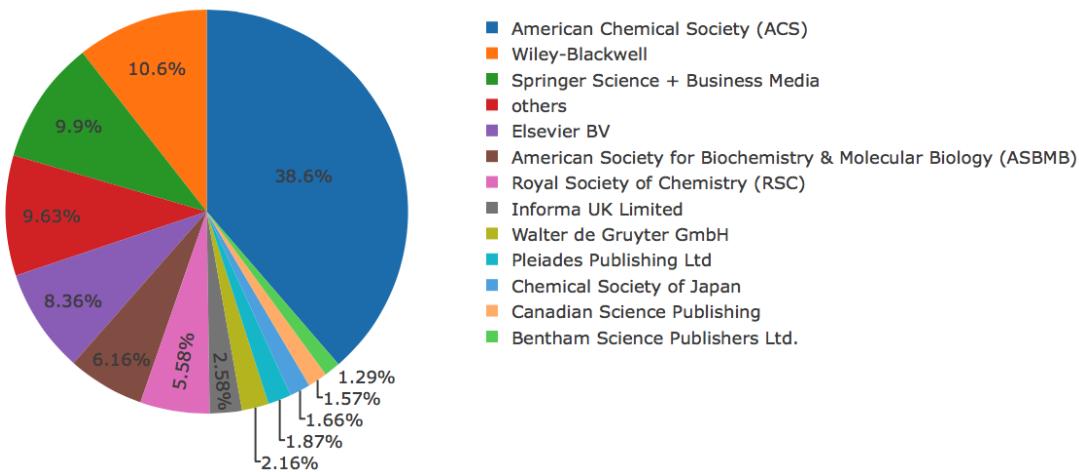


Figure 2.9: Articles grouped by publisher in the Large Scale Scrape doi database (marked 'Large Scale Scrape Dois' in figure 2.8). Only the top 12 publishers are shown.

As shown in figure 2.9, 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 2.10), 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.

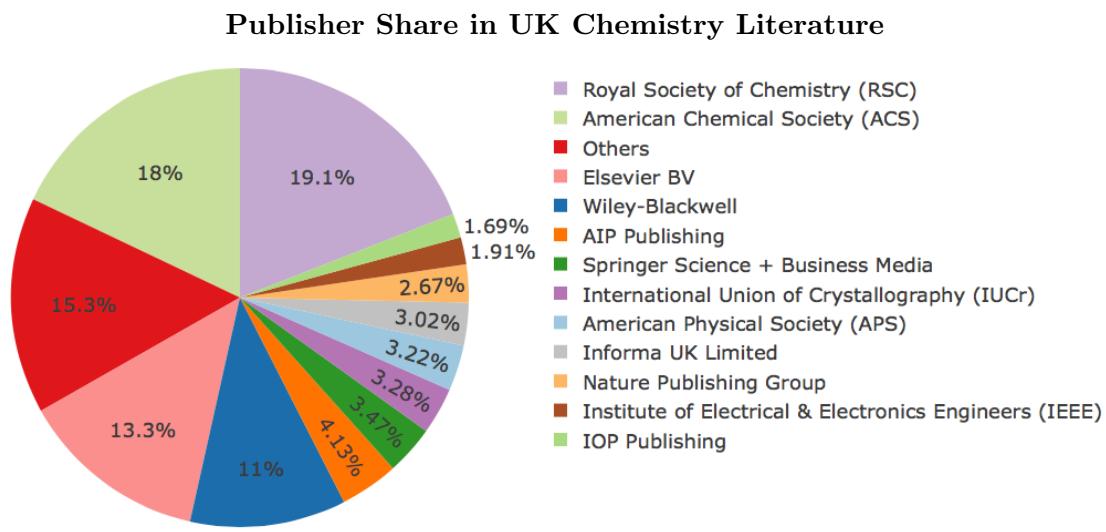


Figure 2.10: Articles grouped by publisher in the UK Doi database published by each publisher. Only the top 12 publishers are shown.

The corpus of combined titles and abstracts of the complete training database was then examined. The word frequencies across all the data were found to be approximately Zipfian, with a gradient of -1.11^{10} See figure 2.11

¹⁰A 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 $\log(\text{frequency})$ vs $\log(\text{rank})$ should be -1.0 [?]

Approximate Zipfian Distribution of Collected Corpus

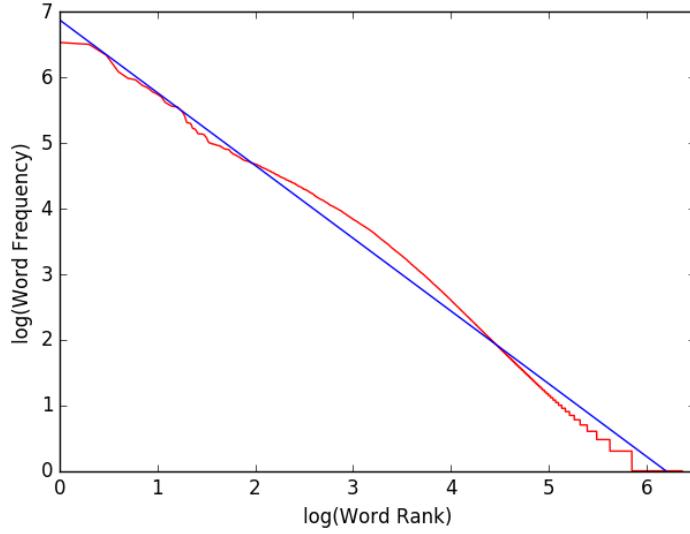


Figure 2.11: The log Frequency of words vs the log of their position in the rank in the word frequency table in blue. Best fit line in red, gradient = -1.11, intercept 6.3.

A summary of the corpus statistics are shown below:

Table 2.3: Titles and Abstracts in Training Database

Total Word Count	61,296,410
Total Unique Words	2,326,725
Total Document Count	464,712
Mode Words per Title	11
Mean Words per Title	12.2
Mode Words per Abstract	156
Mean Words per Abstract	119.7
Mode Sentences per Abstract	4
Mean Sentences per Abstract	5.4

3. Techniques for Language Processing

3.1 Background

Natural Language Processing (henceforth NLP) is the application of computer science to study, model and understand human languages using computers. Machine learning, a class of algorithms for fitting and predicting patterns in data, is a powerful technique, with 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

Table 3.1: Bag of words

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

Table 3.2 shows the vector representations for Documents A and B. The higher the scalar product of normalised $\mathbf{v}_A \cdot \mathbf{v}_B$, the more similar the documents are predicted to be. This model is used extensively in industry. A related model, the *bag of citations* model sets vector components are the presence or absence of citations. Both models are widely used in industry for analysing the publishing landscape.

3.3 Word2Vec

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

A recent successful approach has been the Word2Vec algorithm [?] [?]. The Word2Vec algorithm uses a neural net to represent words as vectors in a continuous rather than discrete space. Vectors for words with similar meanings will point in similar directions in this ‘Semantic space’. The Word2Vec algorithm is fed a language corpus sentence by sentence. The words within the sentences have a semantic relationship, which the algorithm uses to infer word meanings.

This is achieved with two architectures, Continuous Bag of Words (henceforth CBOW) and skip-gram. The CBOW architecture uses a shallow neural net to predict a word’s vector by summing or averaging the vectors of the surrounding words in a training sentence. The skip-gram algorithm predicts the vectors of words surrounding the current training word. By training with many input sentences, prediction vectors are gradually improved.

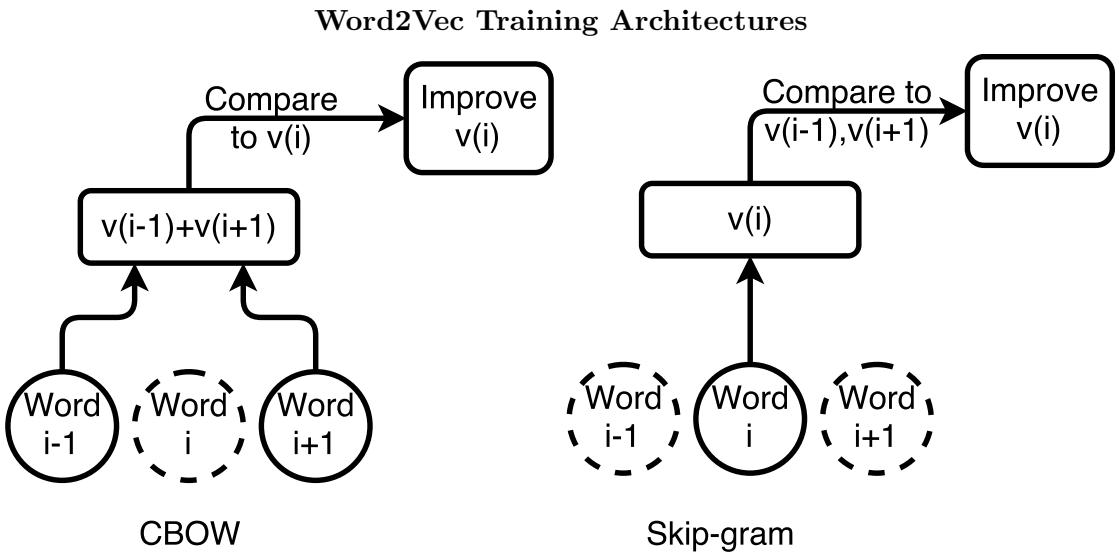


Figure 3.1: The training architectures of the Word2Vec training algorithm. Word vectors are denoted $v(i)$ for word i . In CBOW word i is predicted by the vector found by summing vectors surrounding i , and $v(i)$ is adjusted to be closer to this prediction. In skip-gram, word i 's vector is pairwise 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 words surrounding the current training word. The order of words within the window does not matter, but because the window ‘slides’ along as the algorithm considers words $i+1$, $i+2\dots$ word ordering is represented in the model . In skip-gram, a random number of nearby words are used for the prediction vectors for word i .

The model has added sophistication built in to reduce the importance of very 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.

Word Vector Relationships

$$\text{King} - \text{Man} + \text{Woman} \approx \text{Queen}$$

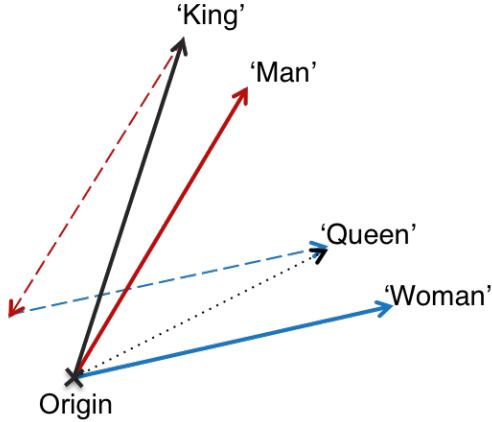


Figure 3.2: Schematic Representation of how concepts can be represented in word vector space. Word2Vec is able to replicate this behaviour. The vector found by $\text{vec}(\text{King}) - \text{vec}(\text{Man}) + \text{vec}(\text{Woman})$ is approximately equal to $\text{vec}(\text{Queen})$. The model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

Word2Vec models are able to 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.¹

3.4 Doc2Vec

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

The nature of the collected metadata detailed in §2 lends itself naturally to the Word2Vec and Doc2Vec algorithms, as it is a large store of natural language, rich with encoded information. The focus of the machine learning analysis phase of the project was directed at applying the Word2Vec and Doc2Vec algorithms to the dataset to try and automatically learn and classify chemical semantic concepts. These investigations are detailed in the following sections.

¹It is interesting to consider if chemical examples could be developed. This is considered further in §8

4. Algorithm Development

4.1 Premise

The aim of the machine learning phase was to apply the Word2Vec and Doc2Vec algorithms to the training dataset 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.

4.2 Data Sanitisation

The documents (titles and abstracts) in the training dataset 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 whitespace, 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						
"	+	?	-	—	-	-
#	,	@	-	'	Ξ	
\$.	[-	'	→	
%	/]	★	',	→	
&	-	^	▪	•	✖	
\	:	_	”	†	⇒	
'	;	`	≈	◦		
(<	{	≠	“		
)	=		~	①		
*	>	}	-	-		

Figure 4.1: All the punctuation removed in sanitation. Only these were found in appreciable quantities in the training dataset.

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 removed¹ (stopwords were taken from the Porter stopwords corpus[?] [?]). From inspection of the zipfian frequency table, (§2.4.4), it was apparent that chemistry literature also generates stopwords. Table 4.1 details ‘Chemistry’ stopwords that were identified and removed².

Finally, the processed words were sent through a ‘stemming algorithm’³. Several stemming algorithms were assessed (Porter [?], Snowball [?][?], Lancaster [?] and the Wordnet Lemmatizer [?][?][?]). The Snowball⁴ 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

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

²The 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

³A 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

⁴Also 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	Overaggressive 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:

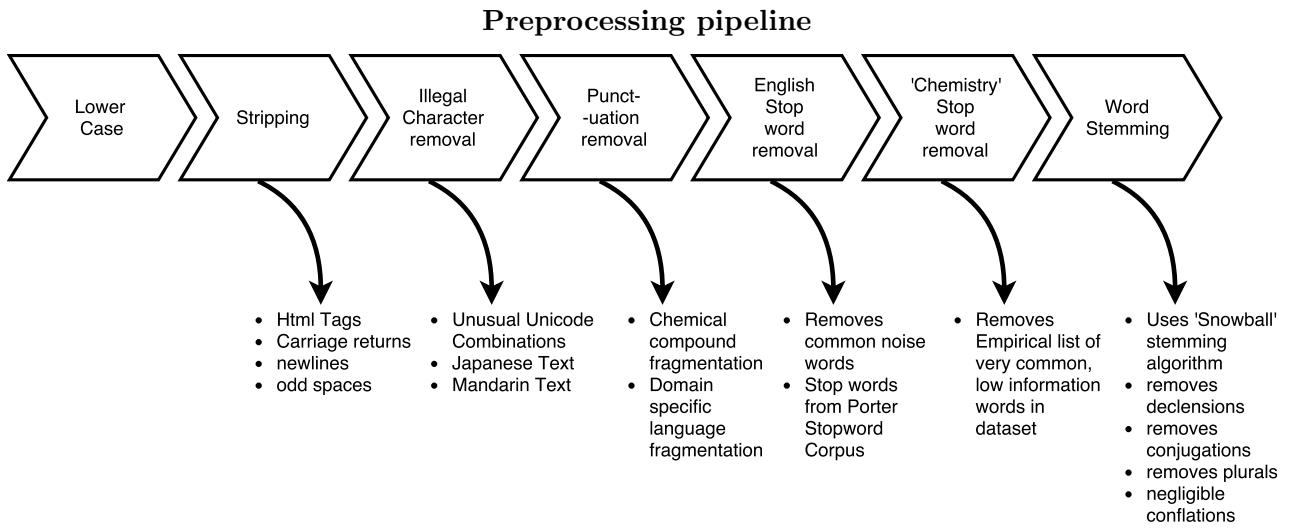


Figure 4.2: All documents in the training database were preprocessed with this pipeline schema before being used in training models

The process is best illustrated by real example from the dataset:

< 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 >.
[?] 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 python implementation [?]. The hyperparameters used for training were consistent for the two models. Training was carried out on all documents in the training dataset. The model was trained with sentences formed by simple splitting of documents

using full stops⁵. After examination of different hyperparameters, the models were run using hyperparameters representing good balance of specificity, speed and generality. The hyper parameters used are detailed in table 4.3. In order to represent documents

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)

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-IDF⁶ 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 [?]. 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 simply averaging word vectors directly into documents. 8 models for document vectors composed of Word2Vec models were constructed, detailed in table 4.4

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

⁶Term - 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* architecture⁷ using the gensim framework in python [?] using the same training data 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 specificity⁸, and also so that dimensions were consistent across all models. The Doc2Vec model was trained for 24 epochs, with hyperparameters detailed in table 4.4

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

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

⁸as well as for computational considerations of analytical techniques, see §5,§??

more work required per document. Negative sampling was disabled as per recommendations in the literature. [?] [?]. 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 difficult to assess model quality, due to a lack of concrete metrics to compare¹. The Word2Vec development team tested models against $\sim 10,000$ semantic and syntactic relationships (See Figure 3.2)[?] [?] [?]. 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 obtained by direct comparison of their word vectors. For words α and β , with vectors ν^α and ν^β . A possible metric is to compute euclidean distance between ν^α and ν^β ,

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

where D is the dimensionality ($D=100$). This metric 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 the *cosine similarity*, a measure of the directionality of ν^α and ν^β . A value close to 1 corresponds to high similarity of α and β . Cosine similarity is computed as:

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

¹This is to say, there is no objective ‘similarity’ relationship value between two words.

The CBOW and Skip-gram models were examined using these metrics. For a given word, they were requested to return the 3 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	Polystyrylbipyrnidin

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 (e.g. the models understood that catalysts and nanocatalysts are closely connected concepts).

It was observed that the skip-gram model gave misleading positives more frequently.³. The CBOW model was considered to be superior for word-word comparisons. It was

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

³in the catalyst case above, skip-gram associated a stemming false negative and gave a specific catalysed compound to the 'catalyst' test word than more obvious, fundamental associations

also noted that CBOW had stronger agreement between euclidean and cosine similarity metrics, however, euclidean similarity gave poorer general perform.⁴. It is noted that cosine similarity is the accepted similarity metric in the literature [?] [?] [?].

5.2 Document Similarities

The models detailed in §4 were then tested for document vector similarity. A document was chosen from the corpus, the 3 most similar articles were computed for each model and results assessed. One test document was DOI: 10.1134/s0036024412120266 [?], the title:

‘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, and gave poor predictions. The remaining models’ most similar documents⁵ for this test document are shown in table 5.3:

⁴‘NRL’ in the ‘Plastic’ category of skipgram is a typical example of poorer euclidean performance. NRL appears to be a reference to the Navy Research Laboratory.

⁵Cosine similarity was used, as it performed better than euclidean similarity for document vectors.

Table 5.3: Document Vector Similarities to [?]

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 polymeriza- tion	10.1080/ 002223381 08074381	Cyclic Acetal- Photosensitized Poly- merization. 9. Photopoly- merization 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 acceler- ators 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 photopolymer- ization with visible-light photoinitiators: Acrylamide polymerization photoiniti- ated 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 performance, and thus were taken forward as the model of choice for further analysis.

5.3 Visualisation Techniques

5.3.1 Network Visualisation

High Dimensional systems are hard to visualise but there are several methods available to visualise high dimensional data. PCA⁶[?], a well established technique reduces dimensionality by a series of orthogonal transformations. T-SNE⁷, a state of the art technique, reduces dimensionality by preserving spatial clusters of vectors at high dimensions[?]. These techniques allows 100-dimensional document vectors to be collapsed to points on an arbitrary 2D plane, to give a visualised ‘snapshot’ of the semantic space. Figures 5.1 and 5.2 show PCA and TSNE reductions ⁸ on 10,000 document vectors randomly

⁶Principle Component Analysis

⁷t-distributed stochastic network embedding

⁸performed using scikitlearn and python

selected from the training dataset [?].

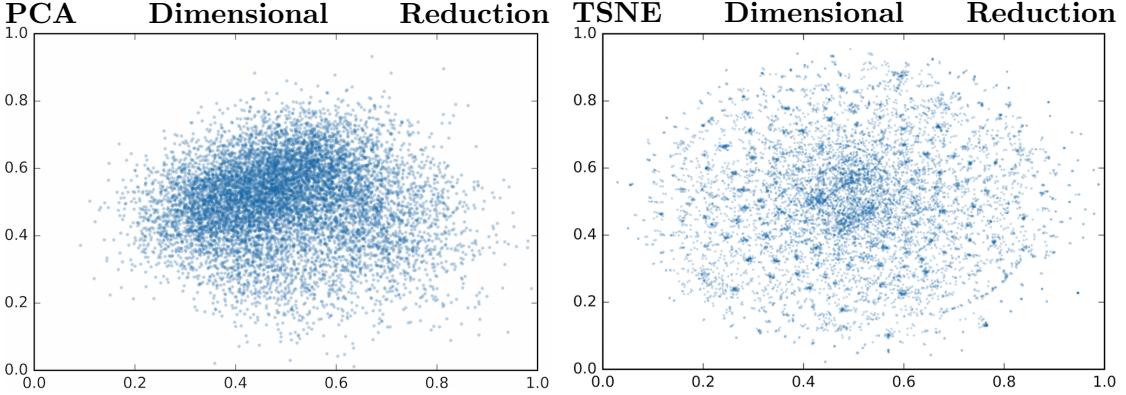


Figure 5.1: PCA map of 10,000 documents in the corpus. PCA has not any particular structure. The dimensional reduction task is probably too difficult for PCA.

Figure 5.2: TSNE map of the same 10,000 documents. Document vectors have gathered into noticeable clusters, with non negligible outlier documents between clusters.

The PCA reduction shows a dark central area, suggesting most vectors are ‘smeared’ about a common direction. The map is not entirely symmetric which is what would be expected for random vectors. It was expected that document vectors would be distributed into clusters representing particular research fields within the literature. This is borne out by the TSNE reduction, which has resolved many clusters. There is a significant portion of document vectors scattered between dark cluster spots, which may be could interpreted as ‘interdisciplinary’. TSNE is based upon euclidean distance, which is noted not to be the best similarity measure so, whilst qualitatively useful, TSNE maps were interpreted cautiously.

5.3.2 Networks and Network Visualisation

A similarity matrix \mathbf{C} was defined between a sets of documents. For a set of documents A (with a documents) and B (with b documents) define document matrices of document vectors \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. The cosine matrix \mathbf{C} is then defined as $\mathbf{C}_{i,j} = \text{cosine}(\theta_{i,j})$ where element $i j$ contains the cosine between i th document vector in A and

j th document vector in \mathbf{B} :

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

where \oslash and $\circ \frac{1}{2}$ indicate Hadamard division and Hadamard square root⁹, $\text{diag}(Q)$ the $1 \times n$ matrix formed from the diagonal of matrix Q . This matrix represents a network where each document in \mathbf{A} is a node with an edge to every document in \mathbf{B} with a weight equal to the cosine. If \mathbf{A} is \mathbf{B} , then the matrix is a fully connected network¹⁰. This network can be visualised using specialist software¹¹ [?]. Figure 5.3 visualises the same 10,000 document sample as a network graph.

⁹Elementwise division and Elementwise square root

¹⁰That is to say, every document node has an edge to every other document in the set

¹¹Gephi, a powerful, open source network visualisation and analysis suite, was used for this purpose.

Network Visualisation of 10,000 document sample

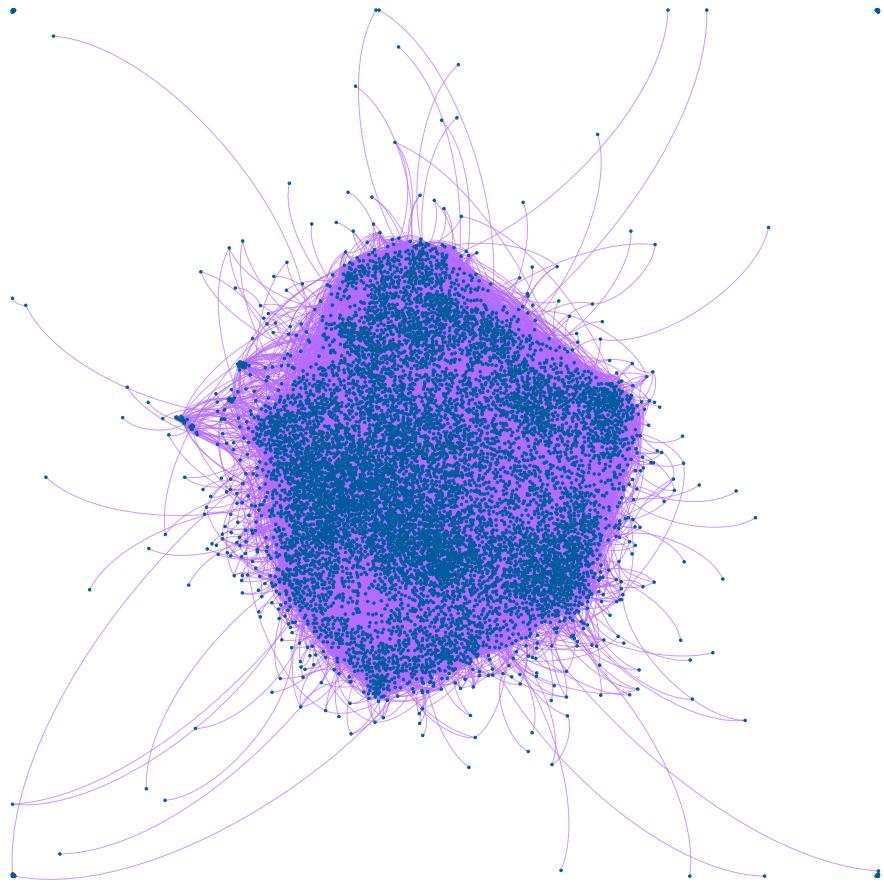


Figure 5.3: A Network visualisation of the 10,000 document sample. Nodes (blue) are spatially distributed by modelling the edges (purple) as springs connecting nodes with spring constants equal to cosine similarity, then allowing the system to approach equilibrium. Edges were only placed between nodes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

It can be seen that concentrations of documents also form in the network visualisation. There are noticeable outlier documents far from the central clusters¹². 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 analytic algorithms to be applied.

¹²These articles are predicted to be short, or qualitatively different from the majority, e.g. addenda or retraction notices, rather than proper articles.

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 project¹. With this in mind, it was decided to focus analysis on the a smaller subset of the training dataset, namely documents from the University of Cambridge Chemistry Department. This dataset is henceforth referred to as *CCD*².

6.1 Cambridge Chemistry research clusters

The CCD 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 high throughput modularity algorithm[1][2]. The result is shown in figure 6.1.

¹Please refer to §8 for recommendations for further work

²Cambridge Chemistry Dataset

CCD Network Visualisation

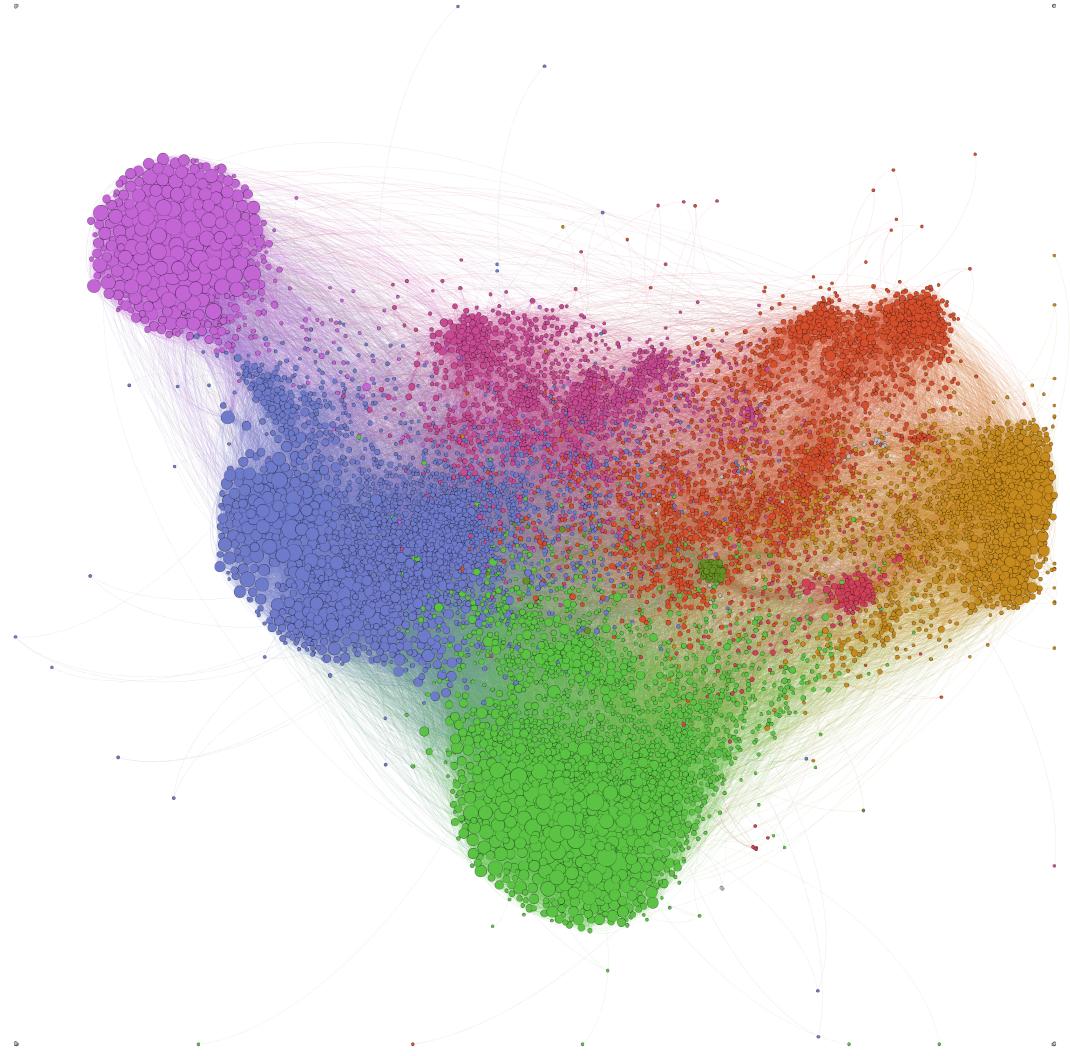


Figure 6.1: A Network visualisation of the CCD. Edges were placed between nodes with weights corresponding to cosine similarity if S_{cosine} . Nodes are coloured by their detected communities, and node size is proportional to the number of connections a node has. nodes are arranged by modelling edges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

It is apparent that the CCD contains clear communities of documents. This corresponds to different fields of research within the department. Some communities detected were small, but some quite large (green, orange, etc...). The algorithm was then applied only to ‘green’ community, which revealed subcommunities within the ‘green’

documents. A program was then written to recursively detect subcommunities in the CCD. This resulted in the CCD being divided into 300 communities of comparable size. The smallest communities were singleton documents, the largest community 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

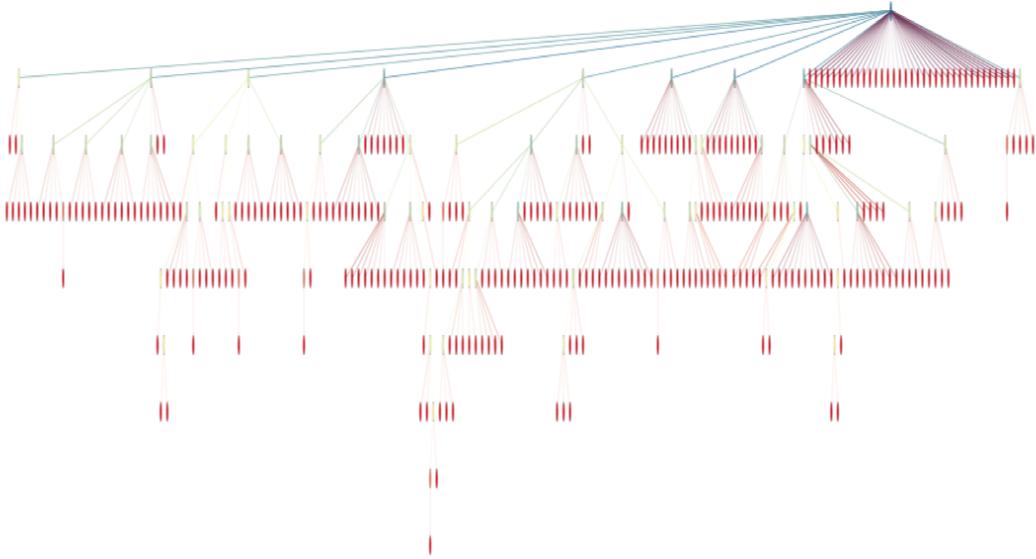


Figure 6.2: Recursion Tree for how communities were derived. The dataset was partitioned using the modularity algorithm. Sets with more than 100 documents were then repartitioned recursively. Sets of less than 100 documents were considered to be communities (red nodes in the diagram). If the algorithm 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 8, and most communities were found after 3 partitions.

Figure 6.2 can be interpreted as showing the *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 described is constitutes an unsupervised categorisation algorithm. The entire process, from model training to finding communities has been performed without human labelling or intuition. It was therefore instructive to examine what the algorithm defined as communities. Communities were examined closely, to reveal that community clustering made

intuitive sense in the majority of cases. Community 275 is typical:

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 <i>Antirrhinum majus</i>
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 flowers³. The connections mostly make sense. Note the surprising inclusion of the cosmetics and lanthanide silicate studies. Upon investigation, both studies use very similar analytical techniques used elsewhere in the community, and both examined intercalation.⁴

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

- Analysis of literature field - plotting trees such as figure 6.2 can give a relational understanding of how facets of a field link up together.

³Note 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

⁴Both 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.

- Research tool: If researching a paper, identifying its community immediately provides the researcher with papers that are related to it. Crucially this is done without simply following citations, so that interesting, perhaps overlooked links between papers 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 the ‘summary’ papers.

6.2 Cambridge Staff Member Similarities

It is not only articles themselves that can be grouped and analysed, but articles can be aggregated together to represent higher order concepts, such as staff members or research groups, or potentially even departments.

To investigate this further, <http://www.ch.cam.ac.uk/publications/authors> was scraped in order to associate the documents in the CCD with particular staff members. A staff member vector \mathbf{f} was defined as $\mathbf{f} = \frac{1}{N} \sum_i^N \mathbf{v}_i$, for an author with N published articles in the CCD, with document vectors \mathbf{v}_i (vector mean).

To investigate author relationships, a cosine matrix was created for each pair of authors A and B, with α and β documents respectively, $\mathbf{C}^{(A),(B)}$ (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{S} , with elements $\mathbf{S}_{A,B} = S_{A,B}$. 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 UPGMA⁵ was applied [3]. This method clusters the authors pairwise in a hierarchical fashion. An effective visualisation of the similarities between staff was to plot a *clustermash* [4] [5].

⁵Unweighted Pair Group Method with Arithmetic Mean

Cambridge Author Similarity Clustermap

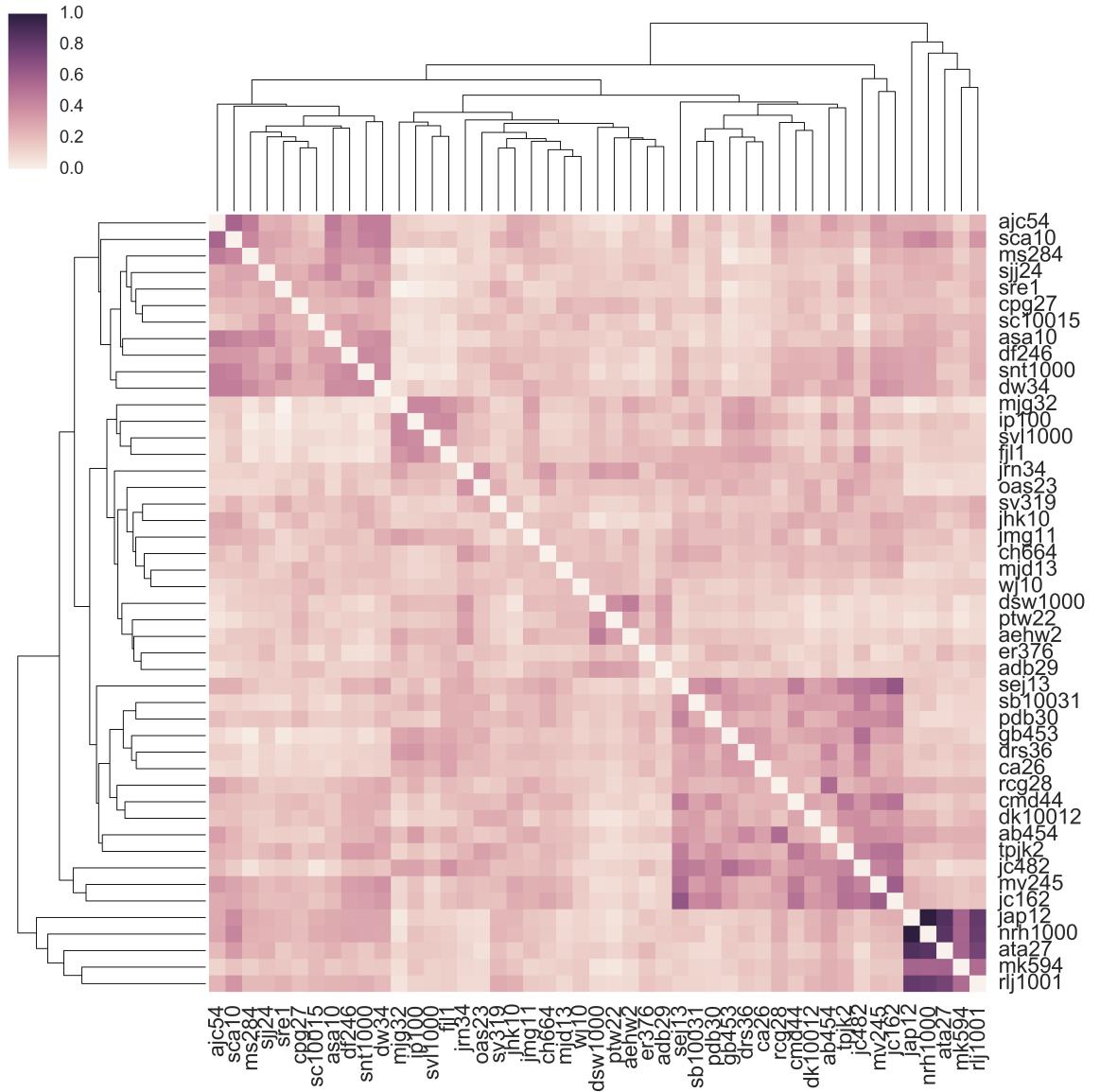


Figure 6.3: This figure shows a heatmap of author similarity. Dark pixels correspond to the author in the pixel's row having similar research interests to the author in the pixel's column. The matrix has been scaled to the range (0 → 1). The authors are arranged by clusters found in UPGMA. The hierarchical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

Figure 6.3 shows the result of generating \mathbf{S} and performing UPGMA hierarchical clustering. The authors are labelled by crsid. The dendrogram tree links authors pair-by-pair, illustrating how the clustering was performed, and how closely related clusters are. An enlarged dendrogram is shown below:

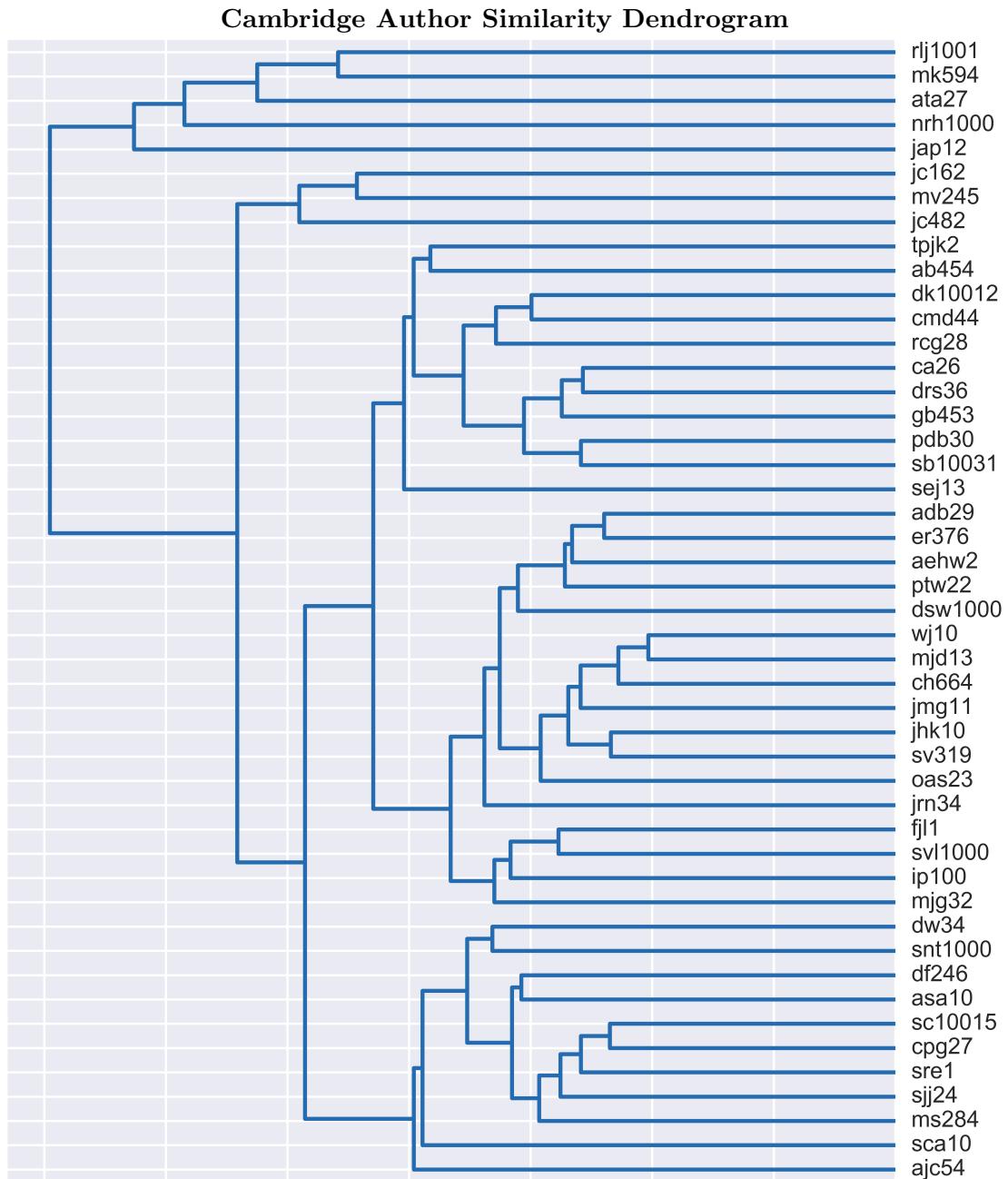


Figure 6.4: The dendrogram of figure 6.3
plotted for clarity

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

the rest of the department. The staff members involved (Professors Jones and Pyle, Drs. Harris, Archibald and Kalberer) are all members of the Centre for Atmospheric Science. The unsupervised model thus successfully ‘predicted’ their department, and indicated that their work is quite separate from most of the work in 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 9.3. Most clusters make intuitive sense, but there is one core of well connected, more disparate members (wj10 to jrn34). These members could be interpreted as forming an interdisciplinary cluster.

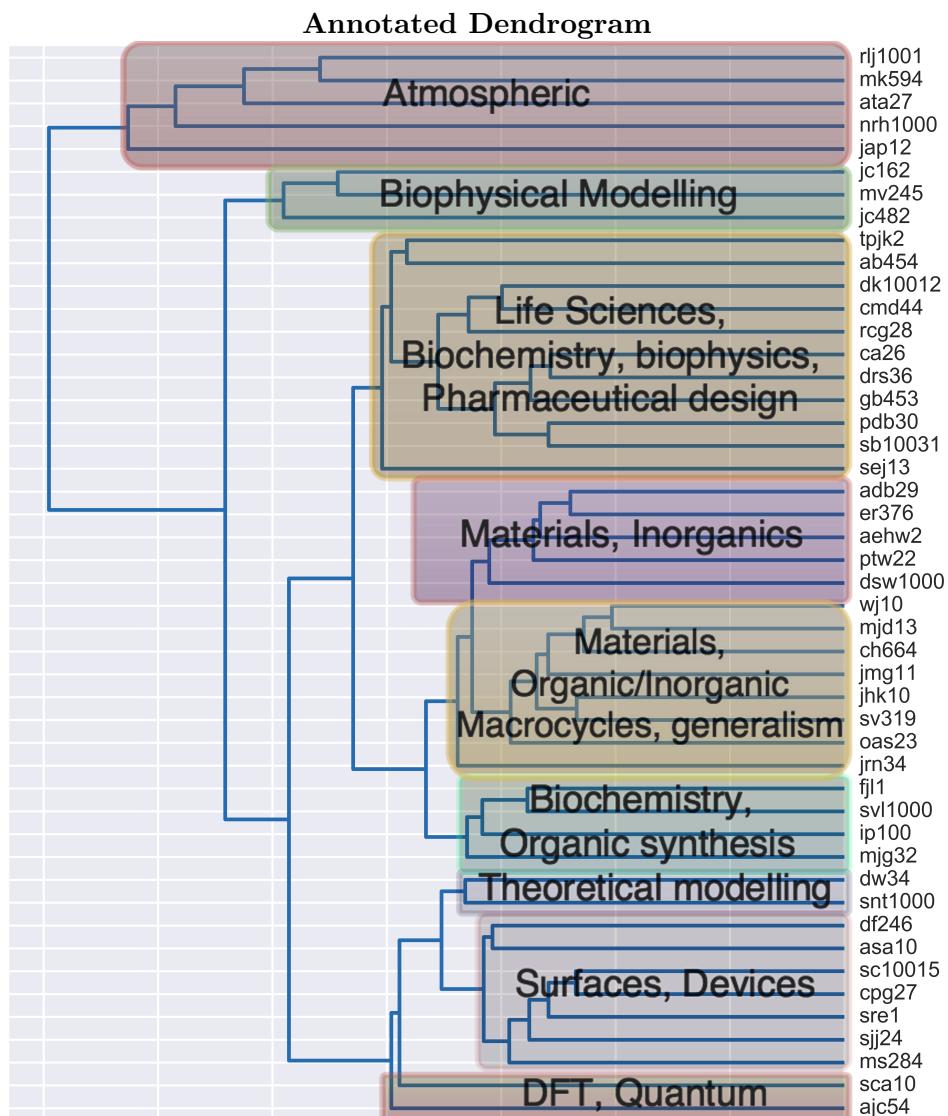


Figure 6.5: Cluster labels overlayed over the distinct branches of the dendrogram.

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 or bias. Thus 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 $\mathfrak{C} \cap \mathfrak{A} \neq \{\}$,⁶. The function f_{assoc} was defined as

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

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

⁶or equivalently $\exists \partial : \partial \in \mathfrak{C} \wedge \partial \in \mathfrak{A}$

Author Community Spread

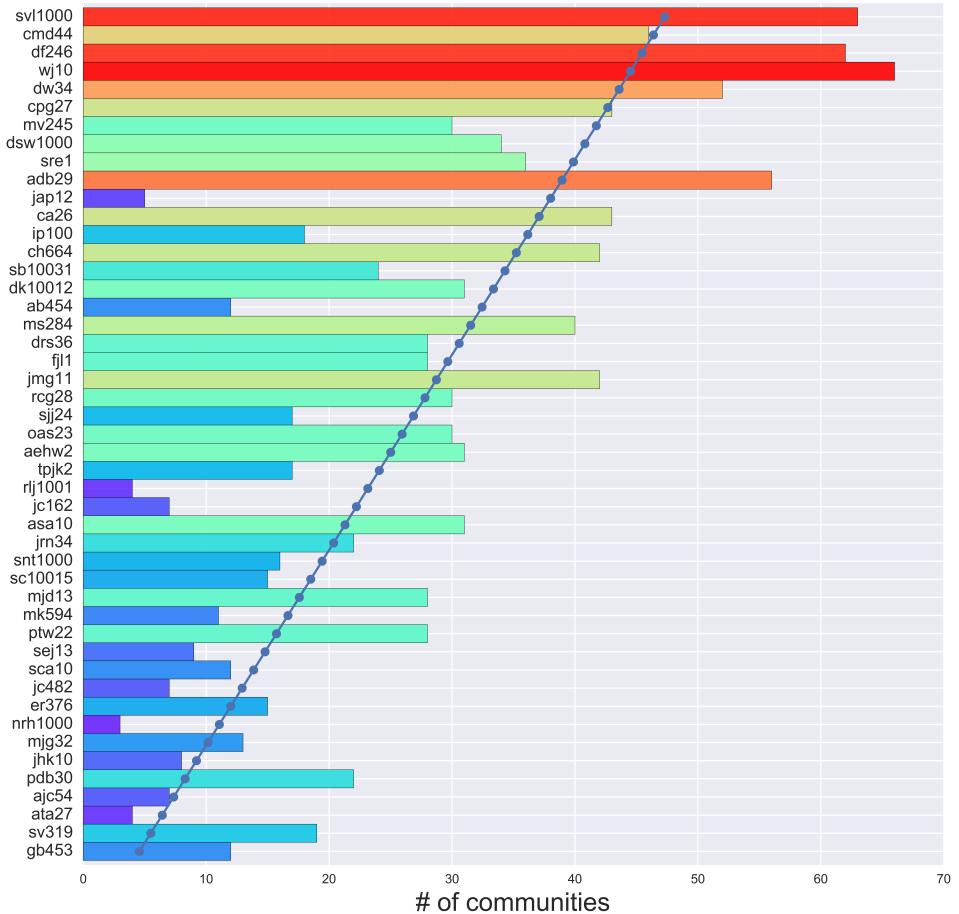


Figure 6.6: Number of research communities authors are associated with. High values (towards red) indicate an author publishing across many communities, suggest more interdisciplinary work, but also higher publication count per author. The authors are ordered by publication count (highest at top). As 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 are widely distributed between communities, whereas others are concentrated. It should be appreciated that communities are not uniformly distributed. For example there are 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 tentative⁷.

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

$$S_{coincidence}(\mathfrak{A}, \mathfrak{B}) = \sum_c^C (f_{assoc}(\mathfrak{C}_c, \mathfrak{A}) f_{assoc}(\mathfrak{C}_c, \mathfrak{B}))$$

Where C is the total number of communities. An association matrix was created, $\mathbf{S}_{\mathfrak{A}, \mathfrak{B}}^{Assoc} = S_{coincidence}(\mathfrak{A}, \mathfrak{B})$, 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: $\mathbf{S}_{\mathfrak{A}, \mathfrak{B}}^{Assoc, scaled} = \mathbf{S}_{\mathfrak{A}, \mathfrak{B}}^{Assoc} / (\mathbf{S}_{\mathfrak{A}, \mathfrak{A}}^{Assoc} + \mathbf{S}_{\mathfrak{B}, \mathfrak{B}}^{Assoc})$, and scaled from $0 \rightarrow 1$. This was a measure of how often authors could be found in the same communities. The matrix is shown below:

⁷It should also be noted that 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 1 paper published in a community

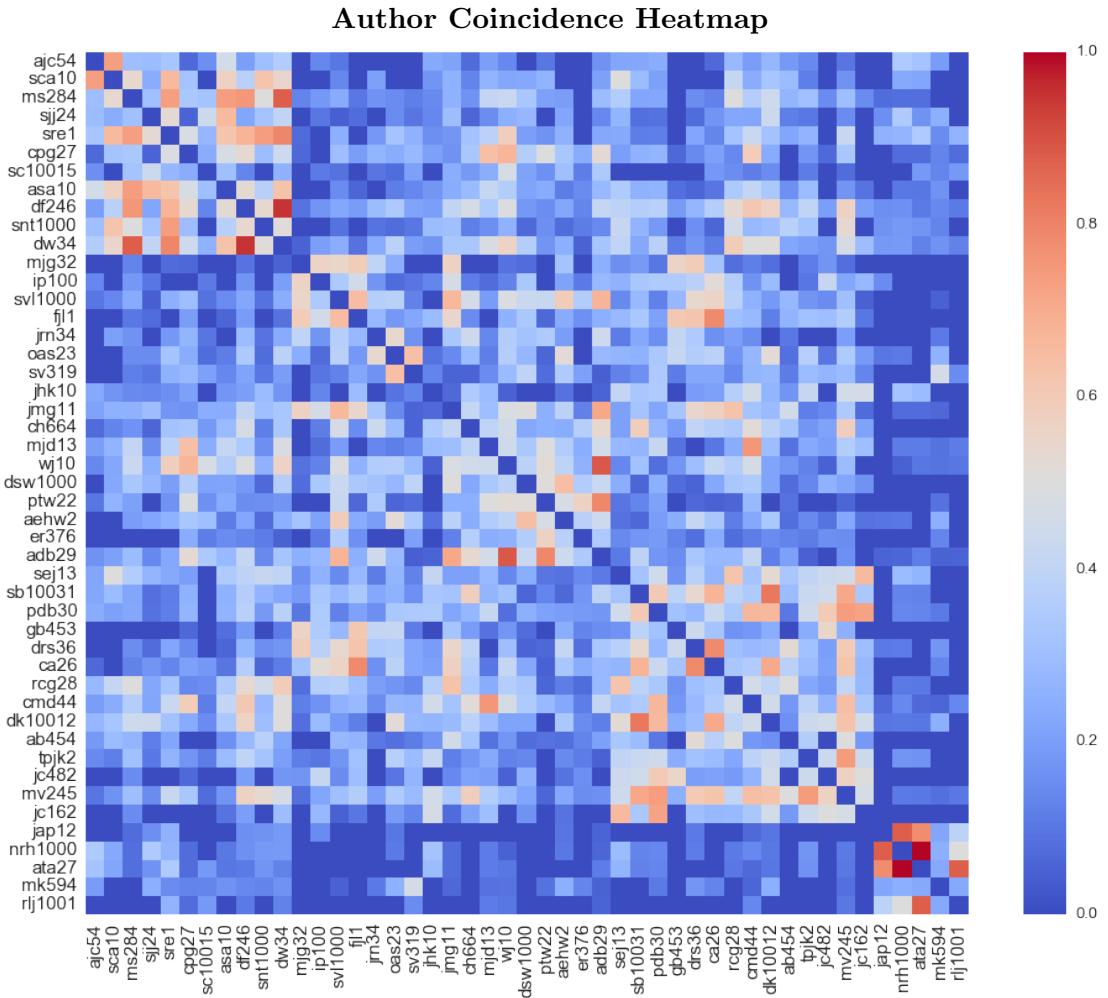


Figure 6.7: Heatmap showing author-author pair values for how often author pairs publish works in the same communities. High values indicated that authors are predicted to have similar publication profiles. Note the authors are arranged with the ordering from figure 6.3.

Figure 6.7 displays where authors have been detected to 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 that the square patterns of higher values close to the diagonal of the map reproduces the clustering in figure 6.3, lending weight to the validity of both analyses.⁸.

⁸This 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 the authors). Because clustering is qualitatively visible in figure 6.7, there is a correlation

Having defined a framework for finding where authors share research interest, the next step was to find where authors were *actually* collaborating. It was possible to identify approximately 700 documents in CCD that were co-authored by staff members in the analysis. A heatmap for actual collaboration between authors is shown below, as well as a metric equivalent to the $\mathbf{S}^{assoc,scaled}$ with elements as the sum of the number of communities both staff members have collaborated in.

between the two methods, i.e. they are consistent

Author Collaboration Heatmap

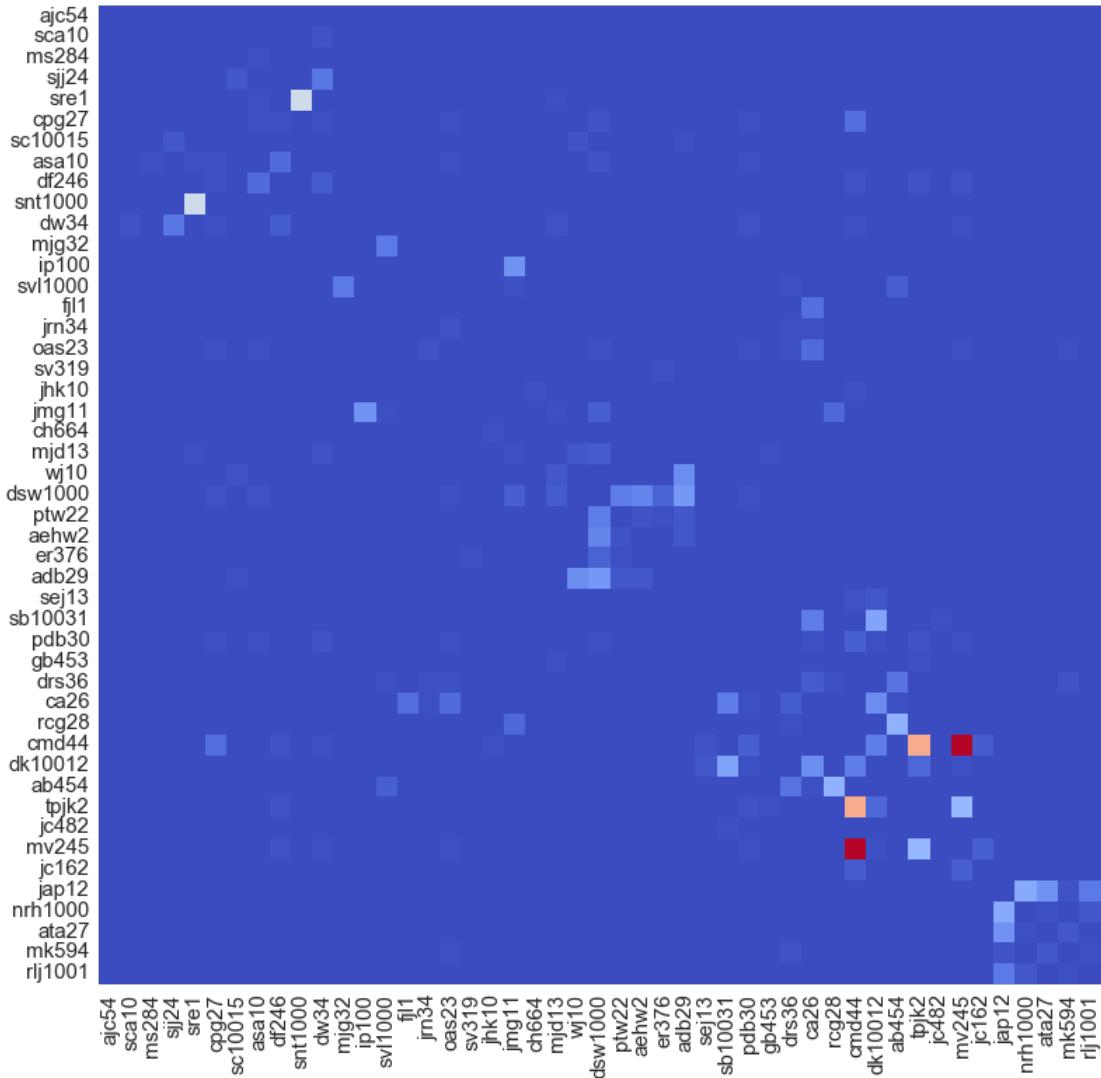


Figure 6.8: Raw collaboration matrix (values scaled to range $0 \rightarrow 1$). Note the general lack of co-publishing between staff members. Again staff are ordered by clustering described in §6.2, but no actual clustering has been performed. Hot spots near the diagonal suggest that author pairs clustered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

Both pictures show the same qualitative picture. 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, 6.7) and an author collaboration matrix (e.g. figures 6.8, 6.9)

Community-summed Author Collaboration Heatmap

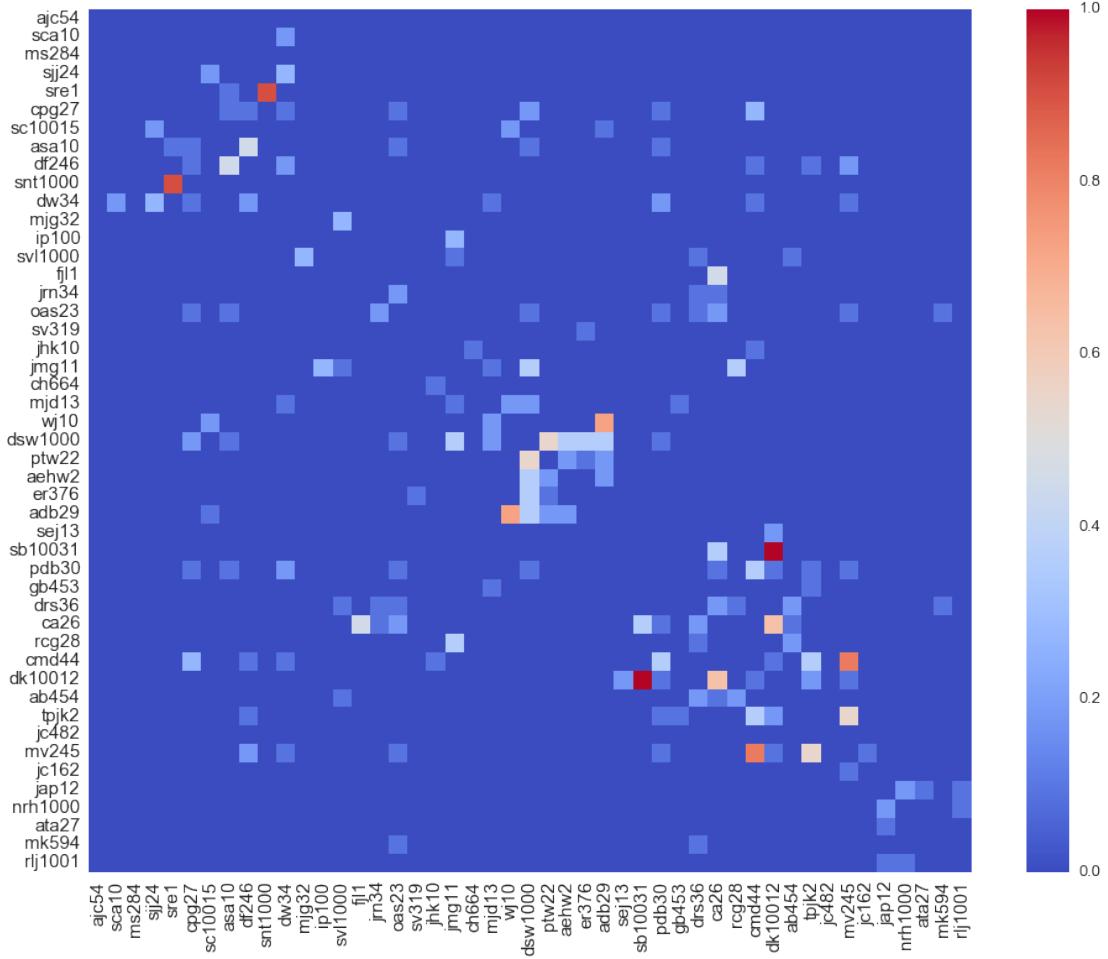


Figure 6.9: Matrix formed by summing collaboration of author pairs over research communities (values scaled to range $0 \rightarrow 1$). Qualitatively similar to figure 6.8. Hot spots near diagonal again suggest authors closely clustered in §6.2 collaborate more frequently

could be interpreted as a *recommended collaboration matrix*, 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. Author Pairs with values to 1 should be encouraged consider working together. This matrix is shown below:

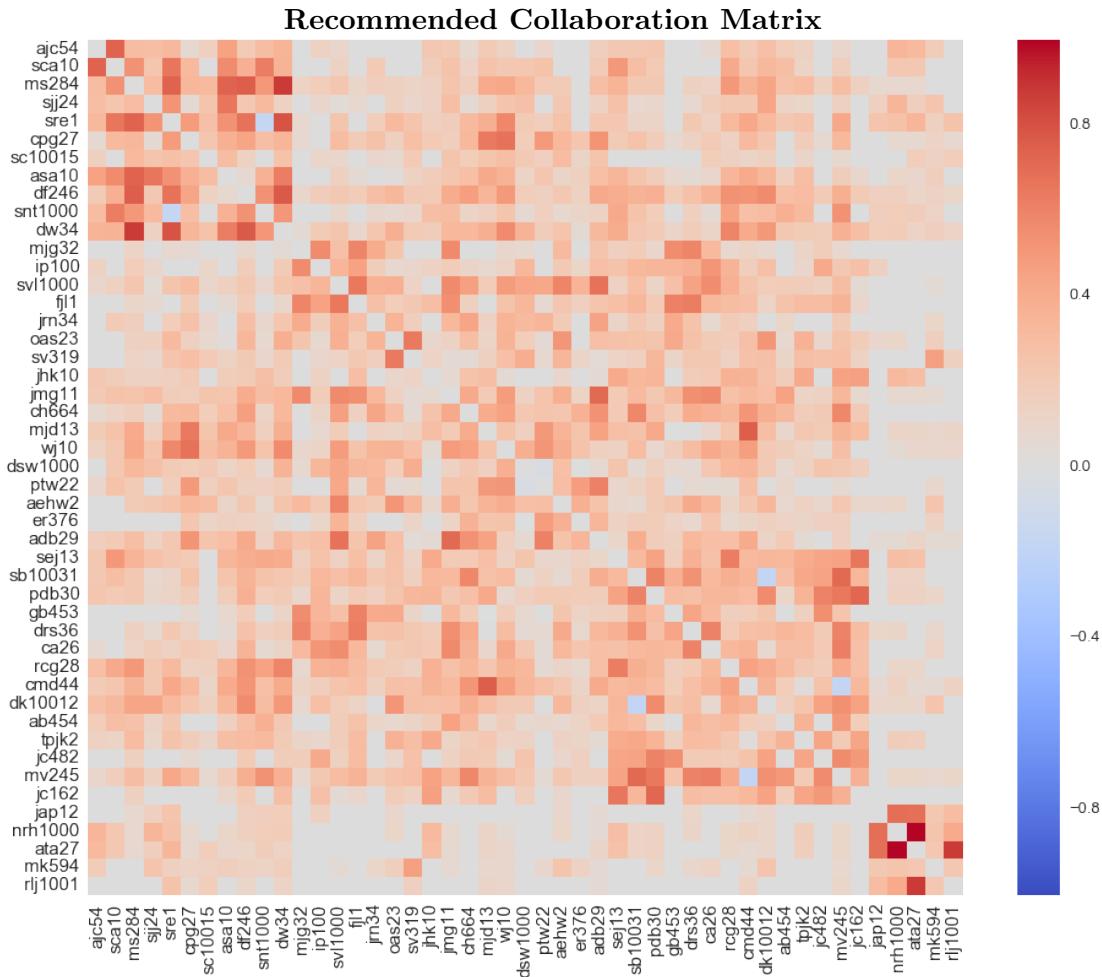


Figure 6.10: High values (Deep red) indicate authors that have similar research but for which there is little evidence of collaboration on published works. Values ~ 0 (grey/white) are where authors are neither similar nor collaborate, or are similar and collaborate closely. Values towards -1, (Blue) 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. Returning to the Centre for Atmospheric Science, which was highlighted as a tight, separate research community, it can be seen that there are recommendations for greater collaboration between particular authors within the Centre. Table 6.2 shows the top 20 scores in the Recommended Collaboration Matrix, where there is stronger evidence to suggest these author pairs should collaborate but little evidence that they are collaborating.

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 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

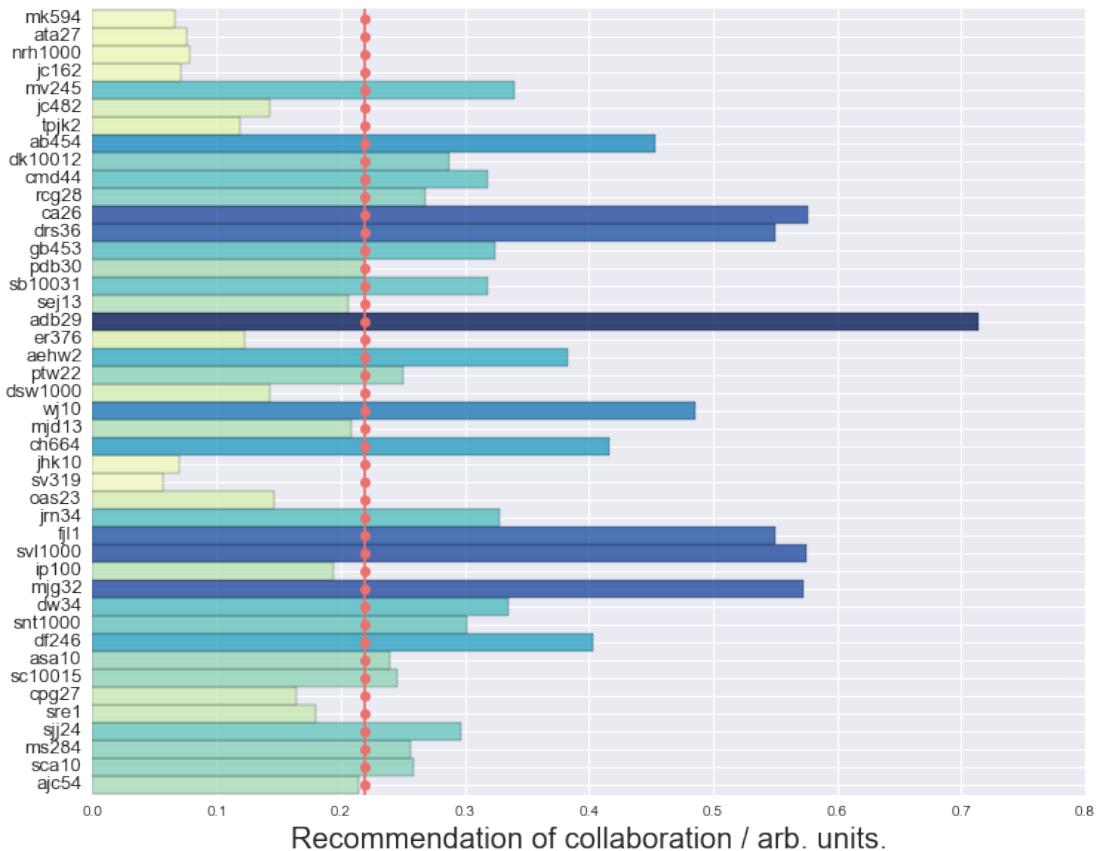


Figure 6.11: Recommendations for a particular staff member from the recommendation matrix, plotted in bar form, (Professor Goodman). (Bars very close to zero have been removed). Large values (Long bars, deeper blue) indicate the author publishes in many similar communities to Professor Goodman, but little evidence was found of collaboration, thus the recommendation to collaborate. The 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 new, constructive debate and promote effective collaboration in the department. The analyses presented in this section are not exhaustive, and there is potential for more fruitful insights to be found. Please see §???. 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. It is also possible some co-authorships could not be resolved due to data incompatibilities between databases.

7. Conclusions

Focusing first on the data acquisition phase, the scraping procedure was regarded as 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 a database of 16363). 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 chemistry departments did not host their own website. This precluded large parts of many important departments. Identifying potential webpages to scrape could have been implemented more effectively. However, the data that was successfully resolved was high relevance, with few false positive inclusions. The scraping program was robust and efficient, and performed well.

The large-scale scraping can be regarded as a success as the data collected was sufficiently populous and chemistry-specific to enable effective models to be trained. It should also be highlighted all of datasets created were from freely available sources, requiring no subscription and could be collected by anyone. This said, it must be acknowledged that most publishers discourage automatic scraping, and the publisher banning was considered as a major failure in the project. That said, it was dealt with swiftly, and did not present a lasting issue.¹.

It should be mentioned that there are existing metadata stores available (such as PubMed). Whilst using one of these datasets would certainly have been easier to use, (and are considerably larger), there was no real available dataset spanning *chemistry* with enough width of data. The Training dataset, whilst taking considerable time and effort to create, was heterogeneous and thus was a more suitable tool.

The algorithmic development section can be regarded as successful. The premise of quantitative vectorial representation of chemical articles was realised, especially by the doc2vec model. It should be mentioned the TF-IDF models failed to produce well behaved vectors, which is not well understood. The success of the model can begin to be seen in the Analysis section, where it's clustering performances were intuitive and instructive. The potential of the models has not fully explored. It is the author's

¹The author wishes to thank the librarians and Professor Goodman for efficiently addressing the problem

opinion that another project could be filled developing further uses of the training data set and extending the methodologies presented. Some model design choices may have limited specificity, such as the decision to use 100 dimensional vectors for computational tractability.².

The analysis that was performed is most interesting, but the usage to chemists is somewhat 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 somewhere.

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.

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

8. 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 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.

8.2 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 [?] [?]. Furthermore, is is possible to predict chemical properties using semantic relationships found in the literature? $\text{Vec}(\text{Compound A}) + \text{Vec}(\text{Compound}$

¹A digital copy is included with this dissertation.

²The author recommends 400 dimensional vectors

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

B) + Vec(Lab Technique) may give vec(Product C). If so, it may be possible to find unexpected reactions. This could be coupled with the RInChI database to form a new type of data-driven cheminformatics.

8.3 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.

8.4 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.

8.5 Structure stemming

Chemical names could be smartly preprocessed to classes of chemicals e.g. 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.

8.6 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 rich⁴

⁴For example, which documents are close to the indium-catalysed vector but do not contain the word indium...

Bibliography

- [1] Vincent D Blondel, Jean-Loup Guillaume, and Etienne Lefebvre Renaud Lambiotte. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:1000, 2008.
- [2] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian dynamics and multi-scale modular structure in networks. *IEEE Transactions on Network Science and Engineering*, 1:76–90, 2015.
- [3] Sokal R and Michener C. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.
- [4] 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>.
- [5] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL <http://www.scipy.org/>. [Online; accessed 2016-03-10].

9. Appendix

9.1 UK Departments scraped

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 Polymer 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.sbcn.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

9.2 Publishers Considered in UK scraping

Publishers collected in 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

9.3 Scaled Communities for staff members in Cambridge

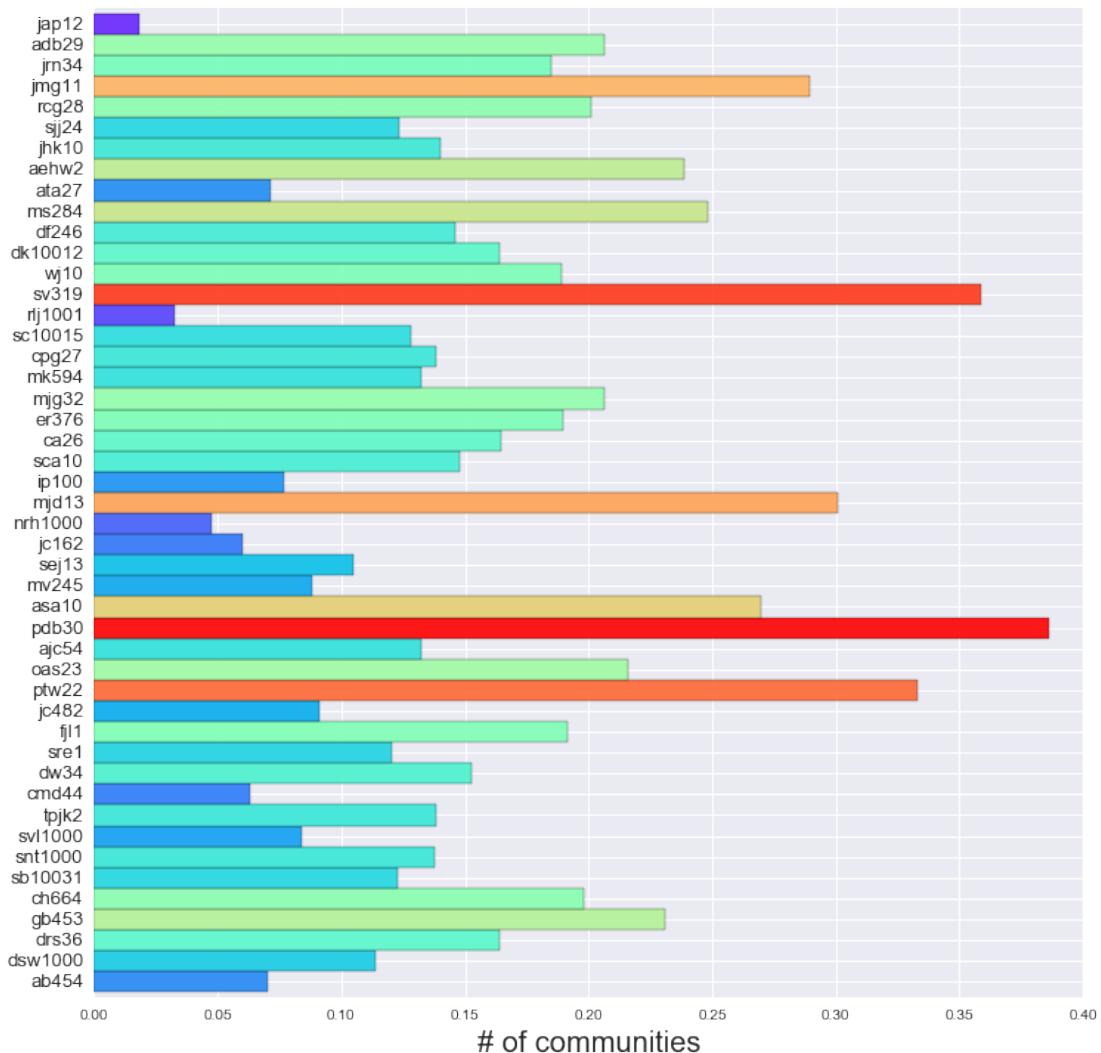


Figure 9.1: Number of research communities authors are associated with. High values suggest more interdisciplinary work. The bars have been scaled relative to the author's total publication count.