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, learn`ed institutions (such as university websites) began to post records of recent publications and other chemical information freely online. Publishers still protect the vast majority of journal article content and some meta-data. Data is valuable, and the insights within, powerful. As such, publishers are unwilling to grant free access to their data, preferring to perform in-house analysis. Article metadata, such as authors, titles and abstracts may, however, be available, and it is this dataset which the project is focussed on.

By collecting meta-data on papers found on the internet, a large representative dataset of chemical academic writing language can be built. Machine Learning techniques can be applied to find novel connections between articles, research communities, authors, institutions and fields. Machine Learning is a rapidly progressing field and data science can reveal key, non-obvious relationships to aid the scientific process. In an increasingly data-dense world, scientists require smarter tools to streamline research in order to be more productive. Several publishers provide services that perform large-scale analysis and provide literature tools, such as SciFinder⃝R and Web of KnowledgeTM. 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.

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 identify and extract chemical informa tion
  + –  Store information in human-readable, computer readable, stable, scalable formats
* Develop novel machine learning techniques to enable meta-data to be interpreted in new ways
  + –  Sanitise input data effectively
  + –  Devise machine learning models to interpret article titles and abstracts to  attempt to extract their chemical meaning
  + –  Quantitatively represent an article’s content using its collected meta-data
* Validate the models and provide evidence of their efficacy
  + –  Develop visualisation techniques for interpretation of algorithm output
  + –  Analyse datasets using the developed model to demonstrate new and useful information
  + –  Provide usable code with which future analysis may be performed  This project is thus an informatics/data project, which split naturally into two sections. The first half of the project was concerned with acquiring data. This is covered in detail in §2. Programs were written in the Python programming language, and two databases were created (one of UK Department chemistry, and a very large database of unrestricted chemistry-related material).  Once the databases were set up, focus was shifted to how to use the data to find valuable insights. §3 and §4 provide the background of the algorithms used and the process of applying them to create useful models.  Having built the models, it was now necessary to examine their outputs and develop methods to interpret results, covered in §5. Finally, when the models were shown to be performing successfully, they were used in an analytical setting to examine relationships between authors and research communities in the University of Cambridge Chemistry Department, and eventually to recommend specific collaborations between staff (§6).

Internet webpages are written in HTML. When a webpage is accessed, the HTML code is sent to the user, and the browser processes and displays the webpage in a humanreadable format.

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

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

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

The initial approach was to analyse the HTML tree to automatically recognise useful data generate XPaths1. When this strategy proved unsuitable, a new method was required. Chemical information is usually disseminated as journal articles accompanied by a DOI. By programmatically collecting DOIs, (§2.2.1) it was possible to build up a large database of chemical information (§2.2.2)

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

DOIs consist of a prefix and a suffix. The prefix is subdivided into the Directory Indicator 2 separated from the Registrant Code, assigned by the issuing body[24]. Registrant codes are a minimum of three integers, with further optional subdivisions separated by full stops. The suffix is provided by the registrant and can be any form of unicode-encoded text[24].

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

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

The REGEX approach does not require XPaths in order to extract DOIs from a webpage. This facilitates large-scale scraping from many websites. Some meta-data3 associated with a DOI can be accessed using an online API exposed by Crossref. The remaining meta-data can be accessed by following the http://dx.doi.org/{DOI} link to visit publishers’ webpages.

With this methodology in place, a scraping program was written to collect DOIs from a list of webpages, collecting meta-data in a two stage process. The Crossref API provides article titles, journals, authors, publisher and publication date meta-data, but not article abstracts. These had to be collected by visiting publisher webpages, and collecting with hand written XPaths4. The procedure is summarised in figure 2.4.

The programmatic steps depicted in figure 2.4 are:

1. Request the webpage from the inputted list 2. Process the html and extract DOIs 3. Using the Crossref Online API, verify the extracted DOIs exist 4. Crossref yields metadata:

• Title • Journal • Publisher • Authors • Publication Date

5. For each DOI, follow http://dx.doi.org/{DOI} link 6. Use XPath to collect article abstracts

The program exports complete records as .json files, but also feeds to a MongoDB database. Once the program was written, a list of webpages to scrape was required. §2.3.1 and §2.3.2 describe how this was achieved.

The program was first used to collect the data from the UK. The Goodman group’s website hosts a list of UK chemistry departments http://www-jmg.ch.cam.ac.uk/data/ c2k/uk.html. The list was manually checked and edited to give a list of 68 departments5. The program was run using this list, the results of which are detailed in table 2.1. The DOIs collected were stored in database ∆1 and the complete results were stored in database ∆2.

Conversion losses were due to four components. 45 losses for non-existant DOIs, 2753 to request errors (404 : not-found errors or permission problems), 133 to the program errors and 3148 to missing publication XPaths. The 26 specified XPaths 6 were sufficient to convert 83.8% of successful requests. This was deemed acceptable, as most major publishers had been covered7, and the missing publishers each covered a small number of articles8 The efficiency is depicted in figure 2.5.

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

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

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

At the intermediate point, the program had collected 1,267,495 records. This database was labelled ∆3. Publisher distributions and potential server loads were then carefully considered and capture probabilities were predicted before the second half of the scraping routine was set off to run for three days9.

Most publishers track request volumes sent to their servers to discourage automatic downloading. Scraping constitutes fair use and complies to UK copyright law. Despite the university owning a full-access licence to these publishers’ publications, the collected material was freely available without licence[25][26]. During the scraping run, a bug in the randomisation of requests resulted in detection by ACS and Taylor & Francis, which responded by banning the computer’s IP address 10. The department librarians were able to restore access, and it was agreed that no further scraping would be performed.

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

The overall efficiency of the process is 56.4%, but excluding lost ACS records, the program’s efficiency was 74.0%, similar to the efficiency of the UK scrape (§2.3.1)11.

The successful 714,370 records were merged with the UK results12. Records were rejected with short titles or abstracts, or if the majority of the title and abstract were not written in ascii characters13. This was done to provide higher-quality data for algorithm training (see §4). This filtering resulted in a final training database of 464712 articles. This dataset was labelled ∆6. The database formation process is summarised in figure 2.7 and table 2.3.

It was instructive to examine these databases and derive some simple statistical results, which is explored in §8.8.2.

\

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

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 v. If document A contains word i 2 times, then vA,i = 2. A simple example is given below:

Table 3.1 shows vector representations for Documents A, B and C. The higher the scalar product of normalised vA ·vA, the more similar the documents are predicted to be

and so documents A and B are the more similar. The related bag of citations model sets vector components according to the presence of citations. Both models are used by the scientific publishing industry1,2.

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

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

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

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

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

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

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

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

The models created in §4 were then examined and assessed. As an unsupervised learning algorithm, it is challenging to assess model quality, due to a lack of concrete metrics for comparisons1. The Word2Vec development team tested with approximately 10,000 semantic and syntactic relationships (See Figure 3.2)[18],[19],[27]. The scope of this project does not extend to such elaborate tests. In the section, some examples of model strengths are given and techniques for using word vectors and visualisation are presented.

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

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

CBOW and skipgram models were examined using these metrics. For a given word, the models were requested to return the three words in the corpus with highest similarity.

Some examples are given in tables 5.1 and 5.2 the

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 way3.

It was observed that the skipgram model gave misleading positives more frequently4. CBOW was considered to be superior for word-word comparisons. It was also noted that

CBOW had closer agreement between Seuclid and Scosine, however, euclidean similarity gave poorer general performance.5. It was noted that Scosine is the accepted similarity metric in the literature[18],[19],[30].

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

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

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

High Dimensional systems are difficult to visualise but there are several methods available to visualise high-dimensional data. PCA7[11] and TSNE8[14],[34] techniques allow 100-dimensional document vectors to be collapsed to points on an arbitrary 2D plane, to give a visual ‘snapshot’ of the semantic space. 10,000 document were randomly selected from ∆6 (referred to henceforth as ∆R). Figures 5.1 and 5.2 show PCA and TSNE reductions on the Doc2Vec vectors of ∆R[35]

The PCA reduction shows a dark central area, suggesting most vectors are ‘smeared’ about a common direction. The map is not symmetric which is what would be expected for random vectors. It was expected that document vectors would be distributed in clusters representing particular research fields within the literature. This is indeed seen in the TSNE reduction, which resolved many clusters. There are document vectors scattered between dark cluster spots, which may be could interpreted as ‘interdisciplinary’9. TSNE is based upon euclidean distance, which is noted not to be the best similarity measure. Whilst qualitatively useful, TSNE maps were interpreted cautiously.

A matrix C was defined between sets of documents. For a set of documents A (with a documents) and B (with b documents) document matrices of document vectors were defined, A and B, such that where w and v are document vectors. C was then defined as Ci,j = cos(θi,j) where element i,j contains the cosine between ith vector in A and jth vector in B:

where ⊘ and 2 indicate Hadamard division and Hadamard square root, diag(Q) the 1×n matrix formed from the diagonal of Q. C represents a network where each document in A is a node with an edge to every document in B with weights equal to the cosine. If A=B, then the matrix is a fully connected network10. This network can be visualised using specialist software11[36]. Figure 5.3 visualises ∆R as a network graph.

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 analysis algorithms to be applied.

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 a smaller subset of ∆6, documents from the University of Cambridge Chemistry Department. This dataset was labelled ∆7.

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

It was apparent that ∆7 contained clear communities. This corresponds to different fields of research within the department. Some communities were small, but most large (green, orange, etc...). The algorithm was then re-applied only to the ‘green’ community, which revealed subcommunities. A program was then written to recursively find subcommunities in ∆7. This resulted in ∆7 being divided into 300 communities of comparable size. The smallest communities were singleton documents, the largest was 434 documents, and the mean population was 34.5. The community-finding subdivision process is shown in figure 6.2

Figure 6.2 can be interpreted as showing relationships between different fields of research within the department. The tree is shallow with highly branched nodes, suggesting wide research fields, and much qualitative overlap between fields. The process constitutes an unsupervised categorisation algorithm2. It was instructive to examine what the algorithm defined as communities. Communities were examined and community clustering made intuitive sense in the majority of cases. Community 275 is typical:

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 silicate studies. Upon investigation, both studies used very similar analytical techniques used elsewhere in the community, and both examined intercalation4. 5

Note also that the mean vector for the community was closest to a paper in ∆6 that summarised the community extremely well6. This paper could be considered as a Summary paper. The uses of this kind of analysis include:

• Analysis of literature field trees such as figure 6.2 can give an understanding of how facets of a field link up.

• Research tool: If researching a paper, identifying its community immediately provides the researcher with related papers. This is done without following citations, so that interesting, perhaps overlooked, links can be found.

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

It is not only articles themselves that can be grouped and analysed. Articles can be aggregated together to represent higher concepts, such as staff members7. To investigate this further, http://www.ch.cam.ac.uk/publications/authors was scraped in order to associate the documents in ∆7 with particular staff members.

A cosine matrix was created for each pair of authors A and B, authoring α and β documents respectively (see §5.3.2). The similarity between the author pair

was defined as An author similarity matrix can then be built up, with elements

A similar technique to that described in §6.1 could have been used to create clusters of authors. Since the sample size was now much smaller (47 authors compared to 9467 papers) a more appropriate technique, Dedicated Hierarchical Clustering, specifically UPGMA was applied[15]8. This method clusters the authors pairwise in a hierarchical fashion. An effective visualisation of the similarities between staff was to plot a clustermap,[2],[3].

Figure 6.3 shows the result of generating MAuth.Sim. and performing UPGMA hierarchical clustering. The dendrogram tree links authors pair-by-pair, illustrating how closely related clusters are. An enlarged dendrogram is shown below:

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

rest of the department. The staff members involved9 are all members of the Centre for Atmospheric Science. The unsupervised model thus successfully ‘predicted’ their department, and indicated that their work is separate from most of the Chemistry Department. This is a real success for the model. The dendrogram was then further examined and broken into distinct branches. Each branch was examined and manually labelled (see figure 6.5). Most clusters make intuitive sense, but there is a core of well-connected, more disparate members (wj10 to jrn34). These members could be interpreted as forming an interdisciplinary cluster. The value of this method is self-evident. Clustering staff members informs the department about the width of research (number of clusters), and how resources are partitioned (size of clusters). It should also be stressed that authors are associated without any human preconceptions/bias. Perhaps the most valuable author associations are the unexpected ones, and authors should be encouraged to examine their cluster and consider their ‘neighbours’.

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 C, with documents and an author A with documents we can associate the author with the community if The function was defined as

It was noted that there was significant variation in the number of communities that researchers were associated with. A plot of for each author is shown below:

It can be seen that some authors were widely distributed between communities, whereas others were concentrated. It was noted that communities were not uniformly distributed. For example, there were many communities in ‘Life Sciences’ but few in Atmospheric

Chemistry, as such, interpretation of high values in figure 6.6 directly corresponding to wide research interests should be tentative10.

An association metric Scoincidence between authors A and B was then defined as

Where C is the total number of communities. An author association matrix was created, where high values for author pair A, B indicate

A,B they appear in many research communities together. The matrix was then scaled such

that and normalised from 0 → 1. This was a measure of how often authors published in the same communities. The matrix is shown below:

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

Having defined a framework for finding shared research interests, the next step was to find where authors were actually collaborating. It was possible to identify approximately 700 documents in ∆7 that were co-authored by staff members. A heatmap for co-authorship between authors is shown below, MRaw Collab. 12, as well as a metric equivalent to the MAuth.Coinc.scaled with elements as the sum of the number of communities in which both staff members have co-authored, MCommunity Collab..

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

As a final data step, a matrix defined as the difference between an author similarity matrix e.g figures 6.3 (MAuth.Sim), 6.7 (MAuth.Coinc.) and an author collaboration matrix (e.g. figures 6.8,MRaw Collab., 6.9,MCommunity Collab.) could be interpreted as a recommended collaboration matrix. 13

Author Pairs with values to 1 should be encouraged to consider working together. Matrix MRecommend(=MAuth.Coinc. −MRawCollab.) is one possible example, shown below:

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

collaborating14

The matrix row of MRecommend 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.

The aim is that these maps and plots may trigger constructive debate, and promote effective collaboration in the department15. 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-authorship16.

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

The data collected in global scraping was sufficiently populous and chemistry-specific to enable effective models to be trained. It should be highlighted all the datasets were created from freely-available sources, requiring no subscription. alternative scraping tactics may be required to avoid IP banning if further scraping is required. This said, it must be acknowledged that the publisher banning was considered as a major failure in the project1.

It should be mentioned that there are existing meta-data stores available (e.g. PubMed). Whilst using one of these datasets may have been easier, there wasnt a suitable available chemistry dataset with satisfactory breadth of data. ∆6 was heterogeneous and thus a more suitable tool.

Thee premise of quantitative vectorial representation of articles has been realised, especially by the Doc2Vec model. The TF-IDF models failed to produce effective vectors, which is not well understood. The power of the model can begin to be seen in §6, where clustering performances were intuitive and instructive. Some model design choices may have limited specificity, such as the decision to use 100 dimensional vectors2.

There is further potential within this project. Some further useful applications of the methodologies have been alluded to, some of which take the form of a service or tool rather than concrete insights (on-demand similarities, clustering, recommendations of articles to read, research profiling etc.). The project scope had to be limited at some point. There are detailed discussions of recommended work, and some further investigations/analysis in the appendix §8, especially §8.3 on chemical element analysis. 3 .4.

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.