Maastricht University

Department of Knowledge Engineering

Analysis of Publication Data in Nanotechnology

Master AI Research Project II

Report by:
Benjamin Schnieders
Dries de Rydt
Emanuel Oster
Ruben Schwarzwald

Maastricht, June 20, 2013

Contents

1 Introduction

This article on performing text- and data-mining techniques on publication data in the field of nanotechnology will provide an overview of the methods available and applied on a data set, then present results obtained by exploratory search and answer certain questions to the data set. In the following, these results are discussed and conclusions are drawn.

1.1 Nanotechnology

Nanotechnology is a broad research field touching physics, chemistry and biology, containing technology that is typically made of structures below the size of 100 nanometers. Developments in the field of nanotechnology might heavily influence other fields, such as medicine, constructions and computation.

1.2 Data Mining

The process to extract meaning from data is called Data Mining. It ranges from preprocessing raw data streams into tokens of a kind, to high-level constructs aiming at mimicking human understanding of text or relational data.

1.3 Existing software

Many software suites already exist to ease Data Mining related tasks. In the following, a few are described shortly.

WEKA WEKA, short for Waikato Environment for Knowledge Analysis is a collection of machine learning algorithms for data mining tasks [7]. It is an open source software implemented in Java, making it platform independent and portable. It contains tools for data pre-processing, classification, regression, clustering, association rules and visualization. Besides providing a toolbox of common learning algorithms, WEKA also provides a framework for researches to implement new algorithms without having to re-implement the supporting infrastructure for data manipulation. Since WEKA includes a wide variety of state-of-the-art learning algorithms and makes it easy to compare them, it is a widely used tool for data mining research. Despite this WEKAs graphical user interface is outdated by modern software standard and often takes a long time to compute results. Sources:

RapidMiner RapidMiner is an open-source, multi-platform data mining system [9][10]. It distinguishes itself from other data mining tools through its modern and intuitive User interface. RapidMiner provides many data mining operators, as well as access to most common data sources like Excel, SQL databases and many more. It also features the complete machine learning library of WEKA. RapidMiner does seem to suffer from stability issues when running on low-end machines. Crashes and system freezes occurred often, and were definitely a hurdle when trying to process the large dataset.

KNIME As an alternative to RapidMiner, KNIME extends it's general usage from regular datamining to the domain of text mining through usage of a few very useful textmining plugins [2][17]. Because it is heavily based on the eclipse framework, installing plugins is very simple. This way KNIME can be extended with WEKA functionality, all standard textprocessing tools and several more single-purpose datamining packages, such as chemistry related tools. One downside of KNIME is that it seems to be unstable on Linux-based operating systems. The application can suddenly close and processed nodes will not be saved.

FPM Apriori Apriori mines association rules and frequent item sets using the Apriori algorithm [3]. Finding closed and maximal itemsets is supported as well.

Gephi Gephi is a visualization tool for graph data [1]. It is a multiplatform, open source software. The basic functionality features a wide variety of filters, statistics and algorithms to draw the graph. More features can be added via plugins. Nodes and edged can be color-coded and sized based on values like degree, modularity class or eigenvector centrality. Once a graph is arranged in the desired manner, it can be rendered in high-quality and exported as png, svg or pdf.

TextToOnto TextToOnto is a toolset developed for the KAON framework, aimed at generating ontologies directly from raw text. This can be done either automatically or user-guided [11].

1.4 Specially designed tools

Due to the shortcomings of the existing software, several tools were implemented especially for dealing with larger datasets. The requirements were re-usability, stability, memory efficient storage and performance. This includes being able to load and write data both as raw text, for human inspection, as well as binary dumps, which are quicker to load. These tools are called individually by scripts and save their results after execution.
Besides filters and calling scripts, the tools are written in C++, with usage of the boost libraries [5].

1.5 Nanotechnology Database

The dataset this report deals with is a collection of about two hundred thousand articles in the field of nanotechnology. Each record consists multiple fields. The fields used for further analysis in this article are the following:

Title The raw text title of the article.

Author The author(s) of the article, separated by semicolons.

Abstract The complete abstract of the article.

Times Cited How many times this article was cited in other papers.

Keywords Plus A semicolon separated list of keywords for the article.

Publication Year The year of publication.

Title (NLP) (Phrases) Preprocessed phrases from the title.

Countries The country of publication.

The keyword field has shown to provide some good quality (high TF-IDF) terms, however, many of them are very specific (low TF). For keyword extraction, the title and abstract fields are the most useful. The text in them can be processed with various text mining methods, yielding more meaningful information, which can then be used for classification or clustering.
Authors, publication year and country served for information visualization.

2 Preprocessing

In order to perform classification or clustering on raw textual data, preprocessing has to be done. A bag of words representation consisting of the union of all words in the Title, Abstract, Keyword and Title Phrases fields was created as a preprocessing starting point. This yields 198816 different words. Using only words from titles and keywords creates a dictionary of 78605 words, which showed to be sufficient and of equal quality for further tasks.

2.1 Attribute reduction

Building a classification model using that many attributes is unsustainable. To reduce the set of keywords to the most significant ones, first a stoplist was considered. Inverse document frequency (IDF) is an easily obtainable measure ranking keywords that occur sparsely highly, and frequent keywords lowly. Empirical results have shown that dropping the most frequent keywords does very likely not remove any informative keywords [14].

As more sophisticated selection method, Shannon entropy was measured for each keyword, yielding an estimate on how well the remaining dataset could be described without it. All keywords which did not fulfill the entropy requirements, i.e., did not contribute to the dimensionality much, were removed. For the remaining keywords, cross-correlation was calculated against all other keywords. If a keyword correlates with another to a high degree, it can be regarded as redundant, so only keywords were chosen that did not correlate highly with any other keyword. Redundant keywords are thus merged to one, as also described in [12]. IDF, entropy and correlation thresholds can then be chosen to select only the most significant keywords.

2.2 Data sparsity considerations

Word vector representations of natural text produce extremely sparse matrices. Storing them in an index/inverse index structure is much more space efficient. Using Cosine Distance as distance measure has shown to be a necessity, as smaller differences in distances in a space with many hundred dimensions can often not be detected any more due to floating point imprecision. Using Cosine Distance, or - for binary vectors - the Jaccard Coefficient yields a much more reliable measure.

3 Experiments

This section describes experiments with classical data mining techniques to predict citation counts and most cited/successful authors.

3.1 Citation Count Prediction

Classification attempts were made to predict the number of citations from certain keywords. Attribute selection by entropy, correlation and inverse document frequency as described in Section 2.1 yielded about 200 highly informative keywords. As the correlation coefficient indicates that all keywords are very distinct, a further reduction from this basis is unlikely. For this number of dimension, not enough training instances exist to properly span the hypothesis space.

3.1.1 Extended TF-IDF

A different method to select keywords was implemented. As basis it uses the co-occurrence matrix discusses in Section 4.1. A co-occurrence table is created using all keywords and a binned representation of citation counts. The rows corresponding with citation counts are selected. Again, a co-occurrence matrix is created among the resulting data, yielding distinct keywords for each classification class.

The extracted keywords were thus obtained by creating an extended TF-IDF measure. First, each keyword is assigned a TF-IDF weight w_k, and each goal class receives a corresponding w_g. w_k describes the global significance of the keyword, i.e., whether it has a good coverage of the dataset. w_g describes the significance of the goal class. As there is just one goal class per article, the TF term is constant, but the IDF term describes the size of the goal class in ratio to the number of articles. This is particularly useful, as it will give smaller classes a higher weight, which might otherwise be underrepresented. The keywords are

now weighted with respect to coverage of the whole dataset, but not yet distinguishing different goal classes.
From this weight matrix, the rows describing the keywords with correspondence to the goal classes are extracted, and the process is repeated. Again, all keywords and goal classes are weighted using TF-IDF. The TF term this time is the weight obtained from the previous run. The IDF term is calculated with respect to the number of different goal classes. The final weight for any keyword for a goal class can thus be described as:

$$KeywordWeight_{k,g} = TF_{k_{articles}} * IDF_{k_{articles}} * IDF_{k_{goal\_classes}} * IDF_{g_{articles}}$$

Which is equal to the keyword significance, scaled by how significant it is for the certain goal class, scaled again by the significance of the goal class itself. During matrix multiplication, many terms cancel out, leaving this relatively simple formula.

The extended TF-IDF should select ideal keyword candidates, as it votes down keywords that occur in multiple classes, keywords that occur infrequently or keywords that solely contribute to the majority class. This heuristic method to obtain good keywords is also significantly faster than repeatedly adding a keyword and calculating the information gain or the correlation with respect to all the previous keywords.

3.1.2 Goal class binning

The citation count for an article forms the goal class. Being a natural number, one could use a regression classifier to learn the outcome. However, the output class is distributed very unevenly, as shown by Figure 1. Also, an increase in citations by a small portion might be of significance for a low citation count article, but not to a high citation count article, i.e., the error function is not independent of the distribution. To overcome problems with the distribution, the goal class was binned.

As the data was heavily weighted towards very low citation counts, but many extreme outliers had to be preserved, an automatic clustering of the citation count failed. Instead, classes were chosen manually:

NONE Articles with no citations

ONE Articles with one citation

FEW Articles with more than one, but less than 10 citations

SOME Articles with more than 10, but below 100 citations

EXTREME Articles with more than 100 citations

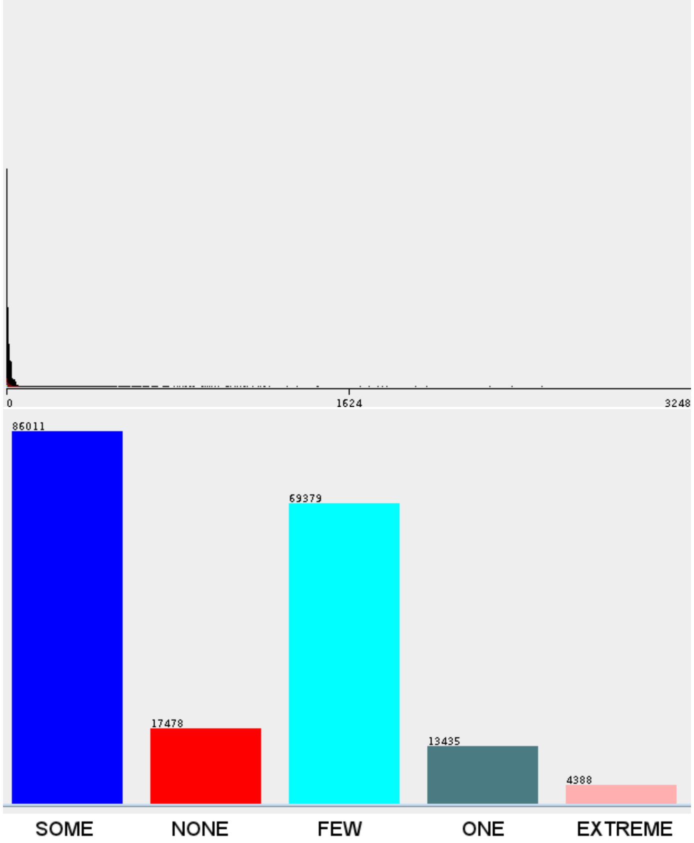

Figure 1: Top: Distribution of citation count. Bottom: Citation count binned into distinct classes.

These ordinal values are very intuitive and exponentially increasing in size, to counter the diminishing number of articles with higher citation count. Not all output classes contain the same number of instances, but weighting smaller bins accordingly can equalize the impact of each class.

3.2 Most cited Author

How often a paper was cited is an important measure of quality. However, if authors collaborate, it is often impossible to find out whom to contribute the effort. While related work [4][16] is usually done on citation graphs, this method is not applicable to this problem. However, all co-authors of each paper are known, and this section describes two methods for ranking authors based on the available information. Rankings for the top 10 authors can be found in Appendix B.

3.2.1 Naive approach

The most naive approach to compare authors is to sum over all work they published, each time dividing the citation count by the number of authors. This way, the naive assumption of equal contribution by each author is made. Using this measure, Charles M. Lieber is found to be the highest ranked author, which is supported by Lieber being recognized as the world's leading chemist in 2011 [15].

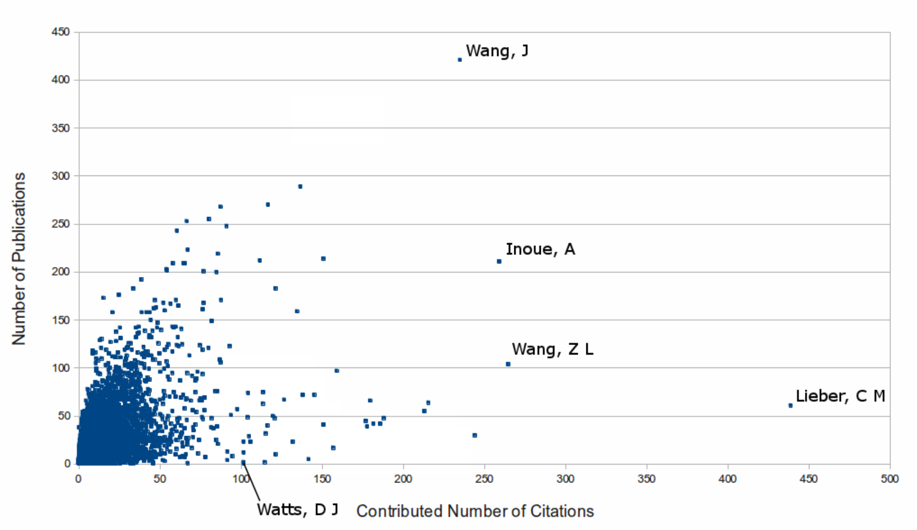

Figure 2: Naive citation count attribution versus publication count

Figure 2 shows the contributed citations versus the publication count for each author. Note that Wang, J. is listed with over 400 publications, which is a clear outlier, and could indicate multiple authors collapsing into one, due to the short notation of names. Another interesting data point is Watts, D J, reaching a citation count of over 100 from a single publication.

3.2.2 Relative Author Rank

The naive approach not only divides the number of citations equally among all authors, it also does not take into account the relative difference in citation counts for multiple authors. For example, if authors A and B collaborate in a publication, and A has a top-ranking, single-published paper, but B published only work that was not cited so far, it is to be assumed that A will have a higher contribution to the citation count of the co-published paper than B.

In order to factor out these dependencies and have a single rank variable, the publications were represented as a linear problem, with a matrix A holding binary values that note which authors worked on a certain article, a vector b

Matrix A	Author 1	Author2	…
Article 1	1	0	…
Article 2	0	1	…
Article 3	1	1	…
…	…	…	…

×

Vector x
Author Weight 1
Author Weight 2
…

=

Vector b	100	2	90	…

Figure 3: The cooperation matrix A multiplied with a relative author weight vector x yields the citation counts for the articles b. Solving for x, the authors can be ranked according to their weight.

holding all citation counts for each article and then solving the equation $A{*}x = b$ for x. Figure 3 visualizes the circumstances. x holds then a relative weight for each authors, how much - compared to his co-workers - he contributed to the overall citation count. Figure 4 shows the distribution of positive and negative contribution for all authors. In the very right, Watts, D J can be found again, meaning that he contributed more to the paper published with his name than his colleague; according to this model at least. Charles M. Lieber can be found in the far right as well, but his rank using this measure lowered to 787. This is probably due to the fact that he worked together with many top-rated scientists, who published highly ranked papers themselves. Only strongly contributing authors can reach rank 787 of 270890.

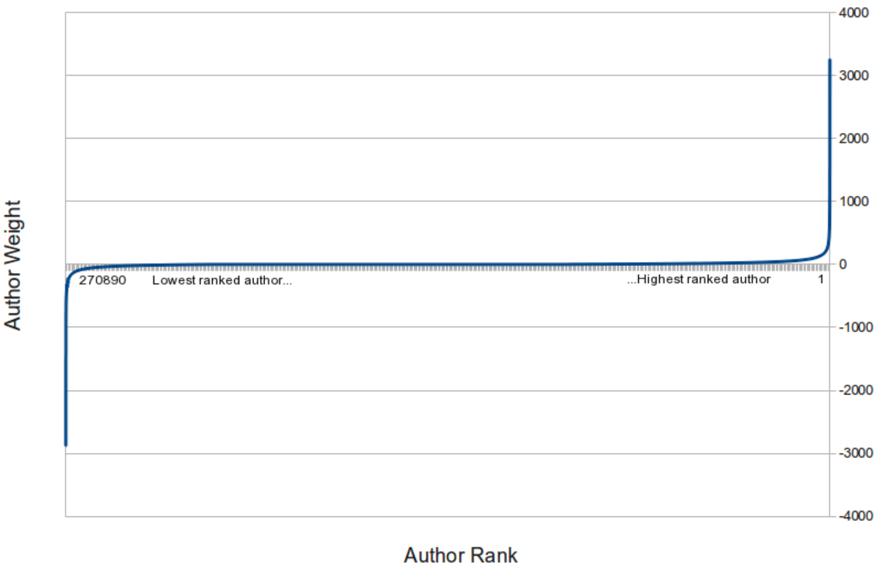

Figure 4: Author rank versus relative author contribution (author weight)

4 Exploratory Search

On an unknown dataset, an exploratory search is performed to reveal hidden structures or information that was formerly not known or well defined. Text- and data-mining techniques can help replenishing results when performing keyword searches.

4.1 Automatic Taxonomy Derivation

A taxonomy is an arrangement scheme for a multitude of entries, grouping similar entries together. Various clustering methods can be used to build a taxonomy, but a simple and straightforward approach is to create a co-occurrence matrix of all keywords [18]. This resulting matrix lists, for each keyword, how often it co-occurred with any other keyword. Using TF-IDF to weight keywords by significance punishes common words, while placing a high emphasis on low-frequent words, that occur together comparatively often. From this matrix, one can select a row for a certain keyword, then read off different keywords that might provide deeper insight.

4.2 Automatic Ontology Learning

TextToOnto was used to generate an ontology directly from the title sentences and the abstracts. While term extraction and POS tagging was performed with reasonable quality, the special nature of short, cryptic title and abstract sentences made relation extraction infeasible. Either no association rules could be learned due to too low support, or the quality of the generated rules was quite bad, even to non-experts.

4.3 Keyword search

To assist in the process of finding answers to specific questions such as which materials might be used instead of rare-earth elements, a search engine was set up. Keywords entered by the user are matched with the closest stored keyword; edit distance is used to provide some tolerance concerning misspellings. Now, instead of simply returning articles with the highest term frequency count with respect to the entered search words, the search engine performs a semantic search by exploiting the taxonomy matrix generated. This is a commonly used query expansion strategy [8][19]. The rows for the search words are selected and subsequently multiplied piecewise with another, yielding a co-occurrence weight for each keyword with respect to all search words simultaneously. This expands the search query to also return documents similar to the ones returned by the simple word query. As most of the costly calculations are performed once during cooccurrence matrix generation, the search process is fairly quick. A search comparing each document description vector to a search vector will need much more calculation time during search.

4.4 Frequent Itemset Mining

Frequent itemsets are another way to describe co-occurrences. Opposed to the cooccurrence matrix, this approach focuses on the single connections between two instances. Apriori was used to extract perfect confidence, high-support data pairs. An example might be frequently co-occurring countries, as a way to inspect and map global relationships and cooperation patterns within the field of nanotechnology.

4.5 Clustering

A clustering task is a grouping of data records based on one or more criteria. Being the most common unsupervised learning method, it aims generally at minimizing intra-cluster distances, while maximizing inter-cluster distances. A good example of such a task is a clustering of countries based on keyword usage. Inside of these clusters, observations can now be made. In this example, countries that are close to each other in terms of keyword distance might be close to each other in a geographical sense as well.

A variation of the standard clustering is a hierarchal clustering. This type of clustering visualizes at which distance threshold clusters would merge. In theory, this means data can be divided into any amount of clusters, by choosing a single threshold.

Distance measure Clustering relies quite heavily on the way distance is measured. If countries are clustered by their keyword usage, then the choice of keywords and the way distance is calculated can greatly affect the outcome. Another challenge is the data distribution. When clustering with respect to countries, the heavy skew towards the USA, Japan and Germany overshadows any other results.

The most common way of measuring distance between instances is the cosine similarity. This similarity can also be used to measure cluster coherence. By grouping the articles by country and then measuring the cosine similarity between them we can generate a dendogram displaying how closely these countries match each other.

5 Results

Classification using bag-of-words representation of keywords to predict for example the countries of publication proved to be ineffective in both quality and calculation overhead. Heuristic measures of keyword importance like TF-IDF and Shannon entropy were used.

5.1 Citation Count Prediction

Predicting the citation count of an article based on only keywords proved to be a challenging task. For a wider variety of classifiers, the numeric output was binned into the ordinal values described in Section 3.1.2 Using these classes as prediction goal, classifiers could be trained with an accuracy of about 70%, and averaged ROC area of 0.72. A ROC Curve and a Precision/Recall chart for a single class can be found in Appendix C.

Classification Difficulties Using a classifier in many hundred thousand dimensions is a complicated task by itself. Dimensional reduction has to be applied on beforehand, however, these techniques are commonly applied heuristically, that is, an assumption about the data and the relevance of certain dimension is made. For example, very common words of the English language are usually removed from keyword lists, as they carry few information.
The dilemma arising from many dimensions is referred to as the curse of dimensionality. In the specific case, even if reducing the description of the dataset to about 200 dimensions, using the techniques as described in Section 2.1, a binary classifier would still need about 2^{200} distinct examples (two for a separation in each dimension), which is a lot more than occurring in the data set. During dimensional reduction, cross-correlation measures and entropy value showed that with 200 keywords, every keyword was independent from the others. Further dimensional reduction would inevitably introduce a classification bias to the system, which then lowers precision on previously unseen data.

5.2 Substitutes for Rare-Earth materials

Using the search engine described in Section 4.3, "substitutes for rare earth materials" can be searched. Returned is a list of articles that feature the search terms frequently, but that also feature words frequently used together with the search words. The result list is sorted by the co-occurrence of the search terms; articles, that feature all search words immediately will be placed on top, ranked by term frequency. Below, all "fuzzy matches" are ranked by the weight of the connection - the co-occurrence strength - and their term frequency. This search term expansion ensures that always articles are returned, even if not all search terms occur in it.
As an example for functioning search term expansion, consider the search terms "graphene", a carbon material recently leading to a physic's Nobel prize, and "STM", short for scanning tunneling microscope. These search terms do not frequently co-occur, however, the search returns results found by "scanning", "tunneling" and "microscope", which are co-occurring with both initial search terms.
Searching the available data for substitutes for rare earth elements or replacements of them did not yield very meaningful results. An explanation may be that the initial search terms were too far from another, spanning a suboptimal result space, or that simply no article in the set uses these or related words in the title or abstract. Both searches and their search vectors and top result are also presented in Appendix D.

5.3 Visualization

Many different techniques exists to visualize high-dimensional networks. For showing data clustering results, dendograms were chosen, also, graphs created by Gephi are used.

5.3.1 Clustering

The most common way to represent a hierarchal clustering is by using a dendogram. The x-axis represents the different instances to be clustered. On the y-axis distance is represented. Different instances will be merged at different heights on the tree. The lower the merge, the more closely these instances are related. At the top of the dendogram is the distance at which all instances are in one big cluster.

Each country was described by a set its 300 most significant keywords, obtained from the taxonomy matrix. Using these weighted keywords, the cosine distance can be used to calculate a distance matrix. From this matrix again, any type of clustering can be achieved. The difficulty is showing the clustering in such a way that it is visualized without cluttering the screen with unreadable labels and without sacrificing too much information. For a dendogram there is simply a limit on the amount of instances before losing the overview.

An example of such a dendogram is shown in Figure 5. The results for clustering the countries by keyword usage intuitively makes sense. Poland is, geographically and in the clustering, close to Russia, Western European Countries form a cluster together. The People's Republic of China is clustered closely with Japan. In conclusion, it seems that geographically close countries have a tendency to write about common topics.

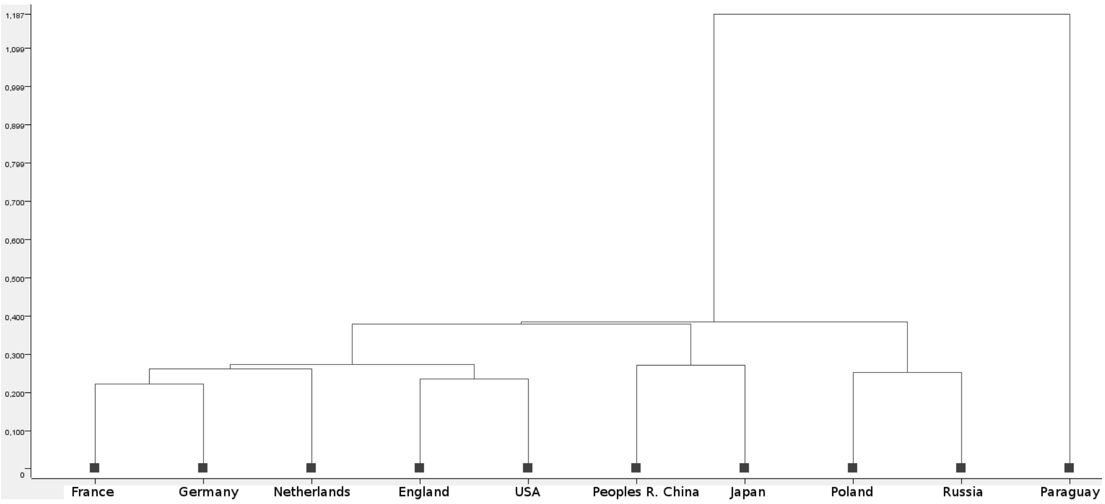

Figure 5: Clustering of countries by keywords, the Y axis represents cosine distance

In order to deal with the dendogram's main limitation (clarity is lost when trying to visualize many countries) Gephi provides a solution. Countries can be given a weight according to their publication count, and then with a given distance matrix Gephi can visualize more countries before losing oversight. A graph visualization of the most important and distinct countries can be found in Figure 6.

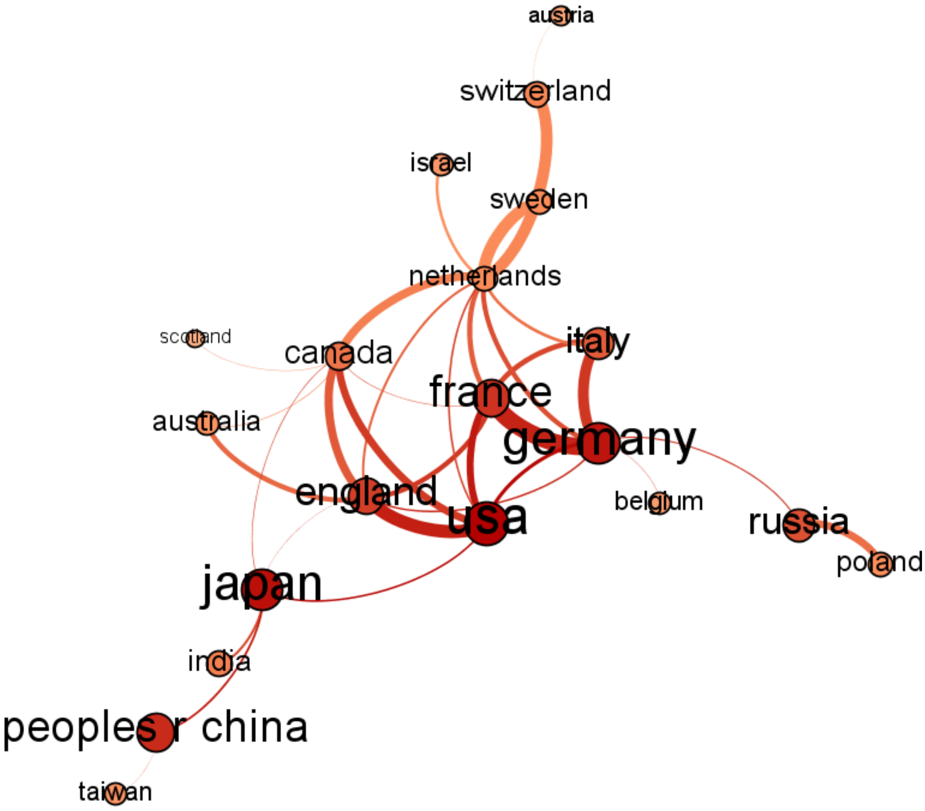

Figure 6: Clustering of countries by keywords. Note the strong connections also implied by Figure 5

6 Conclusion

Many interesting facts can be gained by using exploratory search methods. To be able to effectively use data mining techniques with textual data, many specific textual preprocessing steps should be done. In order to perform classification or prediction tasks on unfamiliar data, specific questions should be formulated and answered, as using prediction classifiers for exploration is highly imperformant. Domain knowledge is particularly helpful for choosing or developing tools that properly understand the text specifics.

6.1 Citation Count Prediction

The best model created for citation count prediction was a J48 decision tree, yielding a ROC area of 0.72 and an accuracy of about 70%. This hints towards a weak relationship between certain keywords and citation count. Between 10 to 200 keywords, selected by different measures, were used as prediction basis for classifiers like NaiveBayes and decision tables.
Using authors or citation counts alongside the keywords may improve classification prediction, however, the general usage of such a classifier is limited. Also, a classifier that tells people what they already know, e.g., positively weighting the age of an article, as it had more time to be cited, is of no real interest.

The approach presented by this paper, using only keywords for selection purposes, might be improved by a more extensive and language-dependent preprocessing. Chemical elements and complex physical expressions may not alway be properly stemmed or normalized by commonly used text processing tools. Comparable work done in different fields [13] uses much more specific and detailed information that is not readily available in the nanotechnology corpus, or citation graphs, [6], which are not accessible as well. Yan et al. [20] use keywords together with author information, which limits the general usage of the model. Concluding, the usage of only keywords for citation count prediction poses a nontrivial task, and the resulting classification performance may be best possible from the given input.

6.2 Exploratory Search

Exploring the depths of any larger dataset will always show some interesting connections. Especially for non-experts, the results achieved are of impressive quality, however, domain experts should guide the search and rate intermediate results.

Creating a taxonomy from a co-occurrence map was a useful step for all further tools in the workflow. Keyword relations, country cooperation and unique fields of research for authors could be found using the taxonomy matrix. To non-experts, the results seem to be conclusive, also, many findings are properly backed up by human investigation.

Automatic Ontology Learning Future work is necessary to find out if the field of nanotechnology is simply too complex to be understood by current association relation learners, or if including the full text of the articles will improve ontology generation. However, given the current article collection, it is best to let a field expert design an ontology, then let a program enrich it, as opposed to fully automatic learning.

Substitutes for rare-earth materials The search engine built is able to find articles described by given search words, and also articles that match with keywords commonly used together with the search words. As the search is done on titles, keywords and abstracts, articles that include substituting rare earth materials should be found and returned. However, as the best matches cover different fields of study, it may be concluded that no articles explicitly talking about the subject are present. A non-exhaustive human search supports these findings.

References

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI conference on weblogs and social media*, volume 2. AAAI Press Menlo Park, CA, 2009.

[2] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. *KNIME: The Konstanz information miner*. Springer, 2008.

[3] Christian Borgelt. Apriori - Association Rule Induction / Frequent Item Set Mining. Retrieved June 16, 2013, from http://www.borgelt.net/apriori.html, 2013.

[4] Joseph K Bradley, Patrick Gage Kelley, and Aaron Roth. Author identification from citations. Technical report, Tech. Rep., Dec. 03 2008, 2008.

[5] Beman Dawes, David Abrahams, and Rene et al. Rivera. Boost C++ libraries. Retrieved June 16, 2013, http://boost.org, 2013.

[6] Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the 24th international conference on Machine learning*, pages 233–240. ACM, 2007.

[7] Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer & Ian H. Witten. WEKA - A Machine Learning Workbench for Data Mining. Retrieved June 16, 2013, from http://www.cs.waikato.ac.nz/˜ihw/papers/04-EF-etal-DataminingWEKA.pdf, 2004.

[8] Gaihua Fu, Christopher B Jones, and Alia I Abdelmoty. Ontology-based spatial query expansion in information retrieval. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, pages 1466–1482. Springer, 2005.

[9] Rapid-I GmbH. RapidMiner. Retrieved June 16, 2013, from http://rapid-i.com/content/view/181/190/lang,en/, 2013.

[10] Magdalena Graczyk, Tadeusz Lasota, and Bogdan Trawiński. Comparative analysis of premises valuation models using KEEL, RapidMiner, and WEKA. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, pages 800–812. Springer, 2009.

[11] Mark Hall. Semi-automatische Ontologieerstellung mittels Text-ToOnto. *Retrieved June 16, 2013, from http://wwwu.edu.uni-klu.ac.at/mhall/data/courses/se_cl/se_cl_semi_automatic_texttoonto.pdf*, 2004.

[12] Jiang-Liang Hou and Chuan-An Chan. A document content extraction model using keyword correlation analysis. *International Journal of Electronic Business Management*, 1(1):54–62, 2003.

[13] Cynthia Lokker, K McKibbon, R James McKinlay, Nancy L Wilczynski, and R Brian Haynes. Prediction of citation counts for clinical articles at two years using data available within three weeks of publication: retrospective cohort study. *BMJ*, 336(7645):655–657, 2008.

[14] Kishore Papineni. Why inverse document frequency? In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[15] Thomson Reuters. Essential Science Indicators. Retrieved June 16, 2013, from http://archive.sciencewatch.com/dr/sci/misc/Top100Chemists2000-10/, 2011.

[16] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 121–128. IEEE, 2011.

[17] Kilian Thiel. The KNIME Text Processing Plugin. Retrieved June 16, 2013, from http://tech.knime.org/files/KNIME-TextProcessing-HowTo.pdf, 2009.

[18] Chia-jung Tsui, Ping Wang, Kenneth R Fleischmann, Asad B Sayeed, and Amy Weinberg. Building an IT taxonomy with co-occurrence analysis, hierarchical clustering, and multidimensional scaling. *Proceedings of iConference*, pages 247–256, 2010.

[19] Jiewen Wu, Ihab Ilyas, and Grant Weddell. A study of ontology-based query expansion. Technical report, Technical report CS-2011-04, University of Waterloo, 2011.

[20] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. Citation count prediction: Learning to estimate future citations for literature. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1247–1252. ACM, 2011.

A Selected Keywords

A.1 Year specific keywords

These keywords were selected using the extended TF-IDF described in Section 3.1.1. All keywords existing in a year are ranked, thus, keywords may appear in multiple years, if their TF-IDF with respect to all articles is quite high. The keywords are sorted by significance, but only the top 10 keywords are displayed here.

1998 quantum, nitrophenoxide, nitrophenoxides, postlabeling, spectroellipsometry, electron, molecular, films, populina, thioethyl

1999 quantum, electron, molecular, films, carboxymyoglobin, melamines, insomnia, milky, quinolinecarboxamide, goodpasture

2000 quantum, hospitalized, azoospermia, electron, hagfish, molecular, films, hydrogenphosphate, optical, monocationic

2001 quantum, nanofillers, electron, films, molecular, enantioseparation, belousov, codeine, mixings, properties

2002 quantum, cannabinoids, nanodosimetry, films, electron, mgcni, molecular, nanocoatings, bifeo, subsystems

2003 wms, microhotplates, itac, sooting, hreaction, lignites, lifted, recrossing, premixed, flame

Note that the 2003 keywords are significantly different from the others. This is most likely due to the comparatively low number of publications in 2003 in the dataset, 24 publications versus an average of 38133 publications in the other years.

A.2 Country specific keywords

These keywords were selected using the extended TF-IDF described in Section 3.1.1 with respect to countries. For readability reasons, only the 7 selected countries with their top 10 keywords are listed here.

Tajikstan atmosphere, stabilized, benzene, aluminum, preparation, nanoparticles, hydrogen

Liberia spectinomycin, ofloxacin, tetracycline, meningitidis, sulfonamide, gonorrhoeae, neisseria, determinant, africa, antimicrobial

USA molecular, quantum, films, properties, structure, electron, thin, surface, spectroscopy, phase

Germany quantum, films, properties, molecular, structure, electron, spectroscopy, surface, thin, phase

Japan films, quantum, molecular, properties, structure, electron, thin, surface, spectroscopy, phase

Croatia dichlorostyrene, ylamine, staurolite, lamarck, benzothia, ococh, tmgn, quinodiimines, methoxypyridine, silacyclopropabenzenes

Thailand portunidae, molecular, nevirapine, quantum, dipyridodiazepinone, washes, thmp, gih, intrasplenic, ulpro

One can easily see that countries with low publication count feature completely different keywords. Tajikstan and Liberia both just feature one publication, and thus the keywords extracted from it make them very distinguishable. For some countries with higher publication count, more similar keywords are extracted. The highest ranked keywords for USA and Germany are the same, as their initial TF-IDF values are very high as well. This means, keywords like "molecular" and "quantum" are significant keywords for the whole set, and as Germany and USA cover a high percentage of the set, these keywords are ranked highly.

A.3 Author specific keywords

A selection of authors with keywords most associated to them.

Shasha, D E nanomunchers

Gourley, P L nanolasers

de Veiga, L A hydroxybenzaldehyde

Paixo, J A hydroxybenzaldehyde

Lau, G drown, murdered

Lieber, C M polynucleosomes, remodeled, incredible, researchers, nanotransistors, kilobase, haplotyping, warping, nanotweezers, swi

Watts, D J synchronization, spread, oscillators, chaos, collective, pulse, networks, coupled, small, disease

Many authors just have few keywors uniquely identified with them. These are authors with few publications, in this case, all other keywords occurring in their publications are not significant enough, so their initial TF-IDF value is below the threshold. In further processing these keywords are skipped.
L A de Veiga and J A Paixo are identified through the same common keyword, so it can be suspected that these two authors heavily cloaborated. Indeed, both authorsjust have a single publication, which they co-published.
Note that these keywords are not the most significant fields teh authors worked in. For example, selecting keywords using TF-IDF for authors yields "carbon, nanotubes" for C M Lieber. However, as these keywords are very common among authors, the re-selection with IDF with respect to the other authors (thus forming the extended TF-IDF) yields "polynucleosomes, remodeled", which are far less common keywords among authors, thus better spreading the set.

B Top 10 Authors

Using the two different ranking strategies discussed in Section 3.2, the top 10 authors are listed in Table 1.

Rank	Absolutely Weighted	Relatively Weighted
1	Lieber, C M	Fire, A
2	Wang, Z L	Watts, D J
3	Inoue, A	Joubert, D
4	Dekker, C	Su, J Z
5	Wang, J	Gal, S
6	Caruso, F	Weber, E
7	Mirkin, C A	Maccagnani, P
8	Alivisatos, A P	Moscatelli, F
9	Xia, Y N	Moronne, M
10	Dai, H J	Chan, W C W

Table 1: The top 10 authors ranked by naive ranking and by relative author rank.

C Citation count Prediction

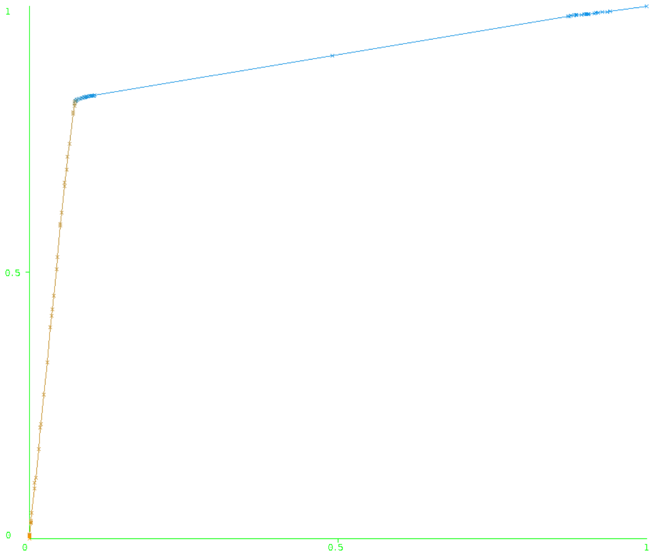

Figure 7: A ROC curve obtained from a J48 decision table for the class "FEW". The AUC is 0.87, total accuracy of this classifier was about 70%. X: False positive rate, Y: True positive rate.

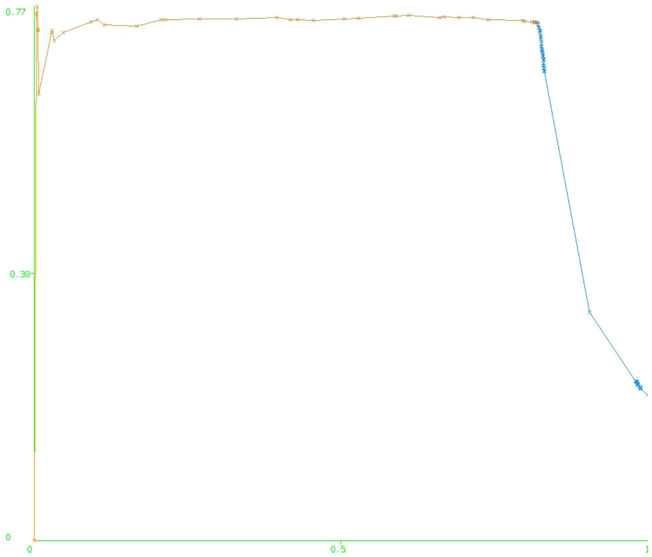

Figure 8: A Precision/Recall chart obtained using a J48 decision table for the class "FEW". Total accuracy of this classifier was about 70%. X: Recall, Y: Precision.

D Search Engine Results

Besides the actual search results, the search engine returns the search vector taken into consideration during search. This information can be used to estimate the quality of the search results, and also shows the capabilities of search query expansion. This section of the appendix will show and compare a successful search and a less successful one.

Search for "Graphene STM" A search vector for a single term is basically the excerpt from the taxonomy matrix. The search vector for graphene, for example, is:

graphene (1) tubules (0.5) carbon (0.37) nanotubes (0.36) electronic (0.29) microtubules (0.25)

Whereas the search vector for "STM" is the following:

stm (1) tunneling (0.5) scanning (0.47) microscopy (0.40) surface (0.31) surfaces (0.20)

Only the 6 most important terms are shown, of course, the actual vector includes many more elements. During search, the engine combines the vectors by multiplication. The result is a vector that includes all frequently co-occurring terms to both search entries:

electronic (0.01) carbon (0.01) structure (0.01) microscopy (0.01) tunneling (0.01) scanning (0.01)

As neither "STM" nor "graphene" are included in the combined search vector, one can easily see that apparently thsoe terms do not frequently co-occur.

However, the extended form, "scanning tunneling microscope" can be found in the search vector. This shows, that indeed a common semantic ground between both search terms is found, and thus the search results will be good. The highest matching article for this search term is:
"Atomic structure of carbon nanotubes from scanning tunneling microscopy"

Search for "Rare Earth Element Substitute" In contrast to the rather successful search before, the combined search vector for many different combinations tried to find substitutes, alternatives or replacements for rare earth elements, commonly looked like the following:

molecular (0.001) structure (0.0002) films (0.0001) properties (7.2e-05) complexes (4.9e-05) quantum (2.8e-05)

Besides the weights being extremely low, indicating a low resemblance with the original search terms, one can easily recognize these terms being the most common significant keywords, i.e., the most general result the search can return. This strongly hints towards the original search terms not frequently cooccurring, and thus that no relevant article with all occurring keywords is in the set. The best matching article is:
"Catalytic mechanism of dihydrofolate reductase enzyme. A combined quantum-mechanical/molecular-mechanical characterization of transition state structure for the hydride transfer step"

E Clustering Results

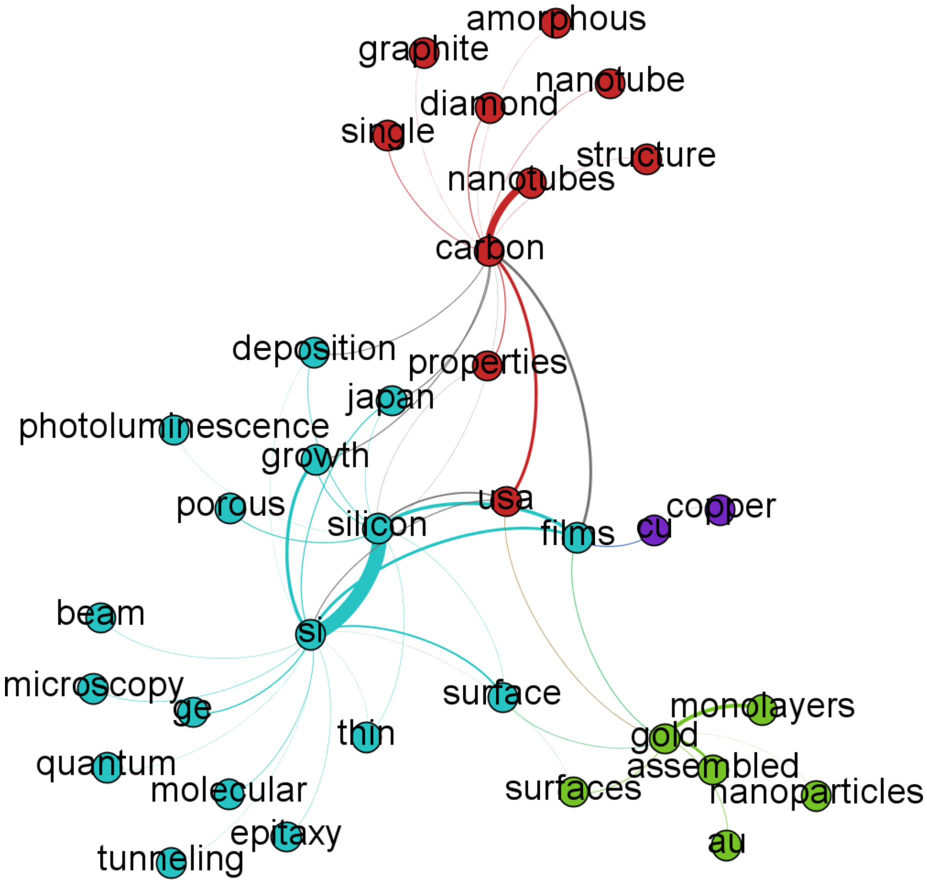

Figure 9: A graph visualization of elements and their most distinctive keywords. Element connection strength is determined by cosine distance between the vectors of 300 most significant keywords.

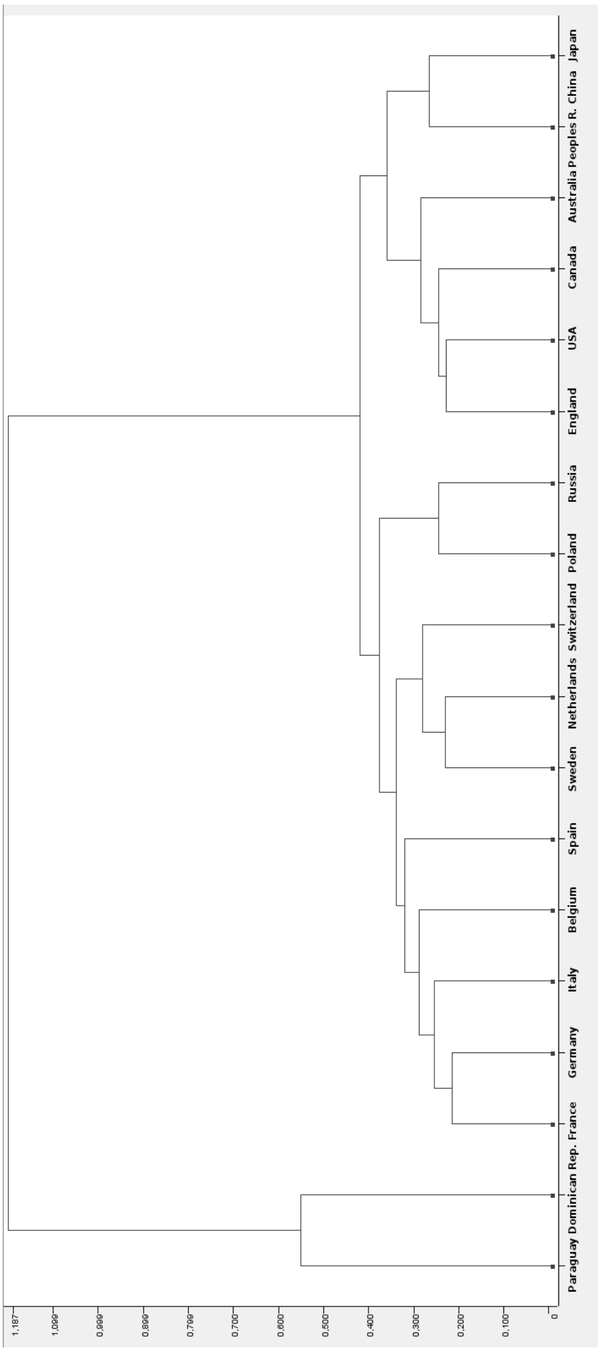

Figure 10: A dendogram including some selected countries. Distance is total link based on cosine distance between the 300 most significant keyword frequencies. Note that geographically close countries are clustered together.