DISSERTATION
for the degree of
DOCTOR OF PHILOSOPHY
in
PHYSICS

Predicting centers of evolving knowledge networks

ÁDÁM SZÁNTÓ-VÁRNAGY

Ph.D. School of Physics
Head: Prof. Tamás Tél

Statistical Physics, Biological Physics,
and Physics of Quantum Systems Program
Head: Prof. Jenő Kürti

Advisor: Illés J. Farkas D.Sc.

Department of Biological Physics
Eötvös University
Budapest, Hungary
2019

Contents

Introduction

In our era of informational revolution, massive data sets are collected, and procedures to process them are especially looked after. The theoretical underpinnings of this communal effort were developed as a common contribution of scientists coming from different fields. Statistical physics has contributed its unique approach and experience regarding measuring the behavior with complex systems.

In the present work, four common questions of this area are to be analyzed, often using a number of measures jointly. The central idea behind is to gain a better understanding about these large data sets, find their elements of outstanding importance ("centers"), and being able to trace them in a visual manner, which, as experience shows, is the most appropriate way for human agents.

A distinctive feature of the following experiments is the general input format of the algorithms presented, which makes it flexible to apply under other circumstances, as well. The foundation of these methods are topics, words, which are mentioned often in the title of publications, articles, and their evolution in time. Often, an analysis of the network structure accompanies the evaluation.

Another emphasized point of the present research is the practicality and usability even on large inputs. Not every common and well-known algorithm possesses the scalability necessary for this purpose. Indeed, in the case of first two research projects presented, the input is limited to the smallest data set being used (which is also relatively large, compared to other possible data sources, with its 400K records and 4.7M links). In all four cases, the source code of the methods being presented are publicly available (at `http://topinav.elte.hu/`).

Chapter 1

Basic concepts

In this chapter, the following common elements of the whole research process will be presented: the data format, content and quantity (Sections 1.1-1.2), the basic procedures and methods that are shared by the different projects (Sections 1.3-1.4), and a short summary and comparison of the upcoming chapters (Section 1.5).

1.1 Data sets

We used five data sets for our study, scaling from 400K to 16M records. The data sets are consisting of records containing a title and a date. Regarding the latter, some of the data sets use year, others use month as unit of resolution. For the sake of simplicity, from now on we are going to mention only years. Three of these data sets are scientific (*APS*, *Web of Science* and *US Patents*, the latter downloaded from Google Patents, see [1]), one is a news data set (*Zeit.de*), and one of them is somewhat of a transition between these two types (questions asked on *Stack Overflow*). Refer to Table 1.1 and Figure 1.1 for their further description in terms of numbers.

1.2 Data fields

Every data set consists of a list of *titles*. The *words* of the titles are extracted afterwards. There is a *date* for every title: the year is used in most cases (except the Stack Overflow data set, where the month is used). Between the titles there are *links* running, which are considered to be the connections of a network of these titles.

Table 1.1: *Size and duration of the data sets.* A word is considered significant if it reached a relative frequency of 0.1% among all records of a year in at least 1 year (see also Figure 5.1). The choice of the used date interval was based on the data quality: the beginning of some data sets are often lacking, the last year is usually not complete. For the meaning of neighbor limit, see Section 5.2.2.

Name of data set	Years	Records (millions)	Links (millions)
American Physical Society (APS)	1965-2009	0.4	4.7
Stack Overflow (SO)	2008-2015	10.7	0.76
US Patents	1976-2012	4.8	109
Web of Science (WoS)	1991-2011	35	392
Zeit.de	1995-2014	0.9	0.19

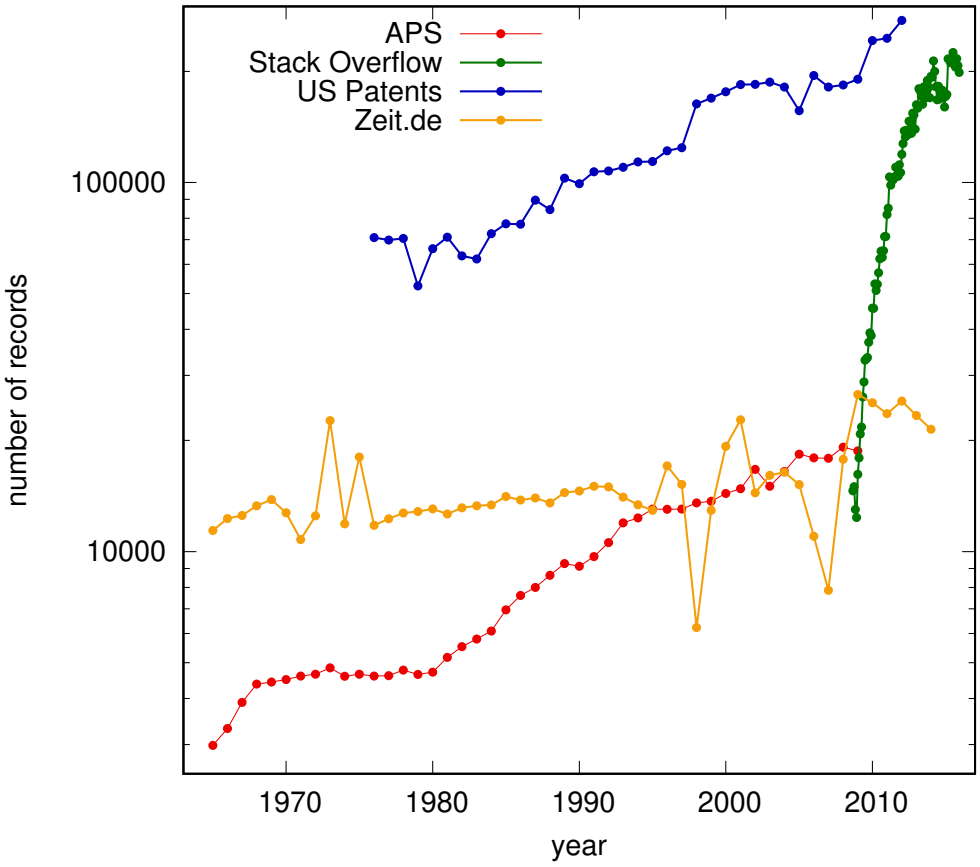


Figure 1.1: *Number of articles in the data sets.* All of them follow a quite consistent, increasing pattern, during the years examined, except for Zeit.de which experiences major jumps. This is attributed to a failure in obtaining the data, and this fact is taken into consideration during processing the data.

In summary, this results in three main input types:

1. words,
2. links,
3. dates.

Before processing the data set, the set of *relevant words* are determined. This is crucial, since typically the frequency of the words usually follow a power law-distribution, therefore, there is a significant amount of rare words, which unnecessarily increases computational complexity. Much computing time can be therefore spared by restricting the procedure to the set of the relevant words exclusively.

1.2.1 Stop words

In order to achieve this, first, the stop words are removed. A *stop word list* is a list of words which are excluded altogether from the investigation. Their importance lies in their frequency, since typically they consist of the most often used words. Such lists are publicly available, assembled by linguists, and include very general terms such as *in, if, or,* etc. In the specific case of the APS data set, some further words like *phys* or *rev* were added to the list of stop words, since APS has journals abbreviated as *Phys. Rev. A, Phys. Rev. B, Phys. Rev. Lett.*, etc., which occur common but do not specify a topic.

1.2.2 Frequent words

Then a list of frequency is made out of all occurring words. This list is then also filtered, based on the following rule: a word is considered relevant if it occurs at least k times in a timeframe (year or month, depending on the specification of the data set) in average. The value k is typically between 1 and 5, and is determined manually, on a case-by-case basis.

For example, the data set APS contains publications in the timeframe 1965-2009, altogether 45 years. Based on this, using $k = 3$, a word in this data set is relevant if it occurs at least $3 * 45 = 135$ times in the whole timeframe.

1.3 Words as topics

Extracting topics from documents is a field of active research (see [2] for the original approach, [3], [4] for later innovations that consider time-evolution of topics). However, in order to be able to apply these algorithms one needs extensive data. Our research uses very restricted input, consisting of years and titles only, that is flexible even in more extreme cases, and turns out to be effective even by the most simplistic methods, by considering all words occurring in titles as separate topics.

The topics are represented by a keyword which occurs in a title of a record in a data set. No topic model is applied to unify the similar keywords (see [2] for a classical example for a topic model, which assigns topics to each document in a set).

For example, if there is a title like *"Axial-Vector Vertex in Spinor Electrodynamics"*, then it will be regarded as pertaining to the topic keyword *electrodynamics* (among others). That is, simply all titles were processed word-by-word and every word was considered to be a "topic". A side-effect of this simplicity is that, for example, the word *vertex* also will be a topic and this may cause confusion, if it is such a term that is shared by several real-world topics. For example, the title *"Vertex-coloring edge-weightings: towards the 1-2-3-conjecture"* pertains to a quite different topic than the other publication mentioned. Notwithstanding, this type of error is not so common that it would prevent the prediction algorithm from producing effective results.

1.4 Topic diagrams / time series

Once we have topics with a list of articles assigned to them, it becomes very easy to follow their rise (and fall). One of the most important tool in this research is what to be called as follows as *topic diagram* (also often called "time series"). For a source of a large data set of time series, referred by 850 publications, see [5] (citation data from [6]).

The topic diagram is constructed by processing all titles and corresponding years in the data set. By traversing the data set once, it is possible to save the data for every topic. One title is processed by extracting all the words it contains (except stop words) and increasing the count for every word, in the given year. At the end of this process we have the count of occurrences of every word, grouped by years.

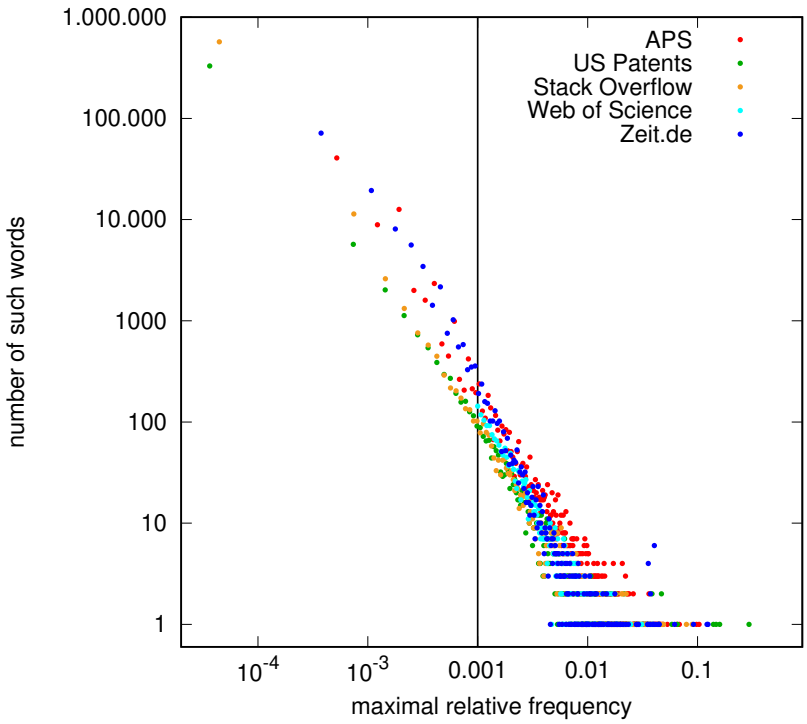


Figure 1.2: *Cutoff of relevant words based on global normalization.* The vertical black line at 0.001 is the cutoff value used in Chapter 5. It marks the border between significant and excluded words. The relatively large white area below the points, right from the border, show that significant words indeed make up a large proportion of all words.

1.4.1 Normalization

The raw data of a topic diagram consists of the number of occurrences of a certain word, in a certain timeframe. For practical purposes, we will apply various types of normalization, in order to make the data more useful and comparable. A value that is not normalized is referred to as *frequency*, while a normalized value is called *relative frequency*, which can use one of the following ways of normalization:

1. **Local normalization** uses the frequency values of the current topic diagram exclusively.
   (a) **Min-max:** fits the lowest value of the topic diagram to 0 and the highest to 1. This has a side effect to increase the noise, which one has to bear in mind.
   (b) **Sum:** every frequency value is divided by the sum of all frequencies values, which has the result that the area below the whole curve (*sum*, or integral) will be 1, in a very similar fashion to probability density functions.
2. **Global normalization** includes information besides the current topic diagram as well. The number of titles containing the selected word divided by the number of all publications in the specified year. Consequently, this normalization produces a very small number. It is useful for determining cutoff level for frequent words (see Figure 1.2, Section 1.2.2).

Often, the two approaches has to be combined in order to take advantage of the benefits of both ways. The importance of the local normalization is that global normalization in and of itself is producing very low numbers, and also not easy to compare more frequent words with less frequent one, since they are on different scales.

On the other hand, local normalization only would result very similar time diagrams, especially for frequent words, because they would inevitably reflect the general trends of the whole data set (by the law of large numbers). Therefore, one can get the most result by combining these two methods.

1.4.2 Representation of a topic

It is often extremely convenient to treat topic diagrams as codes, consisting of coordinates which correspond to subsequent years (or year-tuples). This code consists of the coordinates $a_i, i = y_1, \ldots, y_n$ (where $y_1, \ldots, y_n$ are the years), defined as:

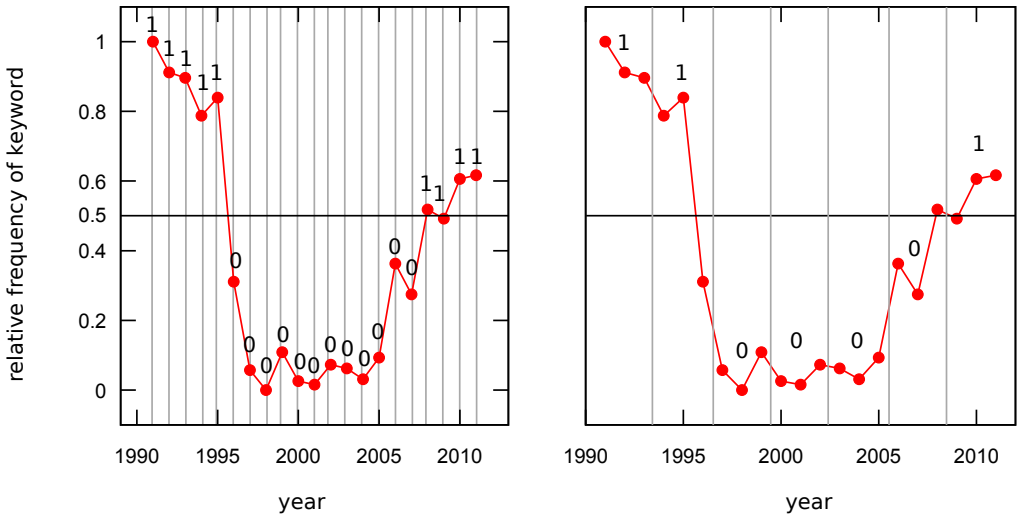


Figure 1.3: *Example of clustering by coordinates.* The two plots show the same values but with different subdivisions. On the left plot every year is considered in and of itself ($l = 1$), on the right plot the average of every three year ($l = 3$) is used. Such a block is coded with 0 if the value is below 0.5, which is the average of the maximal and minimal frequency value, 1 if it is above 0.5. The left plot results the code 11111000000000000001111, the right results 1100001.

$$a_i = \left\lfloor \frac{freq_w(i)}{h} \right\rfloor \tag{1.1}$$

An example of such a calculation can be seen on Figure 1.3, for $h = 2$ (stands for *height*). This means that the Y axis is divided into 2 subdivisions, therefore the coordinates can take 2 values (0 or 1). By increasing this subdivision by modifying h, one might be able to get more accurate results, which can catch more of the behavior of the topic diagrams.

The second parameter l (stands for *length*) is found in the definition of $freq_w(i)$, which is as follows:

$$freq_{w,l}(i) = \frac{1}{l} \sum_{j=il}^{il+l-1} freq_w(j) \tag{1.2}$$

which is in effect grouping the years by groups of size l. Afterwards, $freq_{w,l}(i)$ can be used in Equation (1.1) instead of $freq_w(i)$ (which is really the $l = 1$ special case of $freq_{w,l}(i)$). The two ways of calculating the coordinates are demonstrated on Figure 1.3.

1.5 Record-topic interactions

The rest of the current work is going to analyze the interactions between records and topics, using the above definitions.

In Chapter 2, record-record interactions are examined, with regards to how one publication can encourage (*boost*) the citation of the other, by referencing it.

In Chapter 3, record-topic interactions are examined, essentially from the same perspective, examining the positive effect (*burst*) of a publication on a whole topic.

In Chapter 4, topic-topic interactions are examined, with regards to 4 different *similarity measures*, by comparing their theoretical effectivity.

In Chapter 5, single topics are examined, without interactions, by comparing their first and second half of their timeframe, for the purpose of *prediction*, by means of a simple classification.

Chapter 2

Boosts: the dynamic interactions between articles

2.1 Motivation: challenges in measuring science

As a result of the immense growth of scientific research in the recent decades (see Figure 2.1), objective evaluation of scientific contributions became a major challenge and important focus of research. It is no longer possible to follow the advancements of a field in the same way as it was yesteryear. Computer-aided tools are necessary in order to choose what to read. Scientific data is available in large from several sources (arXiv, APS, Google Scholar, MathSciNet, PubMed, Scopus, Web of Science). Measurements can be evaluated in terms of:

1. **Article.** The most common option is the *number of citing articles*, which is the in-degree in the citation network, where the nodes are corresponding to the publication, the links are the citations.

2. **Author.** The most common option is the *h-index*, the biggest possible number h for an author who has at least h publications with at least h citations.

3. **Journal.** The most common option is the *Impact Factor*, which is (roughly speaking) the average in-degree for the articles published in the specified journal for the last two years. More specifically, it works by considering all the incoming citations in the last two years and dividing it by all the publications issued in the last two years.

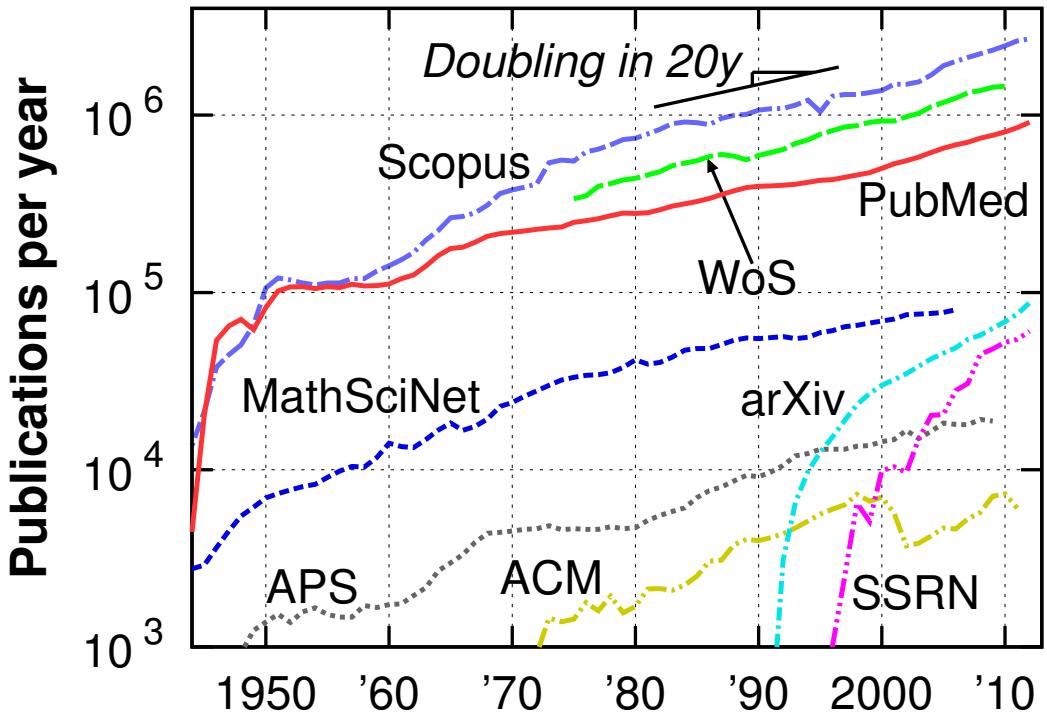


Figure 2.1: *Growth of science between 1950 and 2010.* An analysis based on 8 different data sets, as appeared in [7] (data processed by Adam Szanto-Varnagy and Illes J Farkas, figure assembled by Illes J Farkas).

With the help of this immense amount of data, these tools can be used to help us move to a higher scale of information.

We turn our attention to more complex measurements than simply a publication with high citation count. The purpose is though somewhat similar: to have an overview to the most important events in science, throughout a larger timeframe. To this we could apply the citation count, but, despite its simplicity it has several drawbacks:

1. **Discipline bias.** It is problematic to compare articles coming from different fields of study with the same measure of citation count, since it depends on the overall publication number in the specified field (see [7]). It would be unfair to compare authors based on any citation count-based metrics in a funding context unless they come from the same science field.

2. **Negative effect.** The citation count does not distinguish between those citations that use the quoted article as a background and base for their research and those that

criticize it. The mere fact that an article accumulated a high number of mentions does not imply that they were equally for the good.

3. **Importance of citing article.** The very idea of considering the weight of the citing article was the one to start a revolution in web search engines (i.e., Google PageRank), and it is no less applicable to scientific publications.

Beside these issues, manipulation is also often mentioned (see [8] for example), but it is not listed because it is not specific for any kind of measure. Whatever method is developed for a measurement, human creativity will eventually find the way to manipulate the procedure.

We will present two types of statistics: one is using interactions between two articles, the other between an article and its pertaining topic.

2.2 Definition of a boost

A possible method to broaden the perspective is to focus on links instead of elements. In the case of scientific publications it is the connection between two publications. Such a connection is measurable in terms of the effect of the citing article on the cited one. We will refer to this effect hereinafter as *boosting*. I introduced this notion in order to be able to detect larger interacting groups of articles and detect articles that have a widespread influence in terms of its effect or the number of its effected citations.

A typical example of boosting can be seen on Figure 2.2. Erdős and Rényi published their article about classic random graph models in 1959 ([9]). It became later relevant again by the introduction of the scale-free model by Barabási and Albert (1999, [10]). The number of articles citing the ER model throughout the years makes it apparent that most of the citations are coming on behalf of the later BA-paper.

Generally speaking, the ER-paper is the *boosted*, the BA-paper is the *boosting* one. A formal description and procedure can be made as follows: in order to find the boosting element for a selected, boosted article (or in order to test whether an effect is indeed present or not), we consider its *children* (that is, those articles that are citing the boosted one). Then we collect all the *grandchildren* articles, in order to see, which of the children are the most influential, which might be the boosting. The *boost value* for a selected pair of boosted-boosting article is defined as the proportion of their common children (that is,

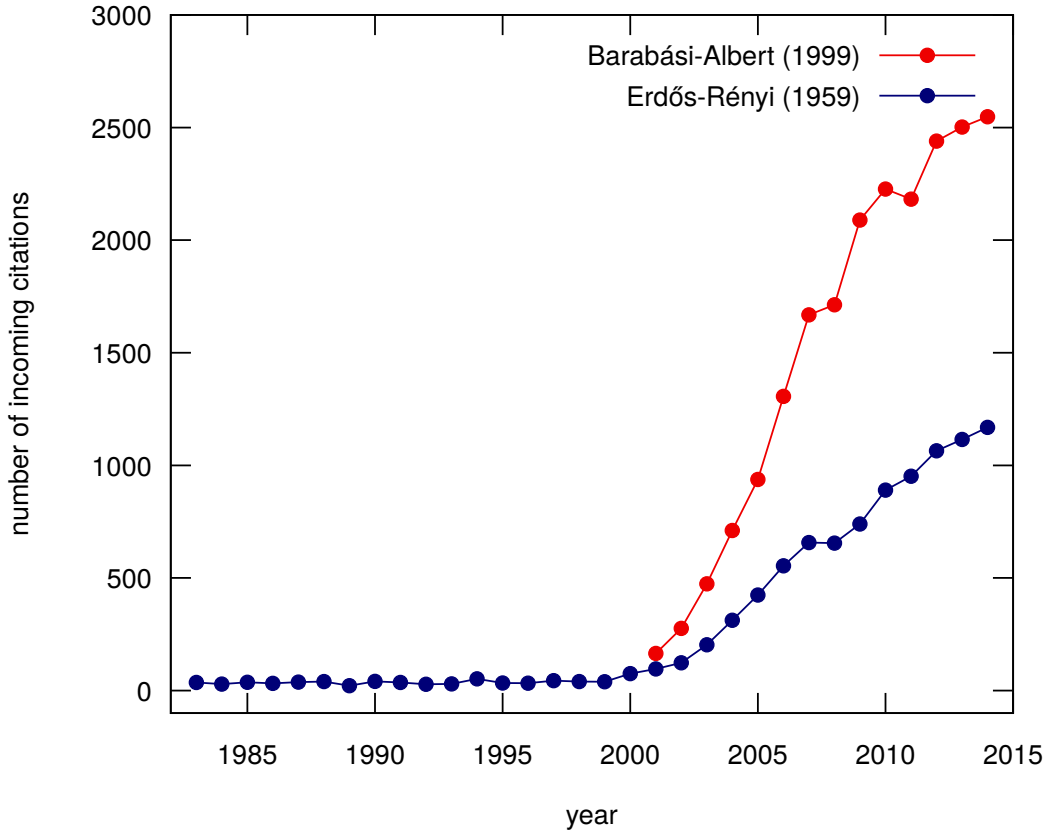


Figure 2.2: *Yearly citations of two correlated articles.* Their parallel histories suggest a cause-and-effect relation. After introducing the scale-free graph model, an increased attention was palpable regarding the original random model. The effect could be further measured by comparing their list of citation and the size of intersection (which in this specific case was not possible because of a lack of data).

the size of intersection between the grandchildren on behalf of the boosting article and children of the boosted article) and the children of the boosted article. Formally,

$$b(b_2 \xrightarrow{\text{cites}} b_1) = \frac{|c(b_1) \cap c(b_2)|}{|c(b_1)|},$$

where b_1 is the boosted article, b_2 is the boosting article, and $c(b)$ is the set of children of article b. This boosting value is calculated for all children, and the one with the maximal value is selected as the boosting value for the boosted article.

2.2.1 Implementation details

The computations were run on the APS data set. Because of its size, it is appropriate for testing such experimental methods, since this algorithm involves the processing the grandchildren of every record, it has a complexity of roughly $\mathcal{O}(n^3)$.

In order for the algorithm to run in a reasonable amount of time it was necessary to optimize two steps: the query of the children of a specific record and the query of the number of the children. After the first is ready, the second is obvious based on that. The complete list of articles and their respective citation counts fit into the memory (11 MBytes). The query of the children was made simple by the use of *sorted grep*, which is a tool to find in sorted files. Given the list of all citations (DOI of citing and cited article), the cited one was put in the first column and the file was ordered. Afterwards, a simple run of `sgrep` returned the children fast. As a result, the whole data set was processed in a few hours, and no sampling was necessary.

2.2.2 The distribution of the boosting value

One would expect that the boosting effect which is clearly explainable in the case of the renaissance of the random graph model is quite uncommon. On Figure 2.3 we see that this intuition is not correct. Indeed, the number of articles are following a decreasing pattern by the growth of the boosting value, but the curve does not have a sudden drop. Even at boosting value $b = 0.9$ there is still the 0.01% of all articles are present (more than 100 records). There are almost 3000 articles (out of 400,000) with boosting value between 0.8 and 0.9.

2.3 Normalizing boost

2.3.1 Dependence on citation count

Since boosting value was defined as a relative proportion, it does not tell us anything about the absolute citation count of the boosted and boosting articles. Therefore one might suspect that the boosting effect occurs by articles of especially small relevance exclusively, since for small numbers it is more probable to get a large proportion of common citations, even by chance (for example, to have 2 common citations out of 3). To test this hypothe-

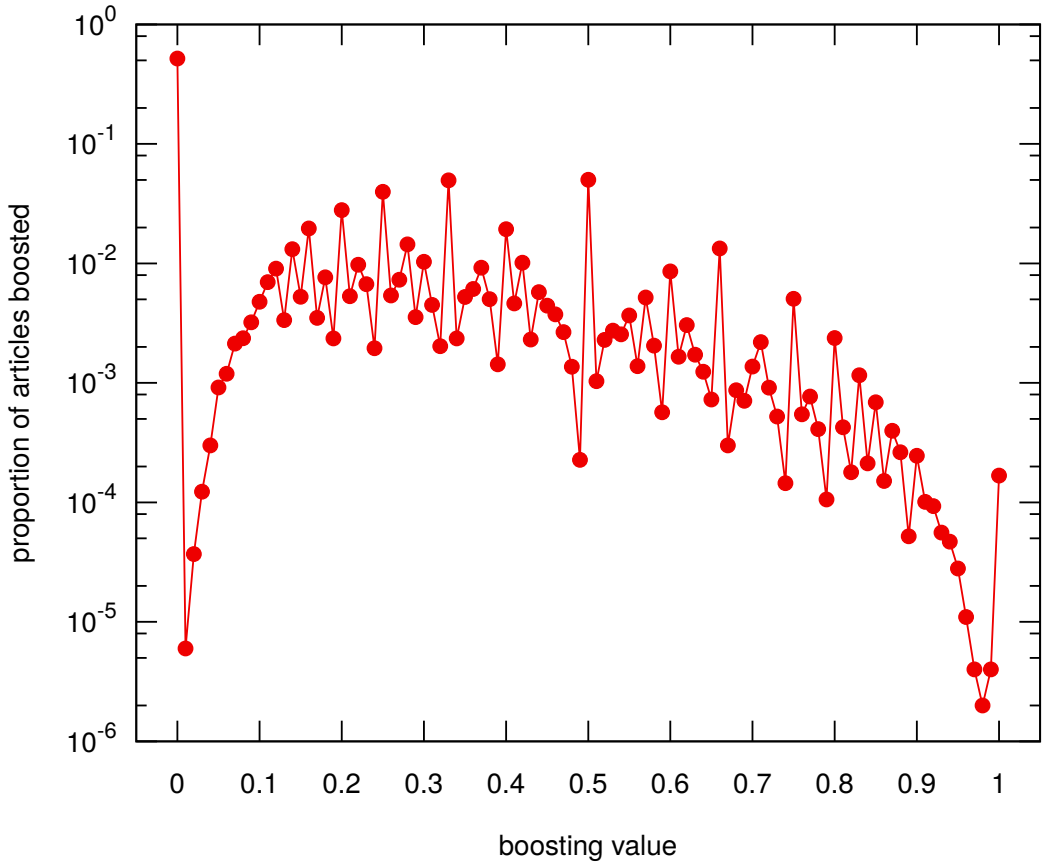


Figure 2.3: *Distribution of the boosting value.* Boosting value $b = 0$ is outstanding because articles without a single citation also belong here, while boosting value 1 collects all such groups of articles that were referred together only. Besides this, the distribution is not smooth because it was calculated as a result of a division operation, which results some often fractions like 1/2, 1/3, 2/3, etc.

sis, let us see the correlation between the citation count and boosting value of all articles of our data set (Figure 2.4).

The vast number of records seemingly concentrate in the middle of the scatter plot, around 100, which is because for articles with citations less than 100 there is only a very limited number of possibilities of what fraction the boosting value can be. These $\frac{p}{q}, 0 \leq p, q \leq 100$ fractions are the points which together form the regular curves on the left part of the plot. Besides this "dense" area there is a natural tendency and correlation which we expected: the higher the citation count, the less its boosting value is. Nevertheless, above this imaginary line we still find records in a large number.

Interestingly, the most outstanding of all of them is almost in the right top of the

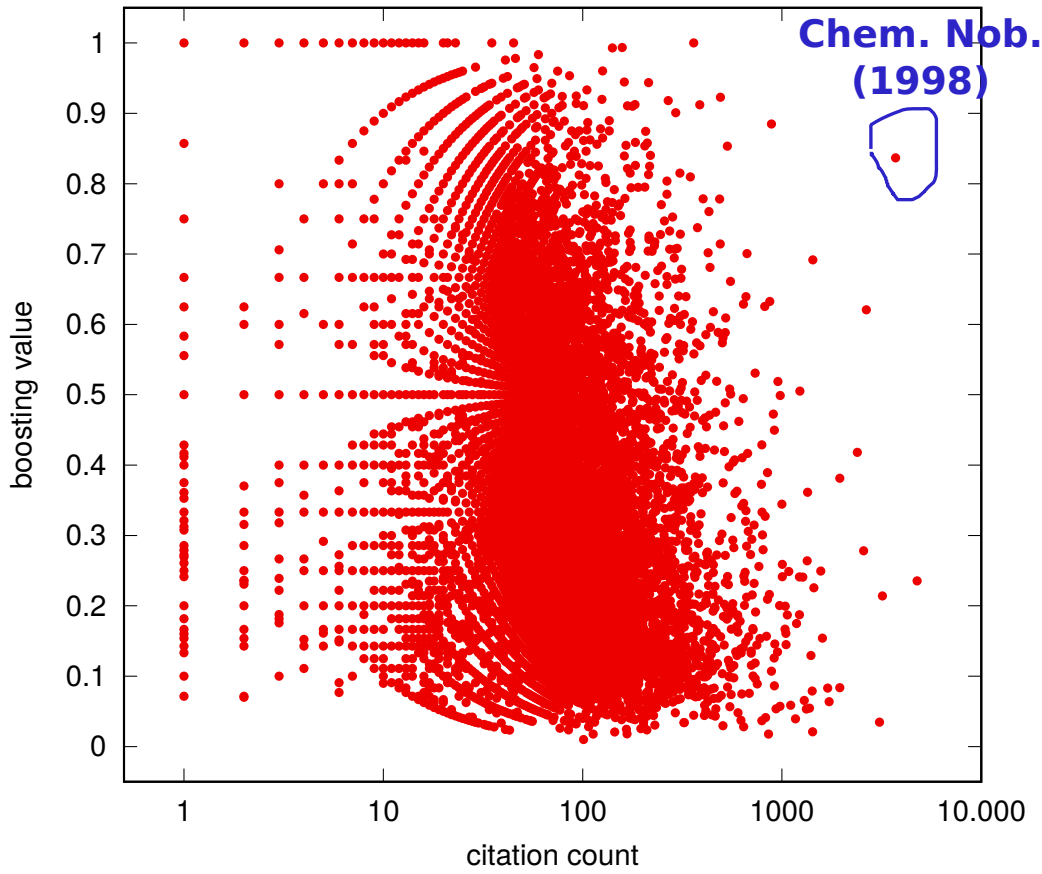


Figure 2.4: *Citation count and boosting value.* The left part of the plot is subject of "fraction bias", since the boosting value was defined as a fraction of two set sizes. This results the regular shapes. The upper right part shows that the records of different boosting value are evenly distributed, that is, there are a number of records which have large boosting and citation number as well.

plot, was actually a paper winning the Nobel Prize in 1998, in chemistry. The boosted article was published in 1964 ([11]), the boosting in 1965 ([12]), and their common author was the Nobel Prize-winning Walter Kohn. The earlier article shares the 83.7% of the citations with the more recent one. This example proves that the definition of the boosting value and its visualization can be helpful in finding and identifying publications of special importance.

2.3.2 Dependence on time duration

In the example of the 1998 Nobel Prize, there is only a year between the two publications. In contrast, our earlier example of graph models went through a duration of 40 years. This directs our attention to another important parameter correlating with the boosting value of an article: the time difference between the boosting and boosted paper. Whenever this time duration is small, we can suspect that the boosting effect is a result of the common topic of the two articles, and the boosted one is not quoted because of the boosting one, rather, they are treated as equal parts of the whole. It follows that we can speak of a genuine boosting effect if we find such pairs with a long time duration, in a large number (possibly occurring together with a high number of quotations).

The relation between time duration and boosting value is being analyzed on Figure 2.5. Just as expected, there is a negative correlation between the two: as the timeframe grows, the boosting value (or, boosting effect) drops. But at the same time, similarly as we have seen by the citation count, the effect does not disappear completely. There is a significant number of boosted-boosting article pairs which both have a big boosting value and quite a number of years passed by between their publications. Notice also that the decrease of the boosting value is very slow, the envelope curve of the scatter plot has quite a small slope. That is, at 5, 10, 15 years there are still present the majority of those elements which produce a boosting effect at all.

2.4 Combining time, boost, and citation

As we have seen, the boosting effect is the most impressive for pairs of articles which

1. have a large number of citations,
2. have a big boosting value, and
3. a large amount of time elapsed between their publications.

These three conditions stand true for our first example of the publications about the graph model. The question is still: is this a single occurrence in history, or are there similar examples, as well? The challenge of answering this question is a technical one: there are three parameters to visualize, while one can see through effectively only two-dimensional plots.

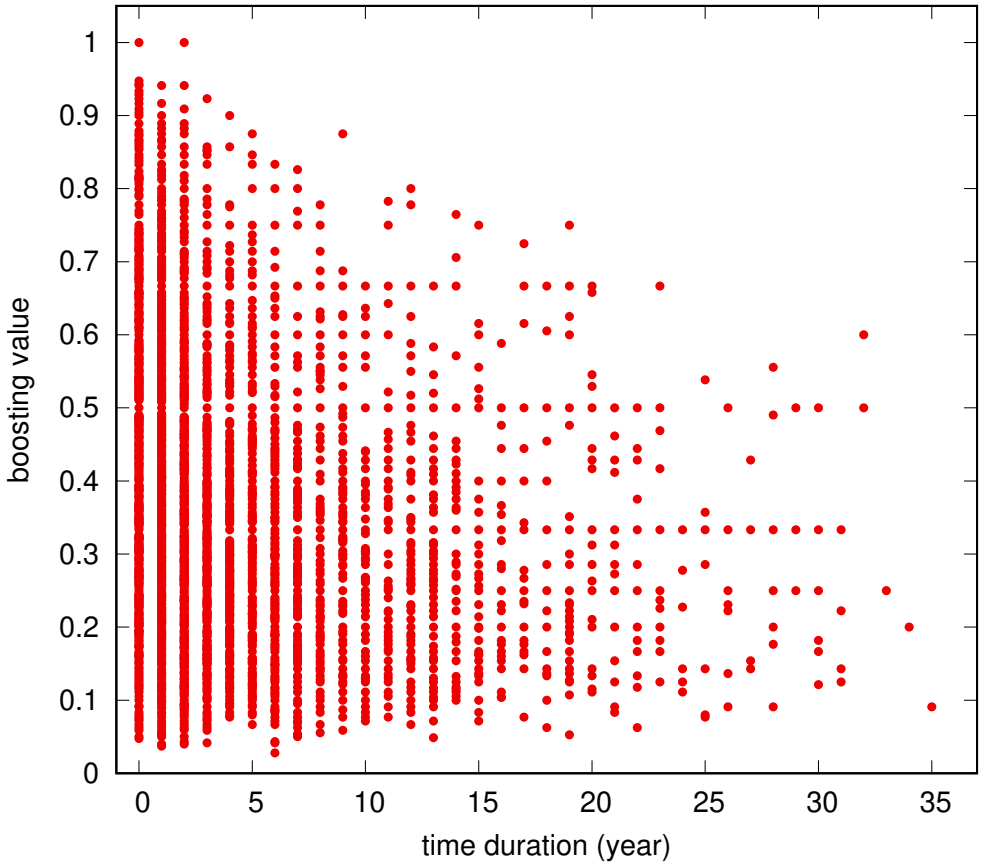


Figure 2.5: *Time duration and boosting value.* The boosting effect can be found not only between articles that are close to each other in time. Although there is such a correlation, since we find a high number of articles with large boosting value in the left side of the plot, nevertheless, their number drops slowly, and there are points with high boosting value even after 10, 20, 30 years.

We can use heatmaps as a solution for this problem, using time duration and citation count as the two axes. We apply two different box coloring: one for the number of records residing in the box (Figure 2.6) and another for the maximal boosting value (Figure 2.7). Those records which fail to reach the maximal value in their box will stay invisible by this method, but experience shows that in the interesting cases we can ignore them. This is because an interesting box has big citation count and time duration, therefore it has a small number of records, amongst which usually no more elements with a large boosting value can be found.

We can use Figure 2.7 for finding specific pairs of articles which behave in a sim-

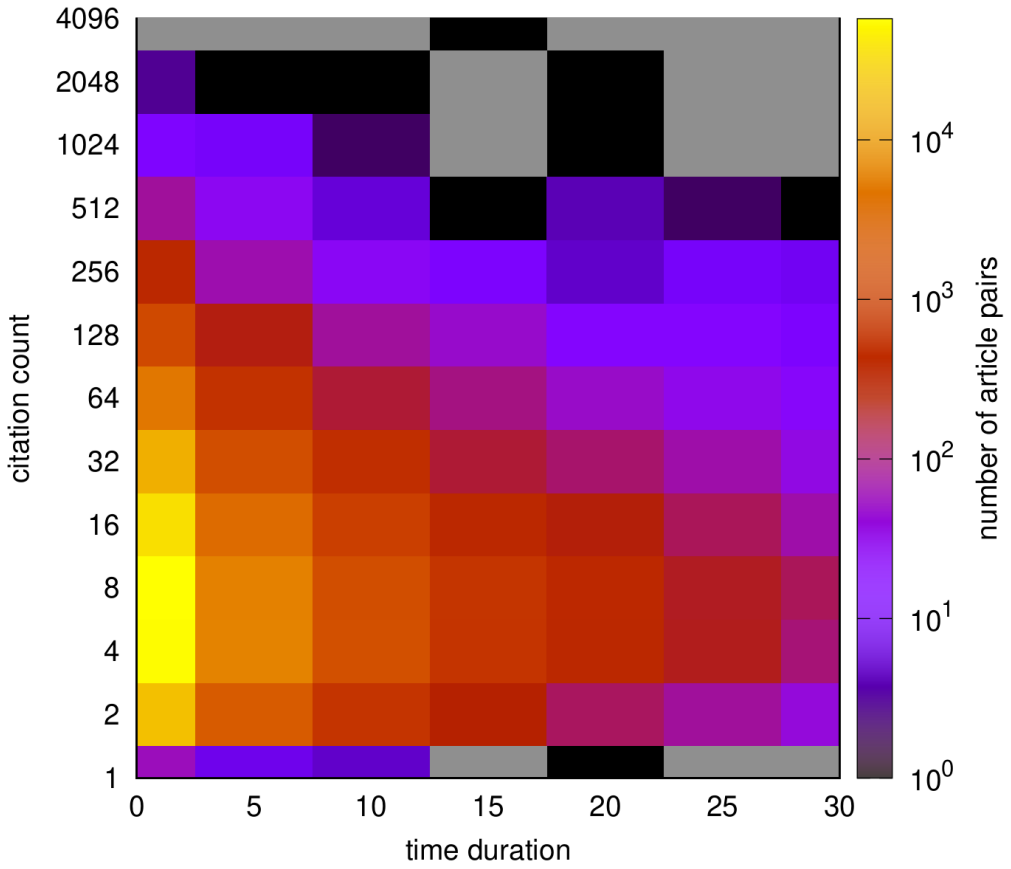


Figure 2.6: *Number of records in citation-duration areas.* This heatmap shows the density of the boosted-boosting article pairs according to the citation count achieved by the boosted one and the time duration between the two articles. The boosting effect mostly occurs on article pairs that are close to each other in time. Note that the prime example presented in Figure 2.2 has approximately 10K citations and 40 year time duration, both of which parameters are way out of the present figure, which indicates that this example is indeed outstanding (at least in terms of the currently analyzed data set).

ilar manner as our prime example about the graph models. The box of the parameters $citation = 256, duration = 15$ is an extreme example on the right top direction of the plot. Its maximal record has a boosting value of 60%. It refers to the article pair of Hertz (1976, [13]) and Millis (1993, [14]) which is known in quantum physics as Hertz-Millis-Moriya theory. (Note that the citation count reflects the state of the APS data set, at the time of the collection in 2009, and counts only citation inside the APS journals. At the time of writing this thesis, on their website, which includes outside citations as well, counts about 1,200 citations for both publications.) The first sentence in the abstract of the

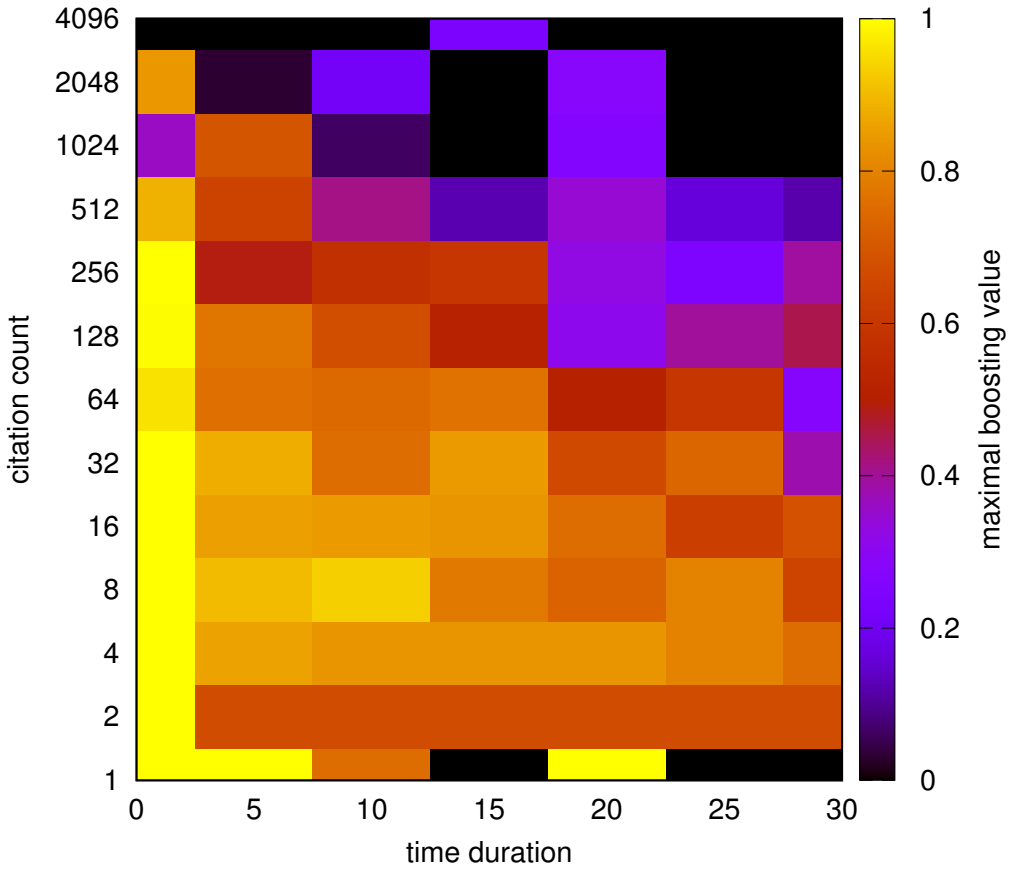


Figure 2.7: *Maximal boosting value in citation-duration areas.* For each box, a single element with maximal boosting value is shown. Its density on the right top area shows that a significant boosting effect occurs even for article pairs which are both having a large citation number and a large duration.

citing article shows clearly that this is indeed a classical case of boosting: "I reexamine the work of Hertz on quantum phase transitions in itinerant fermion systems."

At the same time, the color on this graph, however, is not a guarantee that we will find this effect at every pair. If we examine the maximal element in the box of the parameters $citation = 64, duration = 20$, we find that the publication of Adler (1969, [15]), which is a relatively highly cited one, quotes Steinberger (1949, [16]) in a footnote of a technical computation, mentioning that the source for his assumption is based on private communication as well. This quotation seem to be the cause of more than half of the citation generated by the earlier article of Steinberger.

Based on this heatmap and the presented associated technique, further examples of

```
(64, 20):
0.511 10.1103/PhysRev.76.1180 10.1103/PhysRev.177.2426 90 20
0.379 10.1103/PhysRev.128.2614 10.1103/PhysRevB.32.3792 87 23
0.362 10.1103/PhysRev.155.528 10.1103/RevModPhys.60.209 80 21
0.329 10.1103/PhysRev.121.1093 10.1103/PhysRevLett.47.1913 73 20
0.328 10.1103/PhysRev.184.451 10.1103/PhysRevB.39.6962 67 20
0.319 10.1103/PhysRevLett.50.2066 10.1103/PhysRevC.69.045804 72 21
0.314 10.1103/PhysRevD.22.939 10.1103/PhysRevD.65.025012 86 22
0.311 10.1103/PhysRevB.22.3173 10.1103/PhysRevB.63.144519 74 21
0.292 10.1103/PhysRev.80.797 10.1103/RevModPhys.43.36 65 21
0.284 10.1103/PhysRevLett.35.120 10.1103/PhysRevLett.82.4504 67 24
0.279 10.1103/PhysRevD.14.3260 10.1103/PhysRevLett.85.499 68 24
[...]
```

Figure 2.8: *Boosted-boosting pair database grouped by citation-duration areas.* In this example, a single block is shown which corresponds to a single box in the heatmaps above (see Figures 2.6, 2.6). The coordinates follow the same notion, which helps searching in the database, based on the coordinates found on the heatmap. The database is stored as a plain text file, the fields are separated by a space. The fields are, respectively: boosting value, boosted DOI, boosting DOI, citation count, time duration. Every block is ordered by the boosting value decreasingly.

the boosting effect can be found. In order to make this process easier, a list of all records was assembled, ordered and indexed by the coordinates of the boxes (see Figure 2.8 for an example). This list is a text file of a size 14 MBytes, which comfortably searchable and browsable by hand. Inside the boxes, the records are ordered by their boosting value, in decreasing order. Further fields are displayed: the DOI of the boosted and boosting article, the citation count of the boosted one and the time duration elapsed between the two articles. This view makes it easier to browse the individual cases as well as it served good in the debugging and testing phase.

2.5 Boost network analysis

The subsequent boosts produce a network of the articles. Most of this network is trivial, consisting of tiny components, of about 2-4 nodes. It has nevertheless several bigger components, with highly influential publications in their center. For the sake of analysis, a boosting value threshold of 60% and a citation count threshold of 25 was set. The resulting network consisted of 5,423 nodes and 3,229 edges. On Figure 2.9 its biggest components are shown. On Table 2.1 these components are identified.

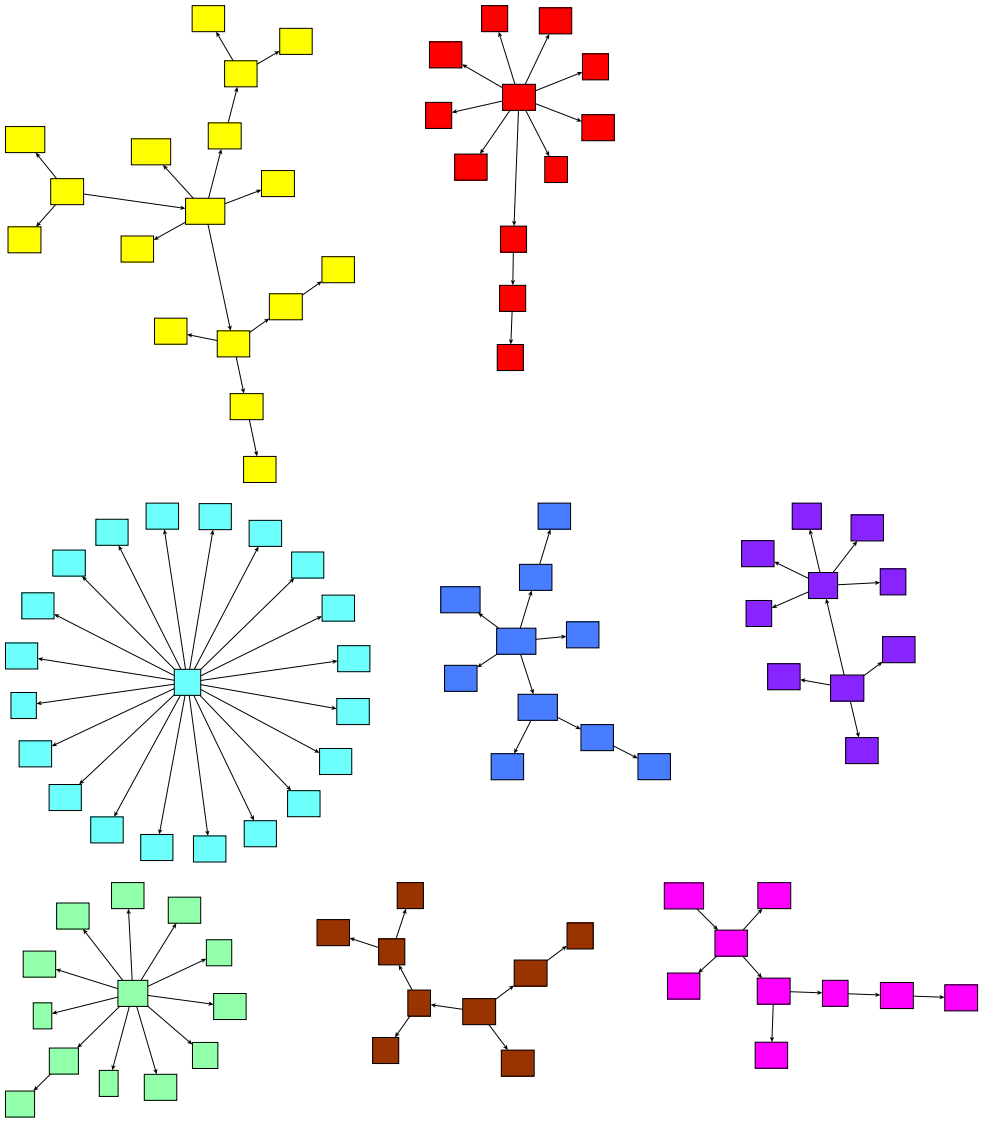


Figure 2.9: *Biggest components of the boost network.* The articles behind the components are given in Table 2.1.

2.6 Conclusion

In this chapter we started to analyze and visualize articles in their context, thereby overcoming the limitations of the classical article-level measures. The first point in our analysis was to measure the strength of a reference by an elementary, set theory-based definition. The basic idea of publications effecting (*boosting*) each other gave us the opportunity to browse the region of publications from a new viewpoint. We introduced navigation

Table 2.1: *Biggest components of the boost network.* As seen on Figure 2.9, in decreasing order of the component size. Refer the bibliography for more details about the central publications. Dominant topics were chosen based on manually choosing most common expression occurring in the title of the articles in the component.

Color	**Component size**	**Central article**	**Dominant topic**
Cyan	21	Barabási (2002, [17])	random networks
Yellow	17	Kostelecky (2001, [18])	Lorentz violation
Green	13	Gammaitoni (1998, [19])	stochastic resonance
Red	12	Colladay (1998, [20])	CPT / Lorentz violation
Purple	10	Nelson (1993, [21])	superconductors
Blue	10	Wunderlich (2005, [22])	Spin-Hall effect
Magenta	9	Harris (2000, [23])	Casimir force
Brown	9	Brhlik (1999, [24])	dipole moment

based on boosting value, citation count and time duration, finally integrating the three approaches into a single database. We established the fact that the boosting effect occurs even in further regions of this parameter space.

Moreover, the definition of the boosting value induced the notion of a boosting network, which contains non-trivial subcomponents. By visualizing it, we successfully identified 8 scientific publications of central importance in their fields, which had a lasting effect on their surroundings. By this, we obtained an alternative evaluation method of article contribution, which can be used either instead of or in conjunction with citation count: its direct effect on its peers.

The source code used for this chapter is available on:
`https://github.com/binyominzeev/boost`.

Chapter 3

Trend-turning publications – identifying bursts

In the previous chapter we saw how individual articles are influencing their peers by boosting them. The articles are able not only to influence other articles, but whole fields of articles as well. In the following chapter, the goal is to find measures that identify this latter type of phenomenon and by means of these measures find outstanding examples of one or more articles boosting a topic.

3.1 The intuition behind bursts

With the help of topic diagrams (see Section 1.4), it is possible to find topics that are growing exceptionally fast. Such an event is called a *burst*. As an example, let us analyze the topic diagrams of the four most common topics in the data set (see Figure 3.1). In the cases of keywords *magnetic* and *model* we can speak about a *localized burst*. The keyword *electron* is decreasing altogether.

The keyword *quantum*, though, has a slow increase. This is not called a burst, even though in the long run such a pattern can reflect more importance than a sudden growth. But since here our aim is to identify individual publications that are responsible for the growth, as described above, the chance for such a result is lower. In the background of such a slow, gradual process usually there is no single, well-defined idea, rather, some deeper, hidden factor might control the events, such as fashion, technical development, governmental funding or other common interest.

Therefore, from such a topic diagram it is not possible to draw a consequence about its initiating articles. This is not the kind of topic that is useful for our purposes, namely, identifying extraordinary effects of individual articles. Hence, in the following we will restrict our interest to localized bursts.

But this restriction is no guarantee for an error-free result of such speculative analysis. It is generally true that by the simple input available at hand it is not possible to have a 100% correct result. The goal is to identify the effect of the publications and all we know about them is those later publications which referred to it. But hidden reasons similar to those just described always exist. Even in a case of a localized burst they might occur, just with a smaller probability.

The judge of the end result, hence, is always supposed to be a human expert. Nevertheless, such a network-based automatized process can act as a good catalyst in discovering hidden, but important parts of the giant data set. This is the most ambitious but still realistic goal that can be established.

3.2 The articles behind the burst

At this point, no formal definition for a burst is given (rather, see later, Section 3.3.1). Based on the examples and the intuition, we start to uncover what is behind these specific burst events. The goal is to identify a specific set of articles that the burst can be contributed to. This is in accordance with what is known in popular science as the *Pareto principle*, or *80/20-rule*, which means that the 20% of the records are responsible for the 80% of the contribution (see Section III. D. in [25]). Later on, we will formulate a precise expression in order to find more similar bursts.

For this purpose, we focus now our attention on all possible articles that is related to the burst and use network science tools in order to identify the most important players.

3.2.1 The subnetwork of the topic

The first step for this is to collect all the articles in the timeframe of the burst, containing the topic keyword. The keyword *"model"* contains 600 articles in the timeframe (1965-1969), the keyword *"magnetic"* contains 2104 articles in the timeframe (1986-1991). Based on Figure 3.1 this might be surprising, since apparently the keyword *"model"* has

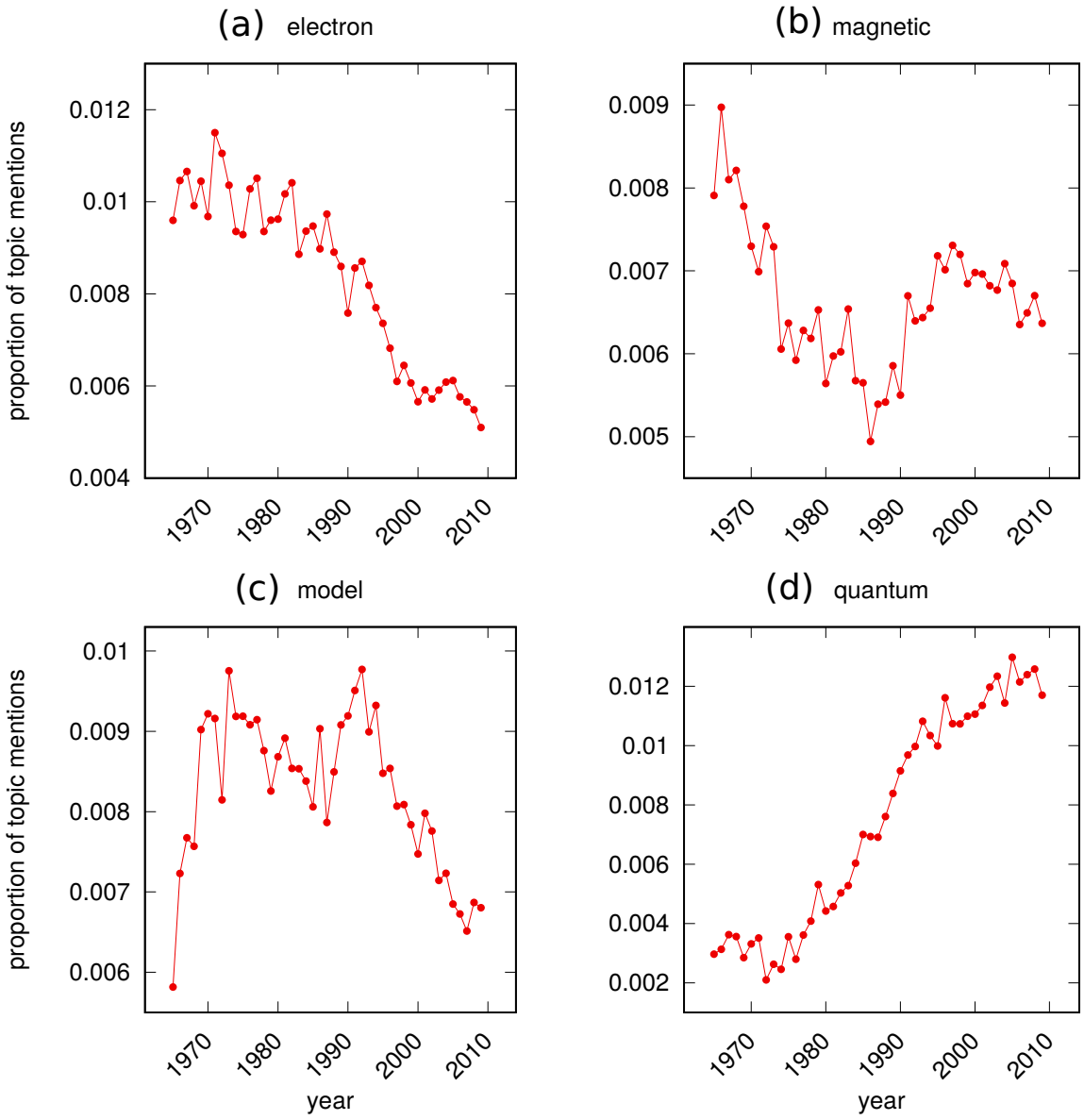


Figure 3.1: *The topic diagrams and bursts of the most prominent words in the APS data set.* The Y axis is the proportion of titles of articles which contain the selected keyword, relative to all articles published in the specified year. Keywords *magnetic* (b) and *model* (c) exhibit a well-distinguishable burst in the years (1986-1991), (1965-1969), respectively. Keyword *quantum* (d) increases gradually throughout the whole examined timeframe, while *electron* (a) does not have a burst, shows a decreasing pattern.

a plot which reaches higher values, but the topic diagram is normalized by the number of articles published yearly. Therefore the relatively smaller success of the keyword *"magnetic"* in the later years yields much more in number of published articles than a relatively

higher success in earlier years.

Among the 600 articles of the keyword *"model"*, 346 has connections, references to other articles of the same keyword. The remaining 254 articles have no connections, they are called *isolated points* in a graph theory-terminology. From now on we are going to try to use information from these connections, therefore, in such a research isolated points make no much use. So we will ignore them. In the network of the keyword *"magnetic"*, there are 721 isolated points. After removing them, a sum of 1383 articles remain.

At this point we have a directed network and we are interested in the effects of every individual article. Every article correspond to a node in this network. A link is going from node A to node B if and only if article A makes a reference to article B. By doing so, we obtain a network, which we will call the *subnetwork* of the topic burst. Later we are going to introduce an algorithm which finds at most one burst for every topic, so we can call it simply the subnetwork of the topic, or *topic subnetwork* (since the burst itself will be unambiguous).

3.2.2 The most influential articles and their citation numbers

Based on this, we have several options of how to evaluate the success, the *penetrance* $p(A)$ of a node A. The simplest is to count the number of nodes $B_1, B_2, B_3, \ldots$ that make a reference to node A. This would result what is called *in-degree* in graph theory or *citation count* in bibliometrics (in a restricted sense, since we now include articles from the specified topic and timeframe only). We can iterate this and include the the nodes $C_1, C_2, C_3, \ldots$, which make a reference to any of the nodes $B_i, i = 1, 2, \ldots$. Or iterate the idea to further levels, even without restricting the number of levels. This type of counting is what is called a BFS (breadth-first search) in graph theory.

This thought experiment goes through all possible fair evaluations of nodes inside the topic subnetwork. From now on we can choose a level threshold l which can be any integer, theoretically speaking, between 1 and infinity. $l = 1$ implies that the nodes are being evaluated by their first-level referencing articles, $l = 2$ means second level, $l = \infty$ means all referencing articles, without a defined limit.

The original goal was to find one or more articles which possibly "caused" all the others. Introducing one of these measures can be indicative of the influential power of the article. Using the limit $l = \infty$ it can occur that for a single article A, $p(A) = N$ reaches the value of the whole subnetwork, that is, the subnetwork consists of exactly $N+1$ nodes

(including the node A itself).

Can we say that such a node A indeed "caused" all the others? That it is the single reason for the whole burst of the topic? No, but as described above, there are always some hidden factors remaining which are not possible to consider working with our simple model. But as a best possible approach, in the following this will be our assumption: that the number of – directly and indirectly – referencing articles is strongly connected to the fact that the article is inducing such a burst.

3.2.3 Layer decomposition and ranks of the articles

Practically speaking, a single article reaching all the others is not the typical case. However, if a small number of articles together does the same, it definitely tells us something about their importance. The problem is that if we have a subnetwork of N articles, then the number of possible subsets which might have this special property is 2^N. So an algorithm finding such a group would be expected to complete in an exponential number of steps, which is practically impossible (often referred to as NP-complete, in computer science). If we impose a restriction on the maximal possible size of this special group, it might decrease the number of the possibilities.

The fact, however, that our subnetwork is directed, makes this whole problem especially easy. One more point is relevant about the scientific citations network: that in most cases it is assumed to be acyclic. That is, it does not contain a directed cycle, which would mean a chain of articles citing each other, and the earliest of them is citing the most recent one. Usually this does not occur, even when it does, it only produces a cycle which has 2-3 nodes only. For the most of the time, it is possible to simply disregard them, as they have really not much bearing on the statistically important end results.

Such a network is called a DAG (*directed acyclic graph*) in graph theory and grants a whole list of advantageous properties. Most of the classic algorithms run much faster on a DAG then a general graph. Even the methods themselves are easier to figure out, even for a beginner programmer, with this assumption.

One of these special properties of a DAG is that it has at least one topological sort, which possible to find in linear time. A *topological sort* is an ordering of the nodes of the graph such that if node A precedes node B according to the order, then edges cannot go from B to A (backwards). In a tree structure, several such orderings are possible (see Figure 3.2). The concept of topological sort can be improved for our purposes into *layer*

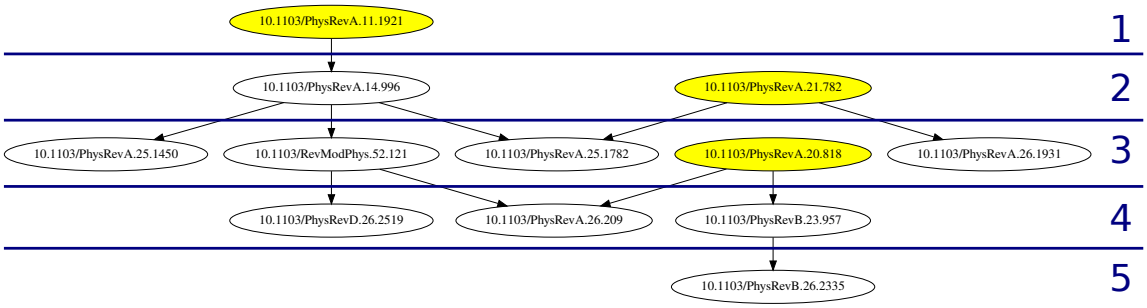


Figure 3.2: *Layer decomposition of a tree.* The figure contains the largest component of the subnetwork for topic keyword *"loss"*. Edge directions are reversed for the graph layout algorithm, Graphviz, [27], in order that it should show the earliest articles on the top, which are cited by others but it does not cite others. Therefore, on this figure – and all following figures – a link is pointing from node A to node B if article B is citing article A. The drawing algorithm uses a random seed for its layout, and might change the permutation of nodes inside a numbered level, but it never changes the number of the level (called *rank*) where to show the articles. The ranks corresponding to levels are shown in dark blue (drawn by the author). A topological sort can be obtained by reading the nodes from top to bottom, by choosing an arbitrary order within the levels.

decomposition, which is unique: with some reasonable restrains, there is one and only one such decomposition for a DAG. For more information about layer decomposition, see [26].

We have to assume that the DAG consists of a single component, since if there are two or more, than nodes of different components can always be exchanged with each other in the ordering. The next thing we have to assume is that the layer decomposition is consistent with a reasonable graph visualization algorithm, which always aims for the shortest possible edges. For example, on Figure 3.2 the yellow node in the second line might be positioned one layer higher, but it would make no sense to draw it this way, since this layout of the graph is much simpler for the human eye to overview.

This line of thought enables us to assign a rank to each article in our subnetwork: the number of its layer in the layer decomposition (shown by blue numbers on the figure). If we number the layers from top to bottom, then the lower the rank of an article, the more important it is inside the network.

3.2.4 Covering by multiple articles

Now let us turn our attention back to the original goal of identifying a set of articles which is possibly small and reaches the majority of the topic subnetwork. First, let us introduce a simple terminology: we say that an article A *covers* other $B_1, B_2, B_3, \ldots$ articles if they refer to A, either directly or indirectly (just as described above on Page 32). In this setting, the articles $B_1, B_2, B_3, \ldots$ is called the *covered set*. Applying this idea to multiple articles, we say that the set of articles $A_1, A_2, A_3, \ldots$ are *covering* the whole subnetwork, if the union of all the articles covered by them and the articles themselves together include all the articles in the subnetwork. The articles $A_1, A_2, A_3, \ldots$ is called the *covering set*. Now we want to see algorithms that find a small set of covering articles in a network. For an example, see Figure 3.3.

How to construct an algorithm to find such a covering set of articles? One approach to this would involve the layer decomposition described above and checking if high-ranking articles are able to cover most of the network. Another possibility is to start with the sinks of the network. A *sink* is a node which has zero out-degree. These nodes correspond to articles that are not citing any other article within the topic subnetwork, but they are cited by others. It is easy to see by induction that in a DAG, (1) there is always at least one sink, and (2) the set of all sinks always covers the whole network. At the same time, sinks often occur in a lower rank position in the layer decomposition.

3.2.5 Component proportions and coverage measure

Choosing sinks as the covering set instead of the high-ranking elements has the advantage that it is independent of any parameters. By selecting all the sinks as the covering set, the whole network is covered. If we want a reasonable alternative with much less covering nodes, but preserving most of the covered ones, we can choose the sinks of the largest component only. Provided that this component is large enough, it will be completely covered, and we can expect it to have a small number of sinks. As a counterexample, we can think of a very small component, consisting of two nodes connected by a single, directed edge. One of these nodes is the sink of this component. In this small component, we need the 50% of all nodes to cover it. In a larger, tree-like structure we can hope for a smaller covering set.

It follows that the method of choosing sinks as coverage set is expected to work better

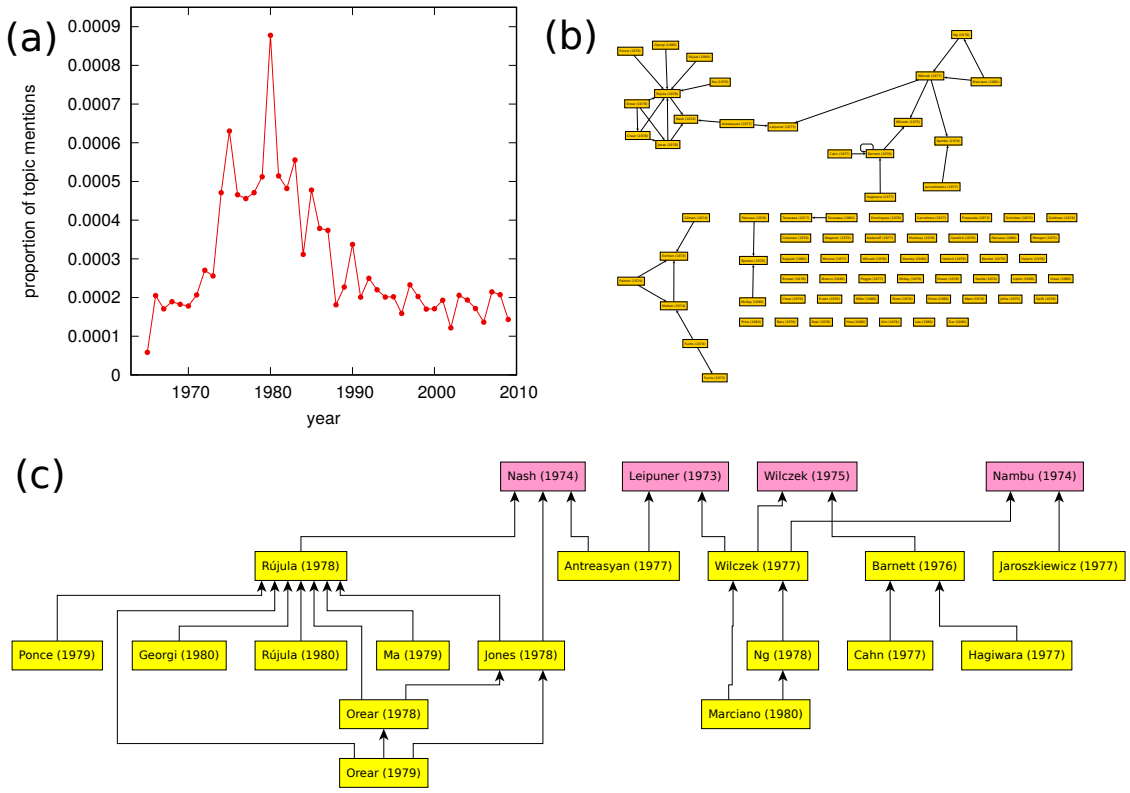


Figure 3.3: *Coverage of the bursty keyword "quarks" by a small subset.* The word *quarks* has a burst at 1973-1980 **(a)**. This timeframe consists of 76 articles **(b)**, of which 32 participate in the citation subnetwork. The rest 44 does not have a connection to other titles containing the word "quarks". In this network of 32 nodes, there is a giant component of 21 nodes **(c)**, which is covered by a subset of 4 sinks. Projected to the whole network, this means that the 12% of the nodes are covering the 65% of the nodes, which results in a coverage proportion of 18% $(12\%/65\% = 0.18)$. Note at the same time that the most influential node, *Rújula (1978)*, is not amongst the covering ones.

when the largest component of the network contains a larger proportion of the nodes. In statistical terms, one speaks about a *percolation*, whenever a giant component appears. A giant component, formally speaking, is a component that contains a constant, positive fraction of the nodes, even as the network is increased to infinity. In our practical case, we can speak about a giant component when it contains more than 50% of the nodes. The percolation can be measured by the proportion of nodes contained by the largest/giant component. In the following, whenever we will mention *percolation* as a numerical measure, we will mean it to be the proportion of this component.

Coverage, on the other hand, can be formally defined as the fraction of the number of

covering nodes and the covered nodes. The smallest this number, the better the coverage (the more it resembles to the Pareto principle mentioned above in Section 3.2). This definition is flexible enough to apply not only to cases when the whole subnetwork is covered by a set of nodes. The intuition is that there should be a significant negative correlation between the percolation and the coverage, since large percolation means that it can be efficiently covered by a small set of nodes (see detailed analysis in Section 3.3.3).

The largest component, therefore, is enhancing the effectivity of the coverage. The other extreme are the isolated points, which are the smallest possible components of the graph (if they can be considered components at all). If we want to have full coverage of the subnetwork, isolated points must always be included in the covering set. But this is contrary to the theory of our goal, which is to identify articles that are highly influential, in terms of their effects on their field. In most cases, articles which are isolated points in their topic subnetworks do not fit this description. They do not have a connection with any other article in the field. Therefore the best we can do with isolated points is to ignore them from our computations. By ignoring isolated points, the proportion of the giant component is increasing (see Figure 3.4).

3.2.6 Ranks, sinks, and their combinations

The disadvantage of simply selecting the sinks while ignoring their position inside the network is that they will include unnecessary sinks as well. Imagine a large tree as an example, with a lot of leaves at the bottom. Now if one leaf refers to an article, which has no other connections, then this article will be such an unnecessary sink. By taking into account the fact that it has almost the same low rank as the leaves, one would be able to filter out such articles from the covering set. See Figure 3.5 for a similar, real-life situation. From this we see that it is possible to take advantage of combining the two different approaches of finding a covering set: using the ranks of the articles and looking for sinks.

Calculating the ranks of the articles has an additional benefit: in some very large topics, the subnetwork contains a high number of articles, and even the covering set is not so easy to process for a human user. In such a case, ranks are useful for putting the articles into order. This is important because the entire goal of this research is to provide a tool for human experts of the field to browse through cases in which one or more articles had an enormous impact on its topic, ultimately causing it to grow, to burst. When after

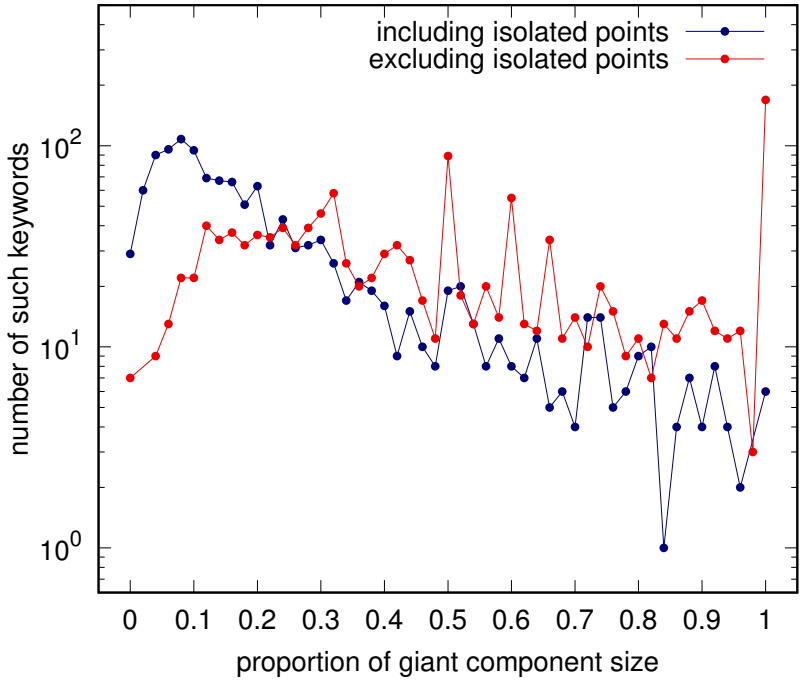


Figure 3.4: *Distribution of the giant component sizes.* By excluding isolated points from the subnetworks of the topics, the percolation effect can be increased. That is, more words produce giant components of larger proportion, which is apparent from the gap between the red and blue curve on the right side of the figure. This method provides additional cases where a percolation occurs and thereby shows topics which would otherwise be hidden.

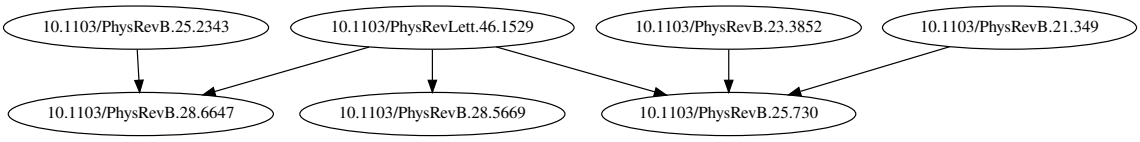


Figure 3.5: *Subnetwork of the topic keyword "segregation".* The network consists of 7 nodes, which form a single giant component. 4 of them are sinks. This is a simple counter-example to the intuition that a relatively large giant component implies that it is possible to cover the network with a small proportion of nodes. Here the half of the network is necessary in order to cover the whole network.

the processing of a big burst of a big topic, a high number of potential causing articles are found, one would want to see the most important ones of them only. For this purpose, the rank can be effective.

3.3 Top burst analysis

For now we have accumulated all tools necessary for delving into specific data. Our aim now is to generalize the concepts presented above for keywords *"magnetic"* and *"model"* (see Figure 3.1) and obtain a map of all bursty topics and influential keywords, connected to one another.

3.3.1 Slope and threshold limits

On the topic diagram presented above we noticed a sudden growth, a burst on these two popular keywords. In order to make this intuition measurable by numbers, one would like to introduce a quantity that measures this phenomenon. In mathematics, differential calculus is used to calculate the slope of a tangent of a function as a limit value of a differential.

Burst implies high slope

We also want to utilize the slope values of these diagrams, but our case is much simpler. For every possible year interval $[Y_1, Y_2]$ of a topic diagram for a keyword w, the slope is simply defined as:

$$s_{Y_1,Y_2}(w) = \frac{freq_w(Y_2) - freq_w(Y_1)}{Y_2 - Y_1 + 1} \tag{3.1}$$

This is the slope of the linear function which is going through the points $(Y_1, freq_w(Y_1))$ and $(Y_2, freq_w(Y_2))$. The reason for the additional 1 in the denominator is that since the type of data is integer here, this compensation factor is necessary for including the start and end years as well (for instance, the 1999-2000 interval is considered to be 2 years). For an example, see Figure 3.6, where the slope values for our example topics are calculated, as well.

Above, in Figure 3.1 we specified a year interval by intuition. Is it possible to have an algorithm to do this automatically? By simply trying every possible year interval $[Y_1, Y_2]$ and saving the maximal value of $s_{Y_1,Y_2}(w)$ one won't get back the "intuitive" year interval given above for this two words. For example, between 1990 and 1991 the word *"magnetic"* has a bigger slope than in the whole interval $[1986, 1991]$. In general, it is always easier to have a "fast forward" for a shorter amount of time than to do this for several con-

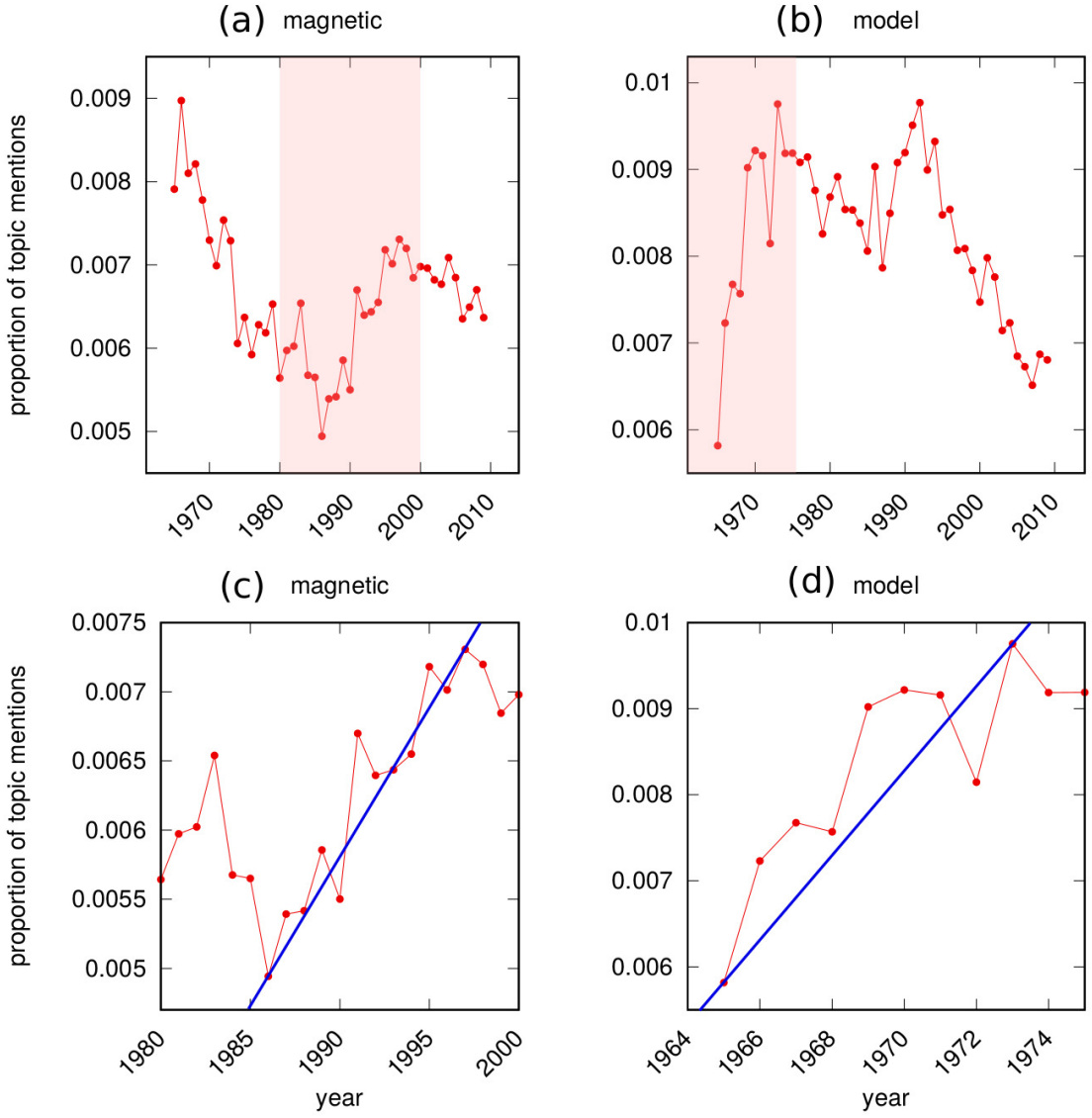


Figure 3.6: *Slope lines in the two largest bursts.* Parts (a) and (b) are the reiterations for Figure 3.1, with their burst interval emphasized. Parts (c) and (d) are zoomed on the burst interval. The blue lines are going through the point of the start of the bursts and the end of the bursts. The slope of the line defines the slope of the bursts, which is possible to calculate with the formula given in (3.1).

secutive years. Actually, for any possible consecutive slope values $s_1, s_2, \ldots, s_{l-1}$, where $s_1 = s_{Y,Y+1}(w), s_2 = s_{Y+1,Y+2}(w), \ldots, s_3 = s_{Y+l-1,Y+l}(w)$, the single year slope must always be bigger than for the whole interval, that is:

$$s_i \geq s_{Y,Y+l}(w)$$

where

$$s_i = \max_{i=1,\ldots,l-1} s_{Y_i,Y_{i+1}}(w)$$

The reason for this is that in this very special case the slope for the whole interval $s_{Y,Y+l}(w)$ is simply the average value of all $s_i, i = 1 \ldots, l - l$ values, and the average must be smaller than the maximum of these measures (provided that they are not all the same).

This means that there is almost no chance for having a maximal slope value for a year interval longer than 1. Therefore, it is not enough to go for the maximal slope value for an algorithm that is trying to catch the year interval in which the burst happened. At the same time, selecting such short intervals as the bursts themselves is also not acceptable, since our goal is to identify articles which initiate the burst. A one-year-long interval is definitely not enough to perceive such an effect.

Identifying the burst by introducing threshold limits

How is it possible to broaden this picture? Longer intervals have the drawback of not being able to reach such a large slope value, but they also have an advantage that during the course of years, a bigger increase can be achieved altogether. So it is indeed better to have more years included in the interval with a lower slope value. For this purpose, we introduce a threshold limit for the overall burst, in terms of the whole frequency interval of the selected word.

For the sake of simplicity, the frequency values on the Y axis of the topic diagram are normalized by the highest and lowest values. The highest value takes 1, the lowest 0, hence every $freq_w(Y)$ value is a number between 0 and 1. A significant burst is expected to expand at least to the 75% of the frequency domain. To this restriction we will refer as the *Y-limit* from now on, while the difference is called an *Y-jump*. This is a parameter which can be adjusted later on, if necessary.

Furthermore, the topic diagram for the keyword *"quantum"* on Figure 3.1 is an excellent example for the fact that a growth can even reach more than 75%, but it still cannot be called a burst, since it is very slow. Therefore, an additional *X-limit* is introduced,

that an increase can only be called burst if it is reached within the 50% of the timespan. This is also a parameter, and especially characteristic for the specific data set. Therefore, when one applies the same method to different data set, one has to calibrate these values again in order to get meaningful results. Specifically, when setting the X-limit, one has to consider the general rhythm which is typical to the data set.

The map of every possible burst

Now let us apply our criteria of setting X and Y threshold limits to the case of the two popular topics mentioned above, *"magnetic"* and *"model"*. On Figure 3.6 it is apparent that *"model"* is actually a burst according to this theory, but *"magnetic"* isn't (it doesn't jump at least 75% of the Y difference in the selected time interval).

Still, one would be curious to know the place of this slope amongst all other possible slopes. Furthermore, such an analysis would be informative in order to learn about the possible threshold value selections. On Figure 3.7, all possible slope values for all keywords are summarized together. It was produced by processing every keyword and topic diagram, and counting every possible time interval and their Y-jumps.

Based on Figure 3.7, we can further specify our threshold limit. Besides including articles above and left to the limit point, an additional layer can be added with smaller Y-jumps, but containing *at most the same number of articles* as the specified limit point has. Practically speaking, the simplest way to achieve this is to draw a line between $limit_1 = (11, 0.75)$ and $limit_2 = (2, 0.5)$, and accept bursts above this line.

From now on, in the case of having more than one of such bursts in the same topic diagram, we will choose the biggest of them. This way, for every topic pertains exactly 0 or 1 burst.

3.3.2 Noise and decay filtering

Figure 3.7 enables us to collect a set of bursty topics and their bursts. But it is not enough to find a burst on a topic diagram. It is necessary to make sure that it actually has a lasting effect. Two issues that might occur even by seemingly large bursts are its *noise* and its *decay* (see Figure 3.8). The two can be, however, effectively filtered out with a single effort.

Noisy keywords do not have any specific trends, they usually look random. Most of

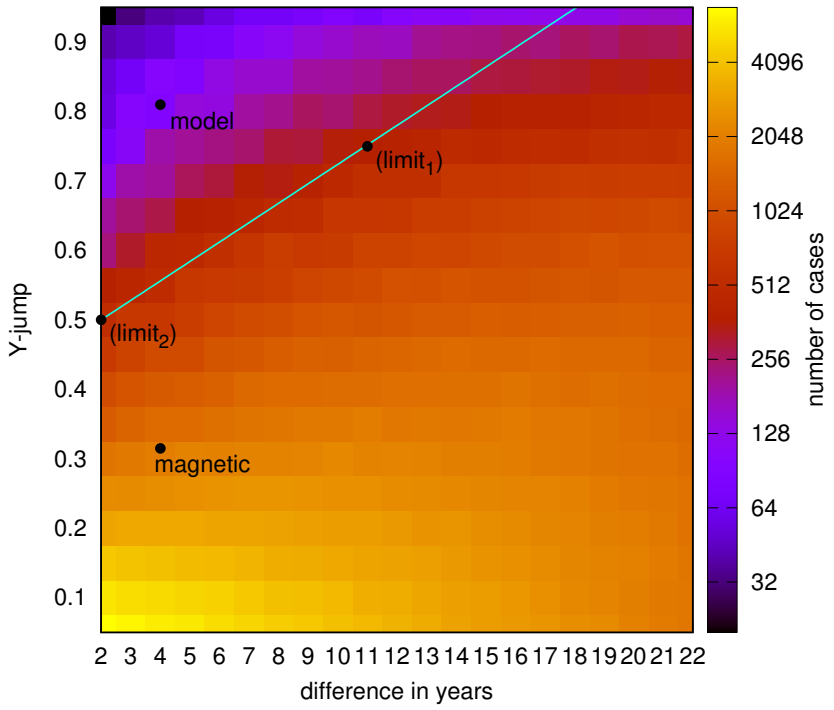


Figure 3.7: *Heatmap about slope X and Y differences.* For a slope to be considered significant, we established that it should produce at least a jump of 75% on the Y axis within a timeframe of 50% of all years present in the data set. The whole data set consists of 22 years, therefore it should be within 11 years. This is represented by the point labeled with $(limit_1)$. Based on this, everything to the top left direction from this point is considered to be a significant burst. As the colors of the heatmap indicate, this area is the least dense part of all. To make the distinction more homogeneous, this area is extended by drawing a line between $(limit_1)$ and $(limit_2)$ and including points above this line, as well, since they do not differ significantly from the original region, by the density of these areas. Keyword *"model"* fits well into this condition, while keyword *"magnetic"* is out.

them are results of the maximum-minimum normalization, which makes their possibly smaller fluctuations much more apparent. It is possible that a topic that is mostly stable around a fixed, large value, has smaller positive and negative changes, which are relatively insignificant compared to the average value, but after normalization one perceives these small changes as big jumps on the topic diagram. The algorithm presented above finding the maximal burst will find one of them and identify as a burst. This does not disqualify the topic diagram as a tool for discovering bursts, as we will see shortly.

Decay, on the other hand, can occur after real bursts. If a topic seem to generate large interest at a specific time, and then suddenly loses it, it might be a reason to disregard the

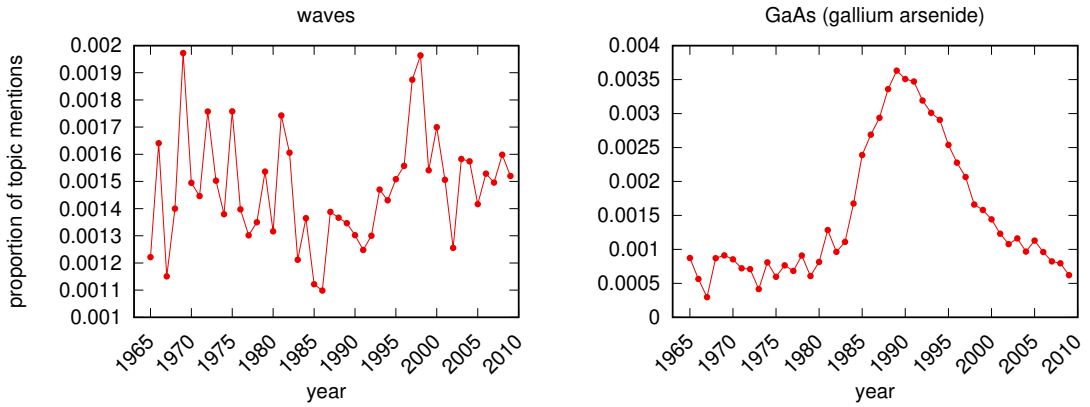


Figure 3.8: *Examples of strong bursts without influential underlying articles.* The keyword *"waves"* has a burst of 94% of its complete Y domain, produced in 2 years (between 1967 and 1969). Nevertheless, this does not imply an outstanding feature of this word, since its frequency values are fluctuating quite strong anyway. The keyword *"GaAs"* (Gallium arsenide) has an outstanding burst, but eventually it disappears almost totally. Both effects are filtered by the Formula (3.2) introduced to measure the decay after (or before, depending on the occurrence in time) the burst.

burst. In this research, the original goal is to find alternative measures of the impact of scientific work. By stating that a specific set of publications contributed to a topic in a significant manner, we mean that they generated a *lasting* burst.

Decay is defined by the lowest point attained after the burst, compared proportionally to the burst itself. Formally, if years $[Y_1, Y_2]$ are identified as the endpoints of the interval of the maximal words, then the decay is:

$$d_w(Y_1, Y_2) = \max_{Y_i \geq Y_2} \frac{freq_w(Y_2) - freq_w(Y_i)}{freq_w(Y_2) - freq_w(Y_1)} = \frac{freq_w(Y_2) - \min_{Y_i \geq Y_2} freq_w(Y_i)}{freq_w(Y_2) - freq_w(Y_1)} \quad (3.2)$$

The value of the decay can be bigger than 1, if the topic diagram eventually falls back to a lower value than it was before the burst. A small decay value indicates a successful burst, which can lead to the discovery of influential articles. A noisy topic implies high decay value, since during the fluctuations it takes a low value as well, which is comparable to the starting point of the burst.

In some cases it is possible that this definition of decay does not make much sense. If the burst occurred at the end of the time range, then it is just not possible that it would produce a decay afterwards, since there is no information available at all about what

happens afterwards. In this case, we define the substitute of the decay called *inverse decay*:

$$\bar{d}_w(Y_1, Y_2) = \max_{Y_i \leq Y_1} \frac{freq_w(Y_i) - freq_w(Y_1)}{freq_w(Y_2) - freq_w(Y_1)} = \frac{\max_{Y_i \leq Y_1} freq_w(Y_i) - freq_w(Y_1)}{freq_w(Y_2) - freq_w(Y_1)}$$

That is, the maximal value before to the burst, compared to the burst itself, in an analogous way as we saw in the definition of decay.

Above in Section 3.3.1 we established the X-limit to 50%, that is, we only consider bursts that occur within at most the 50% of the whole time range. In this experiment, this turned out to be a useful choice of parameter, and we can assume that if the 25% of the time range is still after the burst, then it is enough for calculating the decay. If there is less, then by necessity there is more than 25% before the burst, for which we can calculate the inverse decay. It is easy to see that this is the maximal possible choice to set a minimum for the basis of the formula of decay/inverse decay. Therefore, if one chooses another X-limit in another experiment, then the splitting the remainder of the time range to two halves is always a reasonable rule to follow.

Figure 3.9 shows the comparison of the burst and the decay. For the sake of the comparison of burst with decay, y-jump was used instead of the slope, since it is easier to compare and limit the two values on the same plot. By limiting this figure with a line specifying the maximal acceptable decay, a set of lasting bursts can be obtained. The specific limit was chosen by taking into consideration the number of resulting elements as well as keeping the maximal decay value at an acceptable level. This threshold is a parameter of the model which can be fine tuned, if necessary.

3.3.3 Relations between percolation and coverage

Above, in Section 3.2.5 we introduced the notion of percolation and coverage and suggested the idea of the correlation between the two. For every topic diagram we defined its topic subnetwork consisting of the publications in its timeframe, containing its keyword in their title, with citations running between them as edges. *Percolation* was defined as the proportion of the giant component in this network, while *coverage* is the proportion of sinks in the giant component. The intuition is that if both these numbers are high, this indicates a strong burst, a case where a small number of articles contribute to the popu-

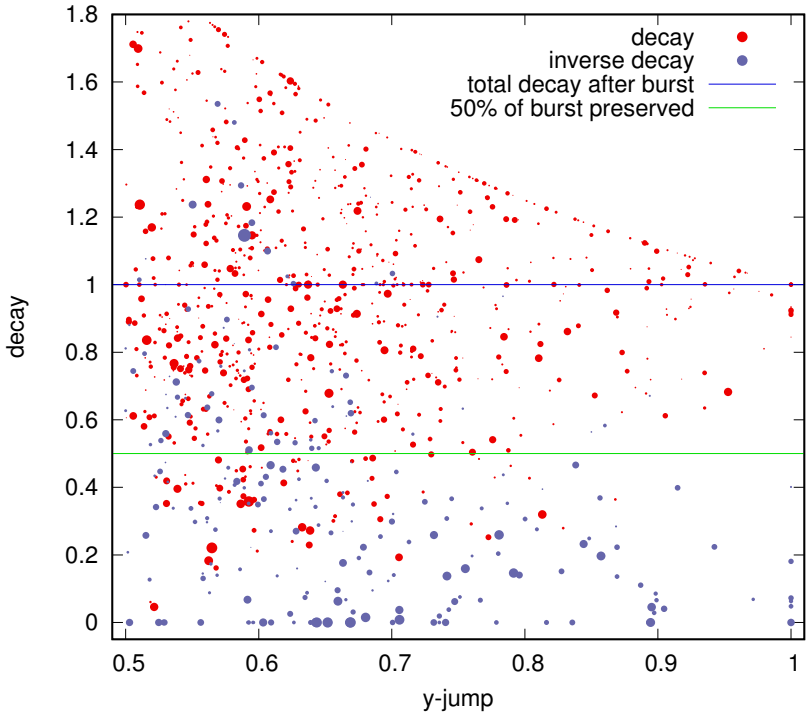


Figure 3.9: *Burst and decay compared.* The measurement used for burst here is the y-jump, the difference of the frequency of the topic keyword between before the bursty interval and after it. It is comparable with the decay, which measures the lowest value reached after the burst compared proportionally to the burst itself. In order to be able to speak about a burst, it is necessary that the burst should take a short amount of time. This is limited by the X-limit (see Section 3.3.1). The y-jump scales up to 1, since a burst can jump maximal throughout the whole Y axis, which is normalized to 1. The decay has a limitation which is apparent at the top of the scatter plot. Obviously, an y-jump of 1 cannot be followed by a decay larger than 1, since the topic diagram is normalized to 1. Similarly, by decreasing y-jump, proportional decay values can rise. Elements below the green line correspond to bursts which succeed to preserve at least half of their increase. Red elements correspond to bursts for whom the definition decay is meaningful, that is, they have time left of the time range even after the burst. For all the other elements, which are colored blue, inverse decay is applied. Red elements has an advantage of known stability even after the burst. Therefore, the biggest red elements are shown separate on Figure 3.10.

larity (measured by yearly word frequency) of the whole field, increasing the mention of the topic keyword. (See Figure 3.3 for a demonstration how this intuition should work on a specific example.)

Now, after having defined exact measures for burst and its noise, we can examine their actual relations. Figure 3.11 compares the measures defined so far for a burst: slope,

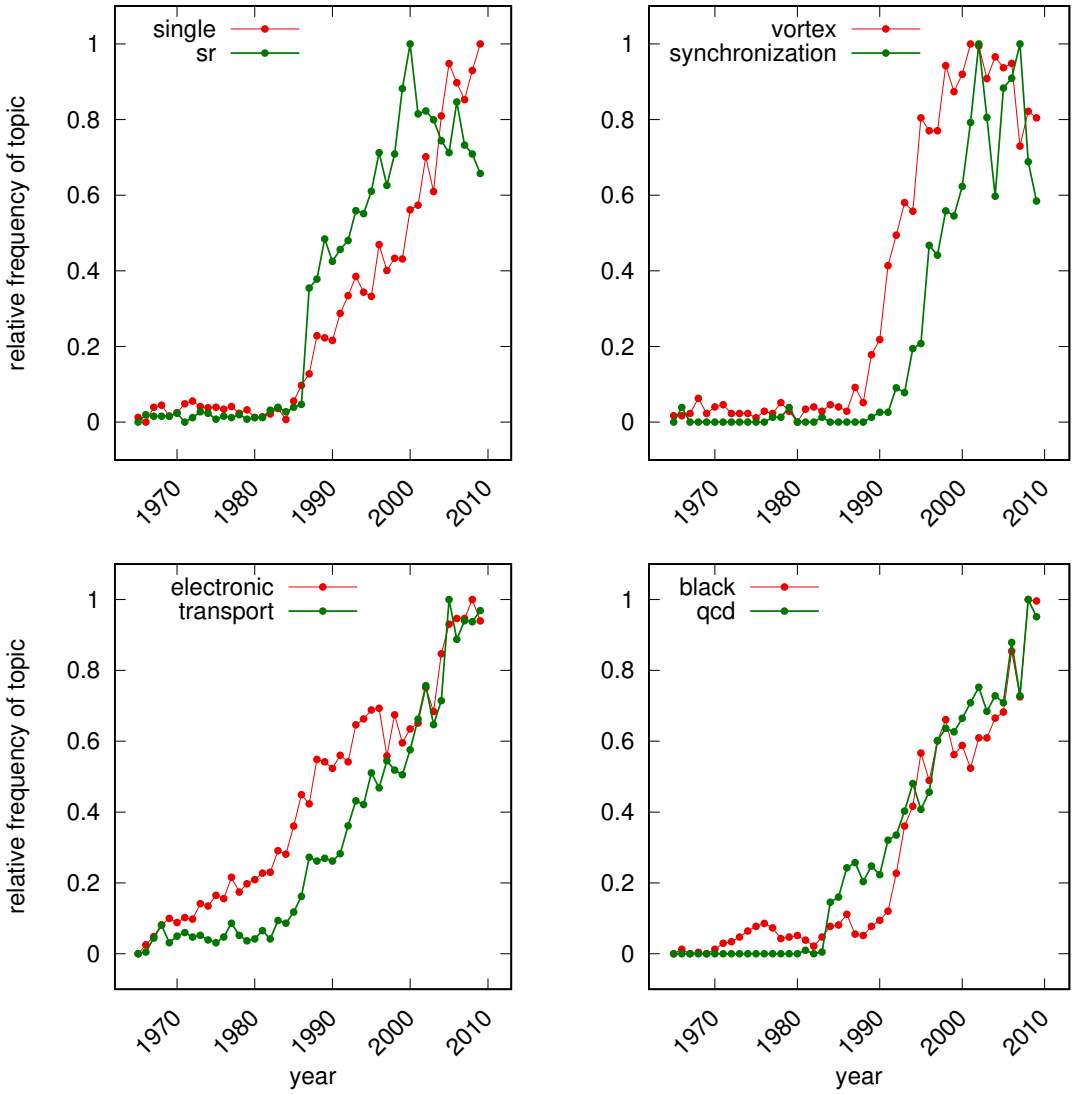


Figure 3.10: *Some prominent examples of bursts.* Selected from Figure 3.9, the topic diagrams of the 8 largest red points below the green line. The green line provides the feature that they are lasting bursts in the sense that their topic diagram does not reach below the 50% of the original burst. By selecting the red points (instead of the blue ones), for some topic diagrams the events after the bursts are visible as well. Not for all of them, because the condition for a point to be red was to have at least 1/4 of the time range after the burst, and the burst can occupy at most the 1/2 of the time range. Therefore, if a burst starts within the first 1/4, it will be considered red, even if it continues to rise afterwards, such as in the case of the word *"electronic"*, for example.

percolation and coverage. A slight correlation between these measures indeed exists, as confirmed by this test. Furthermore, as it is apparent from the figure, since low coverage

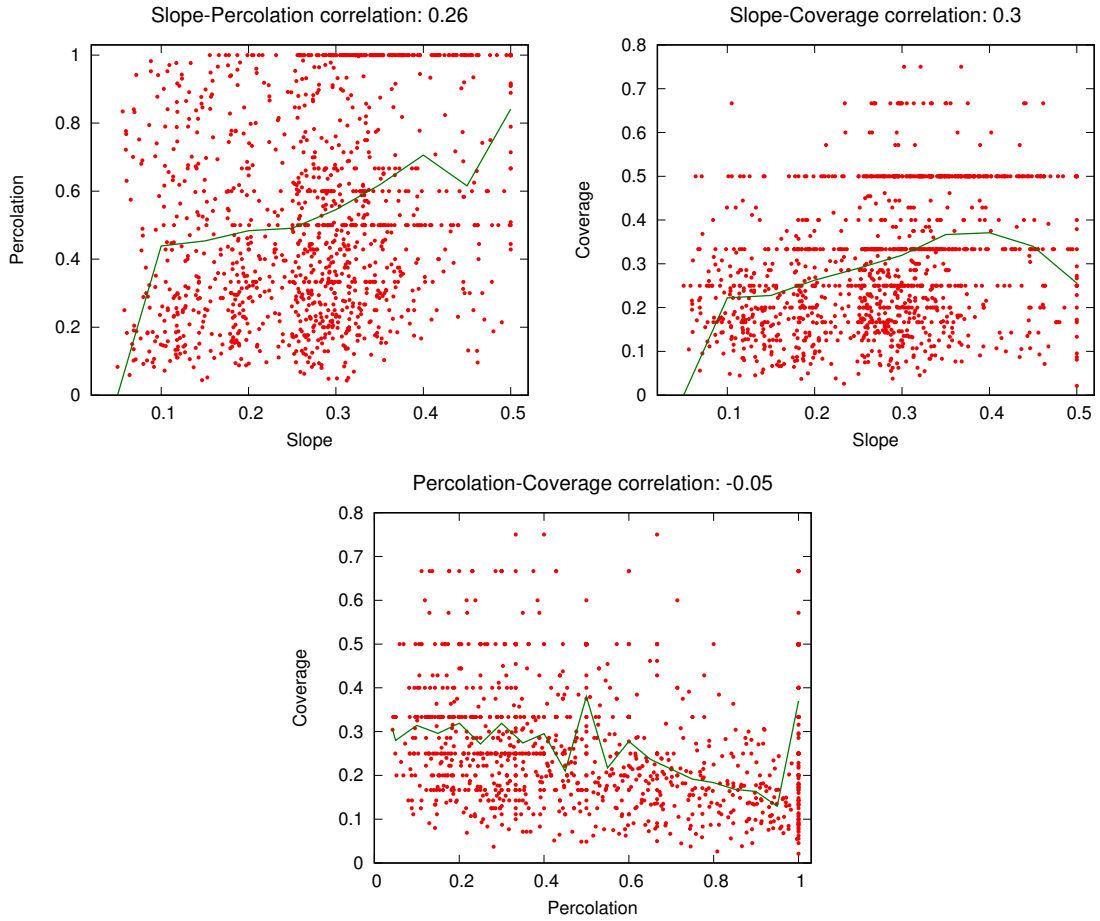


Figure 3.11: *Correlation of slope, percolation and coverage.* Vertical averages are shown by the green line. The Pearson correlation of the X and Y axes are shown in the title of every plot. The correlation of these measures against a random sequence produced 0.05, therefore the correlation of slope is significant with both measures. A relatively large giant component increases the probability of a burst, as expected. Similarly, a small coverage implies a small number of covering articles, which also increase the probability of a burst. This produces the average curve to turn back and as a result, decreases the correlation between the two quantity. At the same time, percolation and coverage do not correlate. This is a classic example of the lack of transitivity of the Pearson correlation. For larger correlations, although, there is a formula based on a trigonometrical analogy.

is correlated with high slope of burst, this hints that the method is applicable to identify a small number of articles which cover a larger number, and they are indeed responsible for generating a burst.

3.3.4 Navigating the bursty topics and influential articles

After having introduced various measures to analyze a topic and its bursty behavior, the concluding task is to put them together in a framework which simplifies the access to it. The results are accessible online, running at `http://topinav.elte.hu/burst/`, the source and data files at `https://github.com/binyominzeev/burst`.

The main page of the online interface can be seen on Figure 3.12. All words resulting from the filtering mechanism described on Figure 3.9 fit on the screen in an arrangement that is easy to browse. The words are clickable, their place is determined by alphabetical ordering, the size is based on the size of the topic, while the coloring varies on a gradient between red and green, and the specific measure can be chosen in the top menu. This makes it very efficient to understand the working of the different measures presented above, to find outstanding elements according to one of the measures, and also to analyze a specific keyword according to all measures.

By choosing a keyword, one gets to the next level, where the largest component of the topic subnetwork is shown (see Figure 3.13). Also here, the publications are shown in a similar "cloud-like" format, by a coloring of a gradient between red and blue. This coloring is based on the ranks, described in Section 3.2.3. As suggested there, the visualization considers ranks and sinks at the same time, by highlighting the sinks on both the graph- and the cloud-based visualization, which are both available on one page, and showing their rank based on the layer decomposition on both views. The rank values are actually calculated by the Graphviz visualization application, by accessing the Y coordinates of the nodes and using these values for ordering them.

3.4 Conclusion

In this chapter, we presented and analyzed the concept of bursts, which is defined as the unusual growth of the mention of a keyword in the publication titles. While in Chapter 2 the emphasis was on the effect that a publication exerts to other publications, here the focus is on the effect on a larger topic, identified with a keyword. We described a method and provided a web-based application that is able to point out and browse the bursts of a data set.

For every burst, we identified its topic subnetwork, which consists of the publications which contain the topic keyword and were published within the time range of the burst.

Figure 3.12: *The most bursty keywords in APS data set.* The words shown here are the same as the points below the green line on Figure 3.9, that is, all bursty words are limited by a decay/inverse decay value of 50%. The figure is a screenshot from the online, browsable version of the burst data set, available at `http://topinav.elte.hu/burst/`. As apparent from the header, it is possible to choose different coloring of the nodes. In any case, they appear in alphabetical order, the font sizes are proportional to the number of nodes in the giant component. All words are clickable and analyzable further (see Figure 3.13), except the largest ones, which are excluded because of computational and visual complexity. The currently selected coloring is based on the maximal slope value of the burst (red is larger value, green is smaller).

This topic subnetworks are analyzed by the relative size of their largest component and the structure of this component: whether it is possible to cover the whole component by a small number of nodes. If so, then the burst itself can be attributed to these nodes. By using the web application, one can find and analyze such bursts in any selected field inside the data set.

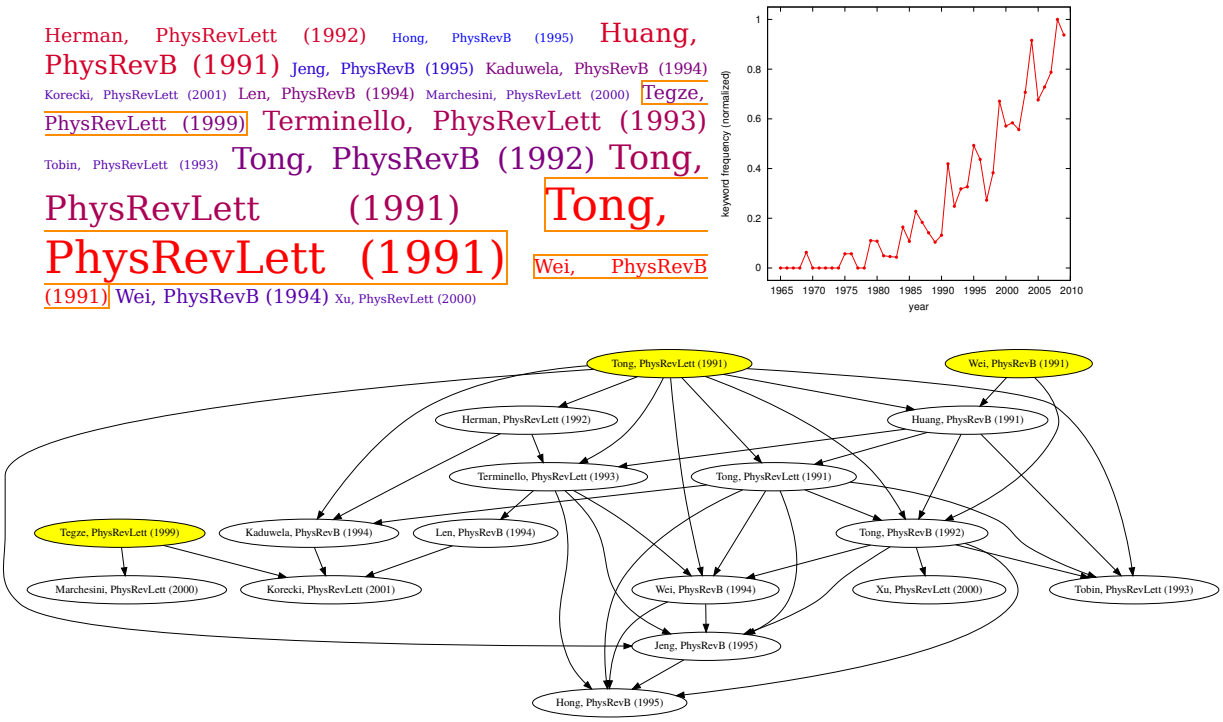


Figure 3.13: *Article cloud for keyword* "imaging" *in the APS data set.* This is the result of the web-based application presented on Figure 3.12 after selecting a specific keyword. For every article, the first name of its first author, the abbreviation of the journal name and the year of publication is showed. Only the giant component of the topic subnetwork is analyzed in the article cloud and on the network hierarchy. By contrast, on the topic diagram, every publication is considered. Sink nodes are emphasized in the cloud by an orange border, on the network by a yellow background. For the sake of the hierarchical visualization algorithm (Graphviz – see above, Figure 3.2 and [27]), the edge directions are reversed, so sinks actually appear as sources. The article colors represent the levels in the graph visualization, as presented above in Section 3.2.3 (red corresponds to a higher level, blue to a lower). The sizes in the cloud represent the citation count of the publication inside the topic subnetwork. By clicking on a selected article in the cloud, the web-based application opens the original publication.

Chapter 4

Comparison of similarity measures

In this chapter, a practical comparison is presented, evaluated on 4 big data sets. Similarity measures are important basically for two main reasons: for classification/clustering applications or for recommendation systems. For different input data types, a number of similarity measures were suggested in the recent years, up to the point when it became necessary to create benchmark frameworks to evaluate them (see Section 4.1), according to their fields and purposes.

In our case, the similarity of *titles* are compared, in order to keep things as simple as possible, which are part of a time-evolving data set. Furthermore, these titles are representing items which can have *links* to each other. This general definition gives an exceptional amount of flexibility for the present research.

The basic idea behind the comparison of the measures is to look for *nearest neighbors*. Every word has a nearest neighbor, and the distance from that nearest neighbor is collected for each of them. Subsequently, the 4 similarity measures are compared based on the number of the words in which they are succeeding to offer the nearest possible neighbor. In other words, for each and every word 1 of the 4 measures is chosen, the one that is providing the nearest neighbor (all 4 measures are normalized to the same exact scale, between 0 and 1). The idea behind this is that a neighbor which cannot be found with one similarity measure, might be possible to find with another one, in which case, the latter is deemed to be more appropriate for this purpose.

4.1 Background: similarity evaluation in different fields

In general, the overview of the literature shows clearly the originality of the idea of the present research: measuring the quality of any kind of similarity measure, without any background information available from different sources.

4.1.1 Publication similarity

Probably the most similar research is [28], which evaluates a data set of scientific publications. The scientific data set which we use is that of the American Physical Society (APS), which contains more than 400,000 records, while in [28] the data set contains 15,000 records. Besides this, we are evaluating 3 more data sets, each of them are larger than APS.

Another significant difference is that they use the complete, full-text contents of the publications, while we use only their title. The advantage of this approach is that the algorithm is applicable even in cases where the full-text information is not available or not existing. Furthermore, they measure similarity by counting common citations of publications, which is a quite reasonable choice. Nevertheless, this option was omitted from this current work, since on major data sets its computational complexity makes it unable to handle.

Most importantly, the purpose of their work is to compare the similarities in terms of their overlapping results. The efficiency of the measures are not addressed, which is the main point in our work.

4.1.2 Time series similarity

Another close research is [29], which is evaluating 7 different similarity measures on 45 time series, obtained from the UCR time series repository ([5]). Time series analysis is an especially active field with a number of practical applications, therefore the evaluation of these measures is indeed of utmost importance, especially considering the fact that the experts of this area are often suggesting novel measures, sometimes exaggerating the quality of their product, therefore thorough analysis is necessary.

Time series are a special type of input data, and their evaluation of measures are strongly (and correctly) relying on this fact. The standard method of testing is based on

functions that produce time series which are clearly distinguishable by human agents, and the similarity measures used for clustering the time-series are supposed to separate them accordingly.

In contrast, in our research, time series are also considered as a possible way of measuring the similarity of words occurring in titles, based on their yearly (or monthly) frequency. One measure is derived from this approach, and compared with completely different types of measure, which underlines the generality of our method. This flexibility leaves open a further possibility to extend the evaluation to those time series similarities, which come out as most efficient, from their work.

4.1.3 Evaluations based on human feedback

In [30] and [31] different online products are used to compare similarity measures. [30] uses social bookmarking sites and their tags as input data, and presents accordingly similarity measures. These are subsequently evaluated against reference measures, which are similar to them, and also they are used to predict tag relations. [31] is working with the social networking website Orkut and using its interface to measure user clicks. This setup has the advantage that it is possible to evaluate similarity measures against something very real: the reaction of real, living human agents. The algorithm is intended to suggest a social group that the user supposed to be interested, and its success is measured by the fact if the user indeed joined to the recommended group.

It is hard to imagine any better way for evaluation than the human input. The advantage sometimes turns out to be the disadvantage, as well, as far as it is not available under every circumstance. In this respect, our present work has a major contribution, as it needs no external data to validate the similarity measures being tested.

4.1.4 Recommendation systems

Recommendation systems are realizing a crucial task in helping users to navigate around massive amounts of information available all around. From the perspective of the vendors, it is the recommendation system that helps selling a product, since it finds its way to the end user, by offering them something which they are willing to buy, to take the simplest example, in a webshop settings. Of course, the importance of this technique goes way beyond this simple application.

In order to choose the recommended items, these systems need a similarity measure. The success of the recommendation may be largely influenced by an appropriate choice. For this reason, in [32] the authors conducted a research which compares and evaluates the most frequently used similarity measures in recommendation systems. They are using a data set of movie recommendations for users (MovieLens), in which there is sufficient data available for testing: the users are rating the different movies, therefore, the task is to predict, how a specific user will rate a new item, in the light of the earlier personal preferences.

In this case, it is possible to split the data into training and testing set, and evaluate the efficiency of the similarity measures against real world data. Again, our current work has the advantage of being able to run the evaluation in the lack of such information.

4.2 Description and computation

During this research project, the set of relevant words are used only (see Section 1.2.2). The parameter choice $k = 3$ is used as a yearly average minimum number of articles, which, for example, in the case of the APS data set, results a list of 2,700 words, a maximum of 7.3M word pairs (most of which are not counted at all).

This section describes three possible approaches to define similarity. The performance of these measures will be then evaluated on the 4 data sets.

4.2.1 Text-based: consecutive words and co-occurrence

The *CN-similarity* of words X and Y are measured by the number of their occurrence one after the other (consecutive words). This measure is not symmetric, because it counts the number of occurrences of X after Y, while the reverse case (Y after X) is counted in a separate variable. In the latter step of evaluating all words, according to their nearest neighbors, when processing the word X, both words occurring before and after are evaluated.

Nevertheless, this will not give the same exact result as if we would disregard the order of the words completely. For example, if the pair X-Y occurs 100 times and the pair Y-X occurs 200 times, then a symmetric measure would result 300 occurrences, and this would be the number assigned to the word X (provided that no other word surpasses

this result). On the other hand, by using the asymmetric measure, one would end up with 200 as a result, since this is the similarity value for the more similar neighbor.

The CN-similarity is calculated by processing every title of the data set, considering each title as a list of words (only relevant words, see section above). Every element of the list is processed together with the word afterwards, and the number of occurrences of pair of words is thus counted.

The *OC-similarity* of words X and Y are measured by the number of titles in which both words X and Y occur (co-occurrence). This is a symmetric measure, unlike the CN-similarity mentioned above.

The OC-similarity is calculated in a very similar manner as the CN-similarity described above, with the exception that the word list produced from the title is processed by a twice nested loop, which is comparing each word to every word occurring afterwards. This way it is made sure that every possible pair of words is counted (since one of them occurs earlier than the other). Furthermore, the two words are registered in alphabetical order (the earlier of the two is mentioned first) so that the symmetric counting should be thereby preserved.

4.2.2 Network-based: connection count

The *KK-similarity* of words X and Y are measured by the number of links in the network, running between titles containing X and titles containing Y. In other words, this is the number of $A \rightarrow B$ links between all A and B titles, for whom X is a word occurring in title A, Y in B. This measure is clearly asymmetric, since it can easily occur that one specific X word refers the word Y quite often, but not the other way around.

The KK-similarity is calculated by loading into the memory every word, assigned to the ID of every article, since the list of the connections is represented by pairs of such article IDs. Afterwards, the list of connections is processed, and for every connection $A \rightarrow B$ all the words contained in A are counted together with all the words contained in B.

This algorithms necessitates for the whole data set to be loaded in the memory, every word in every title, and all the links. This can be a limiting factor for extremely large data sets. Nevertheless, experiences show that for 4 out of 5 data sets the original, simple algorithm was running and ending in a relative short amount of time (about 30-40 minutes for the US Patents data set).

4.2.3 Time-frequency based

The *TD-similarity* of words X and Y are measured by comparing the similarity of their time diagram (see Section 1.4). A time diagram of a word can be converted into a string of numbers using a general pattern described in Section 1.4.2. These strings can then be easily compared, number by number, and a Hamming-distance type of measure is obtained thereby. The measure is symmetric. Note that this is the only measure out of the 4 in which the lower number corresponds to a closer pair (when comparing all 4 measures, this fact has to be taken into consideration).

What makes the computation of this measure especially challenging is the number of comparisons necessary. In all other 3 measures, only those pair of words are considered which occur at least once together. This in and of itself imposes a limit on the running time of those algorithms. In the case of TD-similarity, however, in theory, any two pair of words can be closest neighbors. The most important observation here is that for our purposes it is not necessary to have a complete list of the TD-similarity of every two word. Rather, it is sufficient to choose a non-empty list for every word in which the closest neighbor might reside.

This gives the idea to calculate this measure using the following innovative solution: the whole state space (every possible string of numbers, representing a time diagram) is divided into boxes, and for every word, all neighboring boxes are checked for the nearest neighbor. It is impossible that a word could have a neighbor which is closer than any element in the neighboring boxes (since the algorithm chooses all surrounding boxes). The box itself is nothing else but a simplified string of numbers. The algorithm which assigns the box to a word is the exact same algorithm as the one that is producing a string out of the time diagram. The only difference is the parameters used. For an example of generating box strings, see Figure 4.1. For an illustration of the idea of using neighboring boxes, see Figure 4.2.

For such a computation to be effective, it is crucial to examine the way to choose the parameters of the boxes. During the course of the algorithm, all boxes are processed. For every box, all the neighboring boxes are chosen. The words residing in the central boxes are then compared to each other, as well as to the words in the neighboring boxes. Two boxes are considered to be *neighboring* if in each of their coordinates the difference is maximum 1. For example, 134 and 045 are neighboring, since in every coordinate the distance between them is only 1.

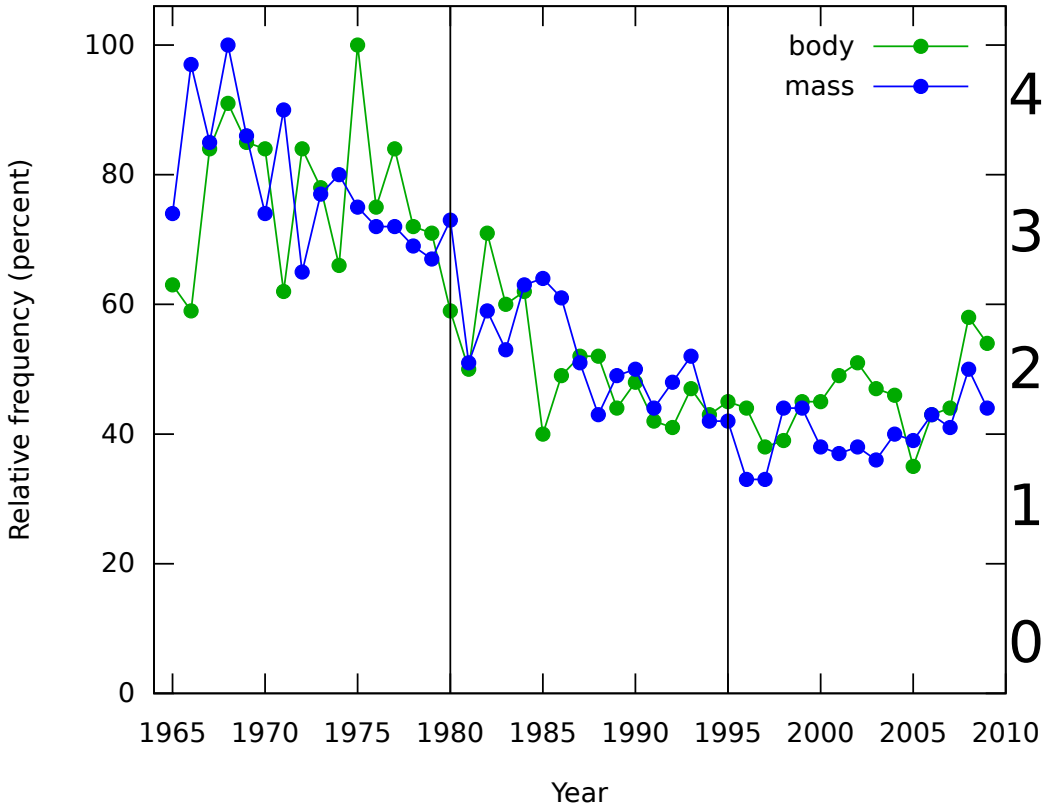


Figure 4.1: *Time diagrams, strings and boxes.* The nearest neighbor of the word *"body"* in the data set APS is the word *"mass"*, based on their respective time diagrams. They are residing in the same box, represented with the string 322, produced with parameters $l = 15$ (which corresponds to $k = 3$, see below) and $h = 5$. The box code is generated by dividing the X axis into $k = 3$ equivalent portions of $l = 15$ years, and for each of them generating the average of the values inside. In the first column, they reside between 60 and 80 on the Y axis, which corresponds to the character 3 (see right side). Note that although these nearest neighbors happen to reside in the same box, however, this is not necessarily always the case (they might also occur in neighboring boxes).

It follows from this definition that if the box string length is k, then the number of all neighboring boxes is 3^k, since every coordinate can either stay the same, or increased by 1, or decreased by 1 (3 options altogether). Furthermore, it is also necessary to choose the number of possible coordinates (earlier denoted by h) to be bigger than 3, otherwise the boxing algorithm would produce the exact same algorithm as the naive approach, which compares every word by every word (since in that case by choosing a box and all their neighbors the whole state space is covered already). On the other hand, by choosing a too large value for h, empty boxes might result, which would cause the algorithm not

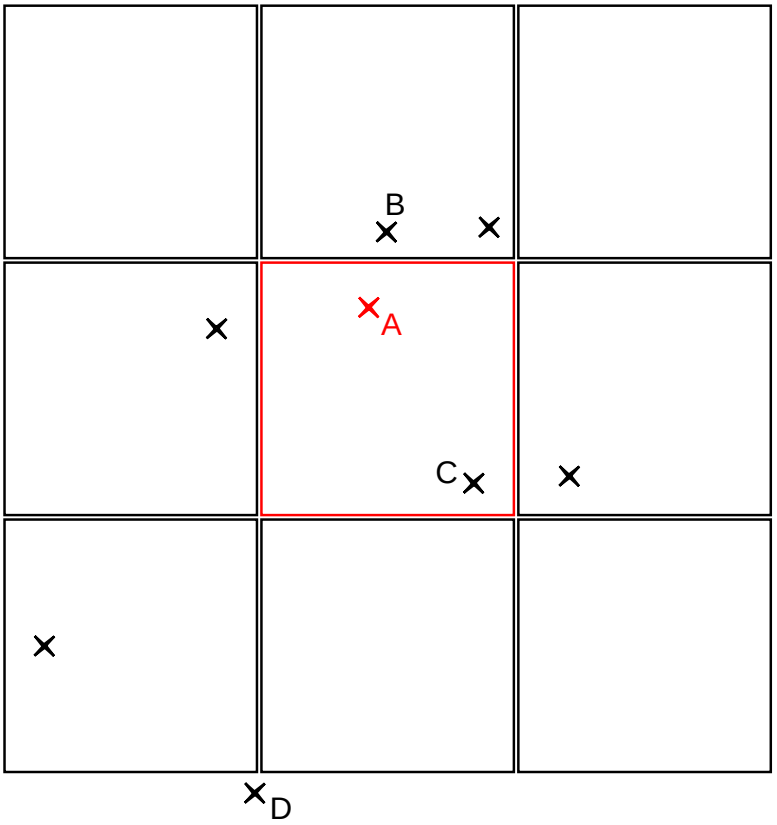


Figure 4.2: *Finding the nearest neighbor using boxes.* This example illustrates the idea behind finding the closest element to the one denoted by A. If we were just relying on a comparison withing boxes, this would result C as nearest neighbor, which is false, since the real nearest neighbor is B. In order to perform a complete check, we must consider also those neighboring boxes, which are just touching the corner of the original box, since the nearest neighbor might reside there, as well. Once the boxes are checked, and we found at least 1 element in them (and, in an extreme case, even 0 in all other boxes), the nearest neighbor cannot be anywhere else. The element denoted by D is illustrating this artifact.

to find the nearest neighbor within the neighboring boxes, and thereby fail to produce a meaningful result.

Based on the considerations above, we have chosen $k = 3$ (which is realized by choosing $l = 15$ in the APS data set, which covers $45 = k * l$ years) and $h = 5$. Thereby, one box and its surroundings are covering $\frac{3^3}{5^3} = \frac{27}{125} \approx \frac{1}{5}$ of the whole state space, which, in practice, turned out to be an effective compromise, producing reasonable running times.

For every box, a list of the words contained in the box are listed. Respectively, for all neighboring box, a second list is collected, containing the words contained in those boxes. (Here, the box itself is also considered to be the neighbor of itself, since it is

quite possible that the nearest neighbor of a word will be found inside the same exact box.) Then a nested loop compares all words in the original box with all words in the neighboring boxes, and registering the closest neighbor for each word, and their distance. Finally, the list of all words, their nearest neighbors and the distance between them, are listed.

4.3 Methods of similarity comparison

The key novelty of the idea being presented here is to compare the most different similarity measures, without any further input information. In this section, the details of this process of comparison will be described.

4.3.1 Evaluating the measures

In order to compare the measures presented above, they are to be mapped on the 0-1 scale. For every word and every measure, the distance from its nearest neighbor is obtained, and the word will give a "vote" for that measure which will provide the closest neighbor, out of the 4 options.

For example, the word *"approach"* in the APS data set has a neighbor with a similarity of 63% according to the CN-similarity, 75% according to the OC-similarity, 85% according to the KK-similarity, and 53% according to the TD-similarity (since this latter one was originally a *distance measure*, therefore, it was subtracted from 100%, in order to be able to read as *similarity measure*).

Since, regarding this word, the KK-similarity was performing the best, therefore, the word *"approach"* will count as a +1 "vote" towards the measure KK. Similarly, all words are processed, and in every data set the number of winning words are counted. Based on this, we will be able to determine the best possible similarity measure, in this sense, which is fitting to the data set. Also, we will be able to reveal deeper insights regarding the nature of the data, which every time advocates a different measure, based on its innate attributes (see next Section).

4.3.2 Normalization of the measures

In order to be able to perform this test, we will need to normalize the input data provided by the first three similarity measures. The fourth one, the TD-similarity, is obvious, and is easily mapped onto the 0-1 scale by simply dividing it by its maximal possible value. For the other three, however, we will need some deeper idea, since they can take virtually any value, and dividing them by maximal possible value (or even changing to logarithmic scale) will unavoidably scale all these measures down to the point where they will have no chance against the TD-similarity, which has a much simpler structure.

First, we observe that this data of the distances seem to follow mostly a power law (see Figure 4.3). This gives the idea of instead normalizing their values directly, it is more accurate to determine their position on this scale, and based on this position we can define their normalized value. For this, we use the fitted function instead of the empirical data, since this tends to be more consistent, especially in the right end of the curve, where the large numbers are occurring sporadically (see Figure 4.4).

During the fitting process, a Kolmogorov-Smirnov test was performed to make sure if the distribution actually follows a power law. Although not all of them gave an affirmative result which would justify the assumption of a power-law (as it might be apparent from Figure 4.4, for which $p = 11.7\%$, instead of an expected $p \leq 5\%$) nevertheless, as a means for normalization, it is effectively serving its purpose, as later results show.

4.4 Comparison and results

The resulting distribution of nearest neighbor distances is on Figure 4.5. The central result of the current work is presented on Figure 4.6, which shows a ranking of all measures, regarding to all data sets. The background data for this figure is found in Table 4.1.

4.4.1 Comparing performance of text-based measures

During the course of the experiment, we used the CN-similarity and OC-similarity, which are based on a very similar idea (see Section 4.2.1 for definition). This gives rise to the question whether they produce different results on different inputs. The experiment shows that this is indeed the case, since in all 4 cases we found a significant difference between these two measures (at least a factor of 2.6 is between them, which cannot be attributed

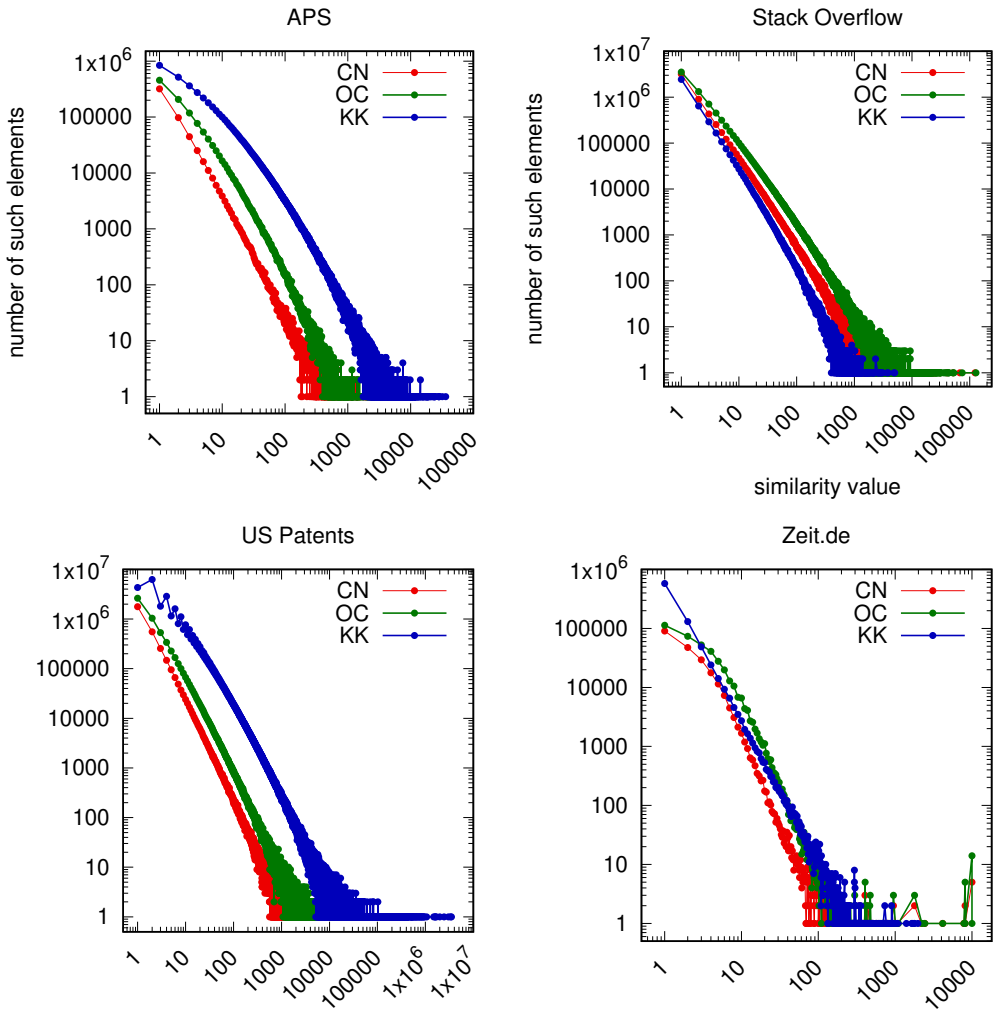


Figure 4.3: *Distribution of similarity frequency.* This frequency histogram is based on the raw measures introduced above, on all word pairs, using word-based (CN, OC) and network-based (KK) similarity. Time-based measure (TD) is omitted, since it already has a natural normalization, inherently based on its definition, therefore further computation is unnecessary. Seemingly, all these measures are scaling according to the power-law, which shows a linear function on a log-log scale.

to observational error).

In the APS and US Patents data set, CN performs better, while in SO and Zeit, OC prevails. The simple possible explanation for this phenomenon is that CN is counting repeating pairs of consecutive words, and this two data set has a more formal, scientific language. It comes out that their language tends to use more phrases, made up of several words. On the other hand, SO and Zeit are less bound, therefore, their users are using the

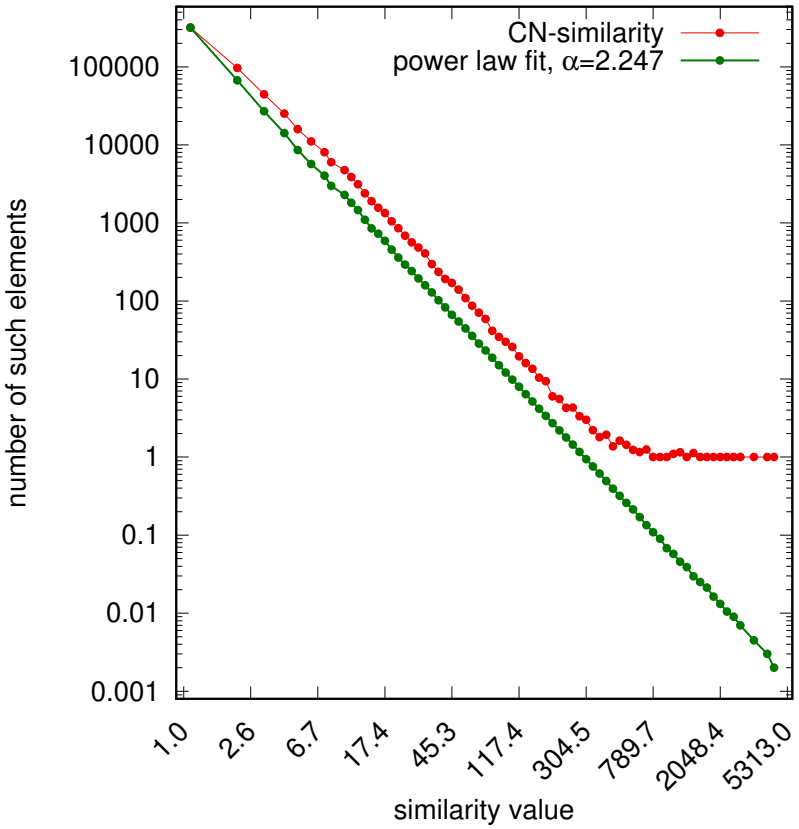


Figure 4.4: *Normalization of the measures based on similarity frequency.* As an example, the distribution of the data set APS is taken, using the CN-similarity (see Figure 4.3). The power law fitting algorithm provided by [33] is used to determine the power law exponent and p-value for the Kolmogorov-Smirnov test. Since this method does not return a point of reference where to shift the fitted curve, the first (topmost left) point was chosen deliberately. The normalization gives a number between 0 and 1, where the topmost left point of the green line corresponds to 0, the bottom right point to 1, and everything in between is calculated proportionally, based on the linear function on the log-log scale. Once the similarity value is determined, only the green, fitted line is necessary for this calculation. As a result, a finer measurement is obtained for the most similar pairs, which are to be found at the right part of the red curve, and switch often between 0 and 1. (The values above are following a logarithmic binning of base 1.1, thanks to Maria Ercsey-Ravasz for the suggestion.)

language in a more flexible way.

4.4.2 Network-based measures and network connectivity

The performance of the KK-similarity, which is based on the links running between mentioned words in the titles (see Section 4.2.2 for definition) show some correlation to the

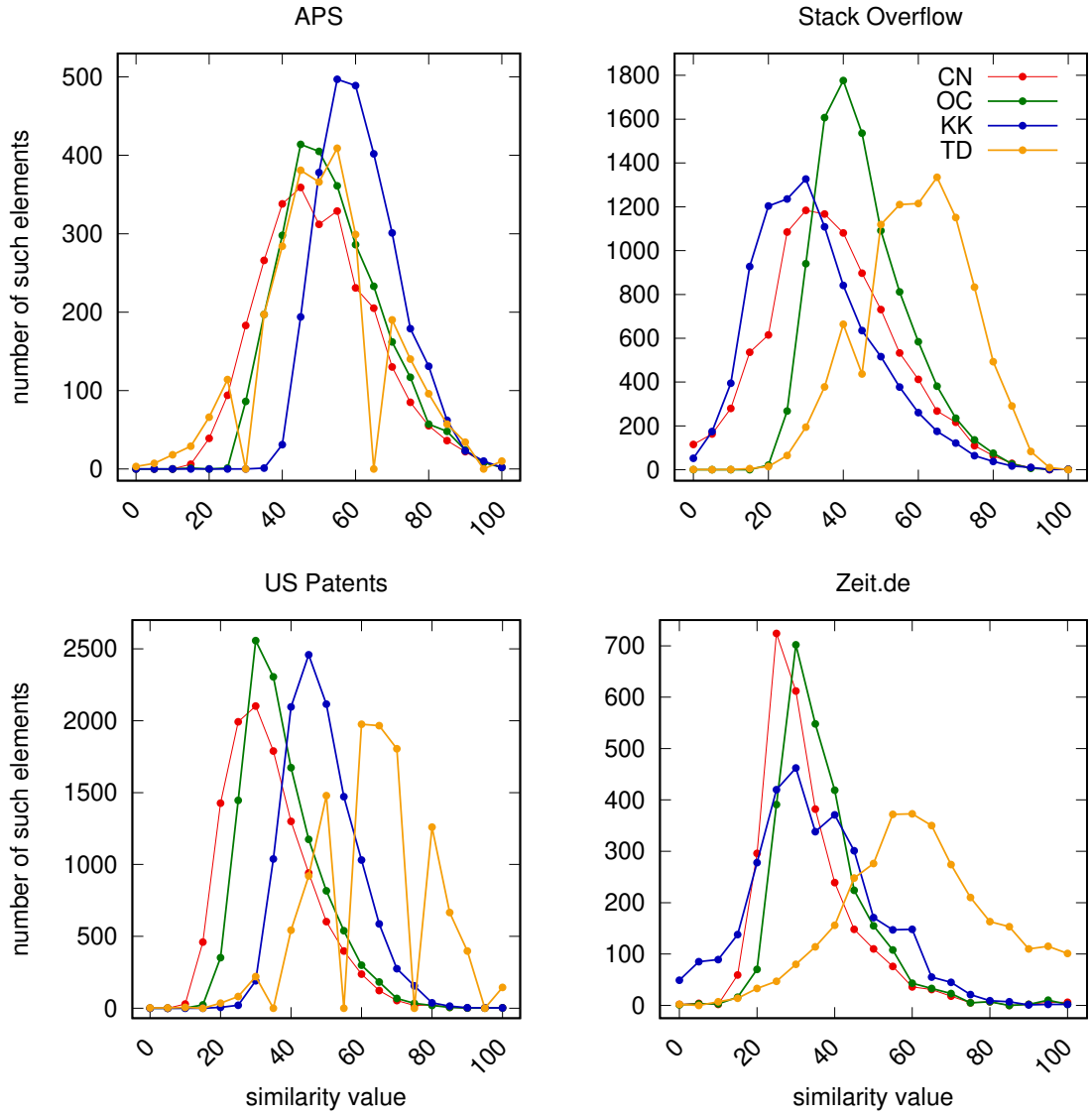


Figure 4.5: *Distribution of nearest neighbor similarity of words.* For every word, the (normalized) similarity value with its nearest neighbor is taken into account. Larger values indicate stronger similarity between two words. A measure is defined to be successful if it manages to identify as close neighbors for a word as possible (see Section 4.3.1). Therefore, the most successful measures are those that have their maximum at the rightmost part of the figure: KK, in the case of APS, and TD in all other cases.

density of the network, measured in the simplest way by calculating the proportion of the edges per nodes (see Table 4.2).

In APS and Patents, where the performance of KK is the best, the number of edges are also larger. On the other hand, in SO and Zeit, where the number of edges is lower,

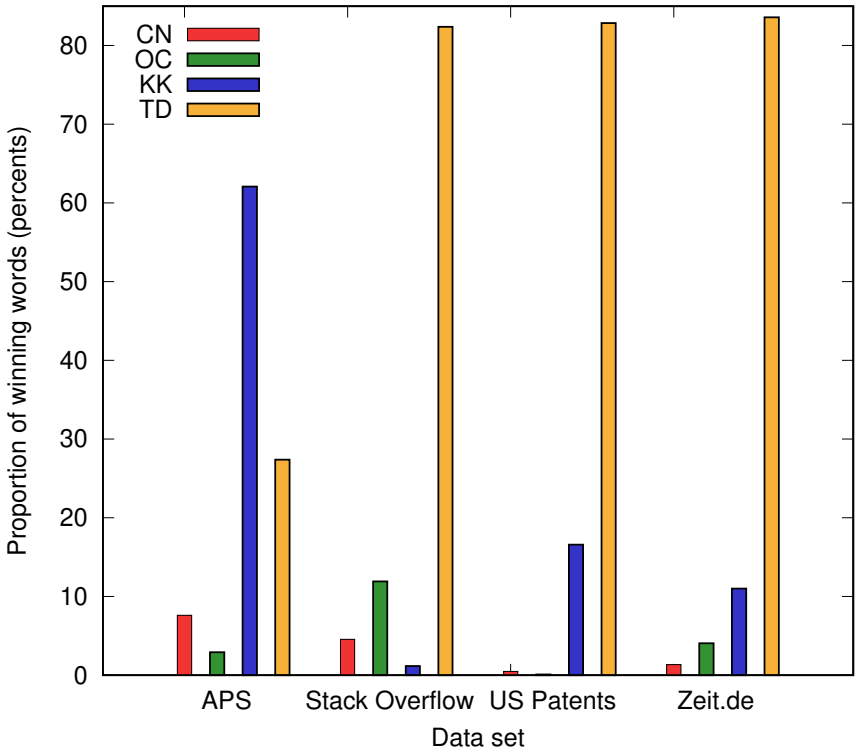


Figure 4.6: *Evaluation of all similarity measures.* Data based on Table 4.1. TD-similarity prevails in almost all data sets, except for the APS, where KK is the most successful.

Table 4.1: *Data set vs. similarity measure comparison.* For each data set and each word, one similarity measure was chosen which is able to find the closest neighbor to that word. The table below shows the number of these words, for each measure. In each line, the maximal element ("winning" measure) is bolded. See Figure 4.6 for visualization.

Name of data set	CN	OC	KK	TD
American Physical Society (APS)	207	79	**1678**	740
Stack Overflow (SO)	433	1133	110	**7832**
US Patents	55	11	1907	**9531**
Zeit.de	44	132	358	**2720**

also KK stays below. At the same time, the large gap between the Patents and Zeit data sets, in terms of edge density, is not reflected by the performance of the KK-similarity.

Table 4.2: *Nodes, edges, and their proportions.* In order to compare the various data sets based on their network similarity results, a simple network measurement is made, by dividing the number of links by the number of the nodes of the network.

Name of data set	Records (millions)	Links (millions)	Edge per node
American Physical Society (APS)	0.4	4.7	11.7
Stack Overflow (SO)	10.7	0.76	0.07
US Patents	4.8	109	22.7
Zeit.de	0.9	0.19	0.2

4.4.3 Time-frequency based measure and technological development

The most striking feature of the comparison shown in Figure 4.6 is the relative low performance of the TD-similarity in the APS data set compared to the rest of the data sets. The visual meaning of this is that the frequent terms used in the titles are following various time-patterns. A time-series clustering analysis would be necessary to verify this observation, which is suggested as a further possibility of the continuation of the research.

A further, partially seemingly contradicting idea which would come to mind is about the progress of technological development. Since APS is the only data set of the four which consists of scientific publications, therefore it might have a context which is under a less of a pressure to change and more pressure to use consistent language. The US Patents data set seems to be the most similar to it, in this regard, but even there, new innovations are expected to come in the patents, which might explain the difference between them.

In order to verify such a hypothesis, a measurement of the scientific progress would be necessary. The authors of [34] have conducted a similar research on scientific data sets, and a similar comparison in industrial data set (like US Patents) would be useful for such a verification.

4.4.4 A case study: words occurring in the intersection

By creating an intersection of the frequent words of all 4 data sets (and removing the meaningless elements), a list of 37 words remain, which is found to be quite surprising, in the light of the various input and intention (and language!) of these data sets. One common word (*"generation"*) is chosen in order to compare the different features of the measures just introduced and to point out the differences which are apparent on the figures above.

Figure 4.7 is using the same structure as Figure 4.6, just that it is restricted to the

specific observed word. Its columns are much more closer to each other, since they are showing the distance of the nearest neighbor from the word "generation", while on Figure 4.6 a general picture is shown, which is using a statistics made out of all words.

In the APS data set, all measures are producing almost the same quality, except for TD, which is shown on Figure 4.8. There we can understand that the TD-similarity is pointing out the slight differences between the neighboring elements. Regarding the Zeit.de data set, we see a very similar effect.

The US Patents diagram is special, because it shows a very specific, growing pattern, which is reproduced by several words, with minor differences. This explains its success, compared to the other measures. This gives an idea to run a clustering on all time-series, as a possible further research (using the techniques presented in [29]).

The Stack Overflow data set, however, is pointing out a possible issue. Since it is using months instead of years, as timeframes, it consists of 88 timeframes, compared to 40-50, that are in the rest of the data sets. This might be the reason why it is oblivious to slight changes in the neighboring elements, and returning high similarity values, even though they do not appear to be more similar than in APS or Zeit.de. This suggests a further research, which has a purpose to optimize the parameter choice for converting the time series into strings, based on the input data.

Figure 4.9 presents a new way of looking at the close elements of a data. Its list of decreasing nearest neighbors show different type of curves, and their rate of decay is a new possible way to characterize the element in terms of its immediate neighborhood. Furthermore, it also gives an impress about the decision which we took in the beginning of planning our experiment, that only one nearest neighbor should be selected. The deeper analysis of this figure can also help by planning further experiments, which include more neighbors for defining the efficiency of the similarity measure.

A further possible continuation of the present work is to develop an online interface to browse the results of the measurements, similar to the ones presented in Chapter 3 and 5. The basis of such a GUI are the figures of the current section, and a further option could be to list the related words and articles, in a similar style as on Figure 3.12. This is also a possible application for browsing and searching through big data, since we were using very few restrictions regarding the type of the input data. Finally, all calculations and scripts are available online on `https://github.com/binyominzeev/similarities`.

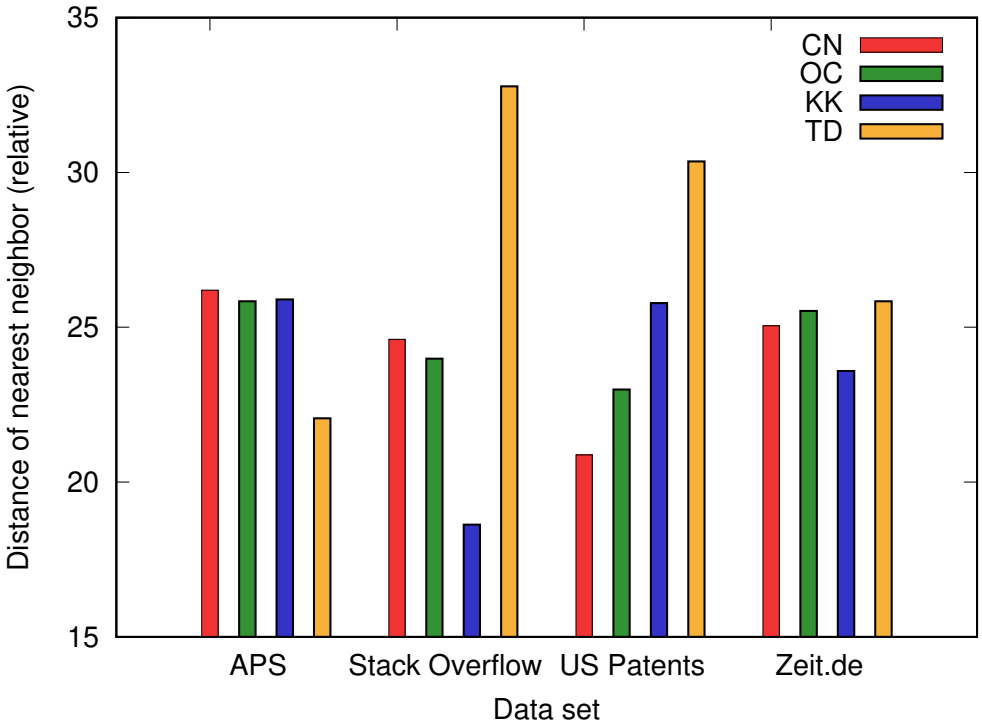


Figure 4.7: *Result of measures regarding the word "generation".* Values are normalized in such a way that within a data set the sum of all measures always returns 100%. This is the reason why the numbers are not the same as in Table 4.3, which contains the original similarity values, but the proportions and order of elements are preserved.

Table 4.3: *Result of measures regarding the word "generation".* The numbers are representing the distance of the nearest neighbor, from word *"generation"*, normalized to common scale (shown in percents), in the selected data set, according to the current similarity measure. In each line, the most fitting similarity measure is bolded, which can be a candidate as a most appropriate similarity for the chosen data set. See Figure 4.7 for visualization.

Name of data set	CN	OC	KK	TD
American Physical Society (APS)	**83.84**	82.68	82.87	70.59
Stack Overflow (SO)	61.25	59.7	46.37	**81.58**
US Patents	59.6	65.62	73.6	**86.67**
Zeit.de	36.94	37.64	34.78	**38.1**

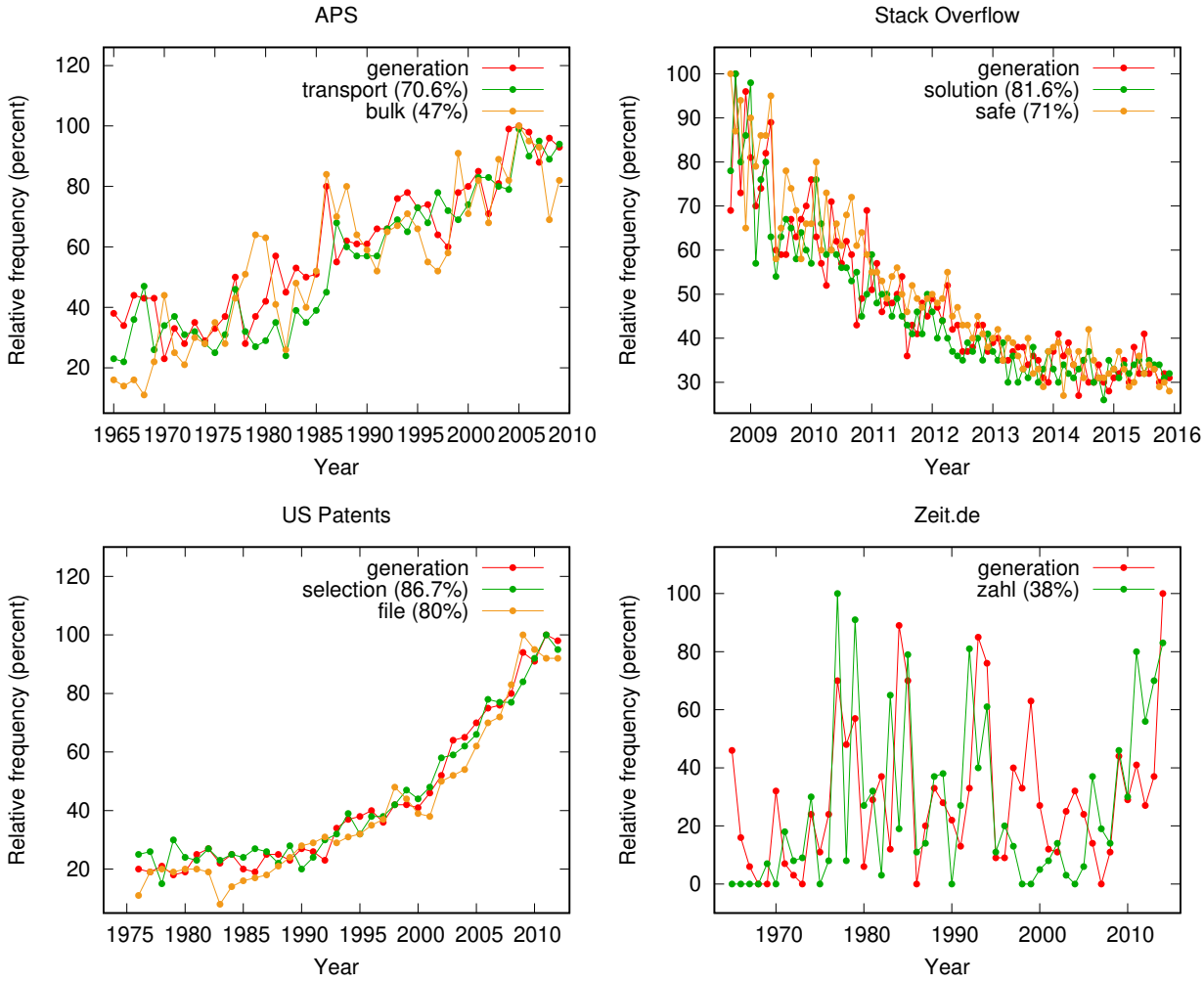


Figure 4.8: *Close neighbors of the word "generation" regarding the TD-measure.* Not all neighbors are shown which are found by the box-method described above in Section 4.2.3, for convenience, since showing multiple curves on a single plot would make it especially hard to see through the data. In the first case, 2, in the second and third, 1 further curve was omitted. In parentheses, the TD-similarity can be read. The more similar element is listed earlier. The result matches the maximal values appearing in Table 4.3.

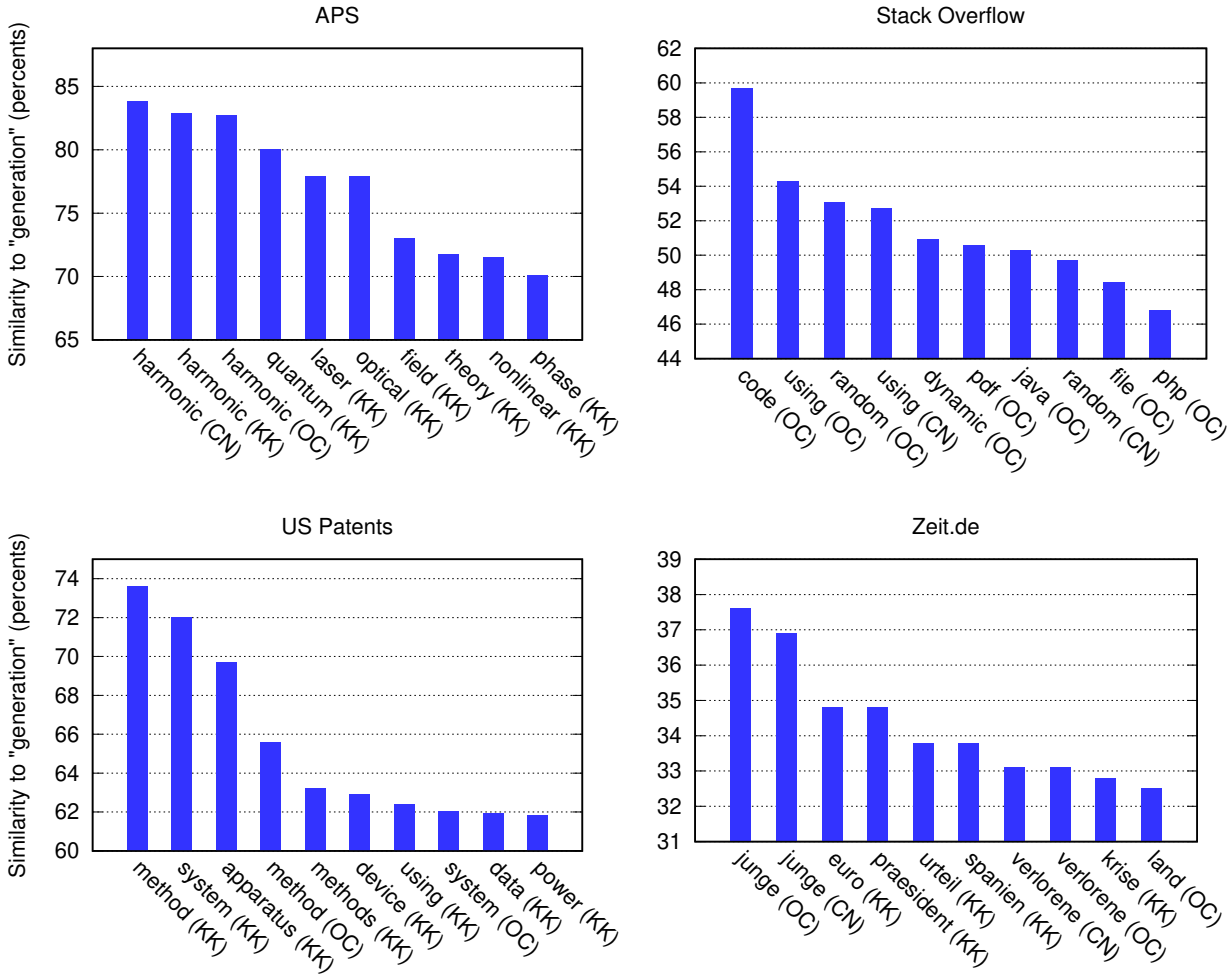


Figure 4.9: *Close neighbors of the word "generation" regarding the CN-, OC-, and KK-measure.* For each data set, the ten closest neighbors, in descending order, were chosen. The neighboring words are written on the X scale, in parentheses, the similarity measure which resulted the visible value. Regarding the asymmetric measures, only the bigger values were taken into the consideration. The TD-measure was omitted from this figure, since it works with a completely different mechanics, and shown on Figure 4.8. The similarity values were normalized to a common 0-1 scale.

Chapter 5

Predicting topic time patterns

In the last chapter, different similarity measures between elements were compared. During this comparison, the concept of *nearest neighbors* played a crucial role in evaluating the effectivity of the measures. In the current chapter, we will use the same concept for the purpose of predicting the future of a topic diagram. The content of this chapter is mostly based on the publication of the current author and the advisor in [35].

5.1 Introduction

One of the most important tasks of data-based research is predicting future trends. The trends can be expressed in almost any data set in terms of *topics* or *keywords*. When one looks at the most basic statistics of an important data set, at the frequency of occurrences of the most important keywords, one can notice their different behavior, in different periods they can be perceived as increasing, decreasing or fluctuating. One would wonder if there is any system of rules behind such a behavior.

Prediction of the success of scientific publications was studied recently by Wang et al. ([36]). Their work is based on observations about typical dynamics of citations. Their model uses a special kind of normalization which fits parameters telling more details about the article, based on its citation history so far. By the assistance of these parameters, one can use their analytical model to plot the citation count of the publication as a function of time, which goes further than the present time, thereby being able to predict the future number of citing articles.

The publication scene, as a sociological culture, has its regular patterns which create

the citation dynamics. This predictive model utilizes these repetitive events. Publications have their natural phases of burst and decay. However, when analyzing topics and keywords, time is a much less reliable factor. Some topics indeed are part of a fashion, as fast they came, so they will fade away. But there are also persistent topics around which have more complex events in the background. In the case of topic prediction, *the relation among the topics* is a good candidate to take over the role of the time factor, which worked good in the case of individual article citations.

The basic idea behind the actual prediction is that a topic keyword is expected to follow its peers: whenever similar keywords tend to increase, the original topic follows, and the same holds true for decreasing. The set of topics can be seen as a weighted, undirected network, where edge weights represent their similarities (see [37] for a review of possible topic similarities).

5.2 Preparing the data sets

During this research project, the set of relevant words are used only (see Section 1.2.2). Rare words that do not occur even in a single year at least with a relative frequency of 0.1% are excluded. This cutoff value was chosen based on Figure 5.1, with the aim of finding a balance between feasible running times and large enough input. *Relative frequency* is calculated by counting the proportion of records in a specified year where the selected word occurs (see also Section 1.4.1). In Figure 5.1, the distribution of the relative frequency is shown. It is apparent from the figure that a large number of words are excluded, while still retaining a significant proportion.

5.2.1 Topic similarity and neighbors

The main idea of the prediction algorithm is to find correlation between the past of the word, the past of similar words (as input), and the future of the word (as output). More specifically, the past is defined simply as the first half of the full time range being analyzed while the future being the second half. For this end, the first necessary tool is a *similarity measure* between words. This similarity measure can be used then to choose similar words, *neighbors*. The average of the first few neighbors can be provided afterwards as input data for the prediction.

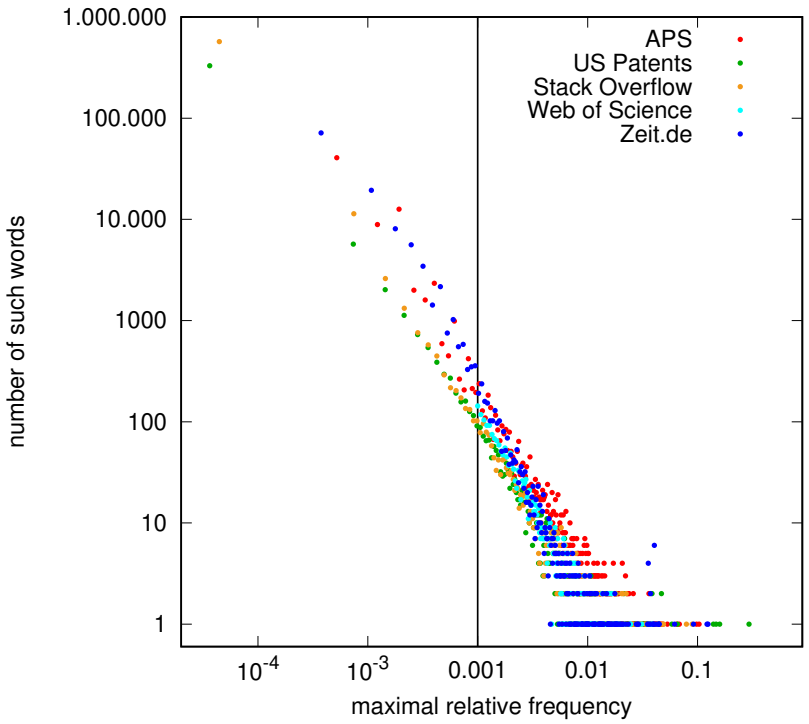


Figure 5.1: *Distribution of word frequencies in the examined data sets.* The vertical black line at 0.001 marks the border between significant and excluded words, which were omitted from further investigations. The relatively large white area below the points, right from the border, show that significant words indeed make up a large proportion of all words. From Table 1.1 it is apparent that they add up to more than 1000 words, seemingly contradicting what is apparent from this figure, where the border crosses the curves around 100. The difference can be explained by adding up all points which occur after the border.

We call a word X to be a *neighbor* of the word Y if the two are mentioned together at least once in the given year (there is at least one record mentioning both). This means that their similarity measure will be larger than 0. Once we have that, we can see if the direction of a selected word significantly differs from the direction of the average of its neighbors. If so, then it can be a sign predicting that the direction of the usage of the selected word will turn around soon. For example, if word X is increasing, but its closest neighbors are decreasing, then we can predict that the word X will turn around as well, soon.

Of course, not every neighbor has the same influence. Furthermore, it would be more effective to choose a smaller set of the neighbors, since then we have much less data to process in order to get the same results. Limiting the neighbors included in the prediction has a theoretical positive side-effect, as well, namely, by disregarding neighbors with a relatively smaller similarity one also filters out potential noise.

The similarity measure should reflect the relation between the two keywords, but at the same time it also should be able to used for ordering. It is defined as:

$$sim(X, Y) = \sum_{t \in T} \frac{freq_{X,Y}(t)}{freq_X(t)}, \tag{5.1}$$

where $freq_X(t)$ is the number of titles (or relative frequency, which results the same fraction) containing the word X at time t, $freq_{X,Y}(t)$ is the number of titles containing both X and Y, and $t \in T$ goes through the whole examined time range. The measure is asymmetric, since it reflects the effect that one word has on the other, and a word that is used often will have a more significant effect on a more rare one than vice versa.

The so-defined similarity measure resembles mostly what we called *OC-similarity* above (see Section 4.2.1), with a different normalization. The reason for this difference is that, while the current normalization is the more natural one, but when comparing different measures, in order to create a fair competition between them, a more sophisticated normalization was necessary to apply (see Section 4.3.2), while, in our case, we can fall-back on the natural choice.

The importance of the neighbors is that they help us to predict whether a word will increase, decrease or stagnate. Presumably, the most similar neighbors are exercising the most influence to a word. In the following, we will verify this theory using exact measurements.

5.2.2 Filtering neighbors by influence

We would like to utilize the neighbors as basis for the prediction of the behavior of a word. For this, it is preferable to consider only the closest ones, while ignoring those that are not so closely related, in order to filter out their effect. This motivates the introduction of a *neighbor threshold*, the number of closest (most similar) neighbors to consider in this average. We determine this value for every data set separately. For this purpose, we sort the N neighbors of a specified word by their similarity, starting with the most similar one (highest similarity measure). For every possible $n = 1 \dots N$ we calculate the distance between the average of the first n neighbors and the average of all N neighbors. These both are a list of yearly frequency values, similar to the topic diagram. The distance between the two is then calculated as a yearly average difference in absolute value:

$$d_w(n) = \frac{1}{|T|} \sum_{t \in T} |freq_{s_w(n)}(t) - freq_{s_w(N)}(t)|, \tag{5.2}$$

where $s_w(n)$ denotes the average of the n closest neighbors of word w:

$$freq_{s_w(n)}(t) = \sum_{i=1}^{n} freq_{u_i}(t), \tag{5.3}$$

where $u_i, i = 1, \dots, n$ are the n closest neighbors of the word w.

Figure 5.2a is an example of $d_w(n)$ in the function of $n = 1 \dots N$. It is apparent that it converges to 0, which was expected, based on the definition of it and the ordering of the neighbors by similarity.

In order to determine the neighbor threshold for a complete data set, for every word w, we calculate its individual threshold n_w, for which $d_w(n_w) \leq 0.01$ stands true, and this is the smallest of such possible n_w values. That is, the minimal neighbors necessary to have almost the same average for every year as if we took all the neighbors into account.

(Here 0.01 is a deliberate, but quite reasonable choice. One would rather take a number when it is apparent that the convergence of $d_w(n)$ to 0 slows down, but this can vary from word to word. It would also be possible to see how the threshold numbers are varying as the function of this yearly average distance limit – for now we leave this open for later possible continuation.)

Afterwards, the "global" neighbor threshold is determined for the data set by examining the distribution of all n_w values for every w word. This is what is shown on Figure

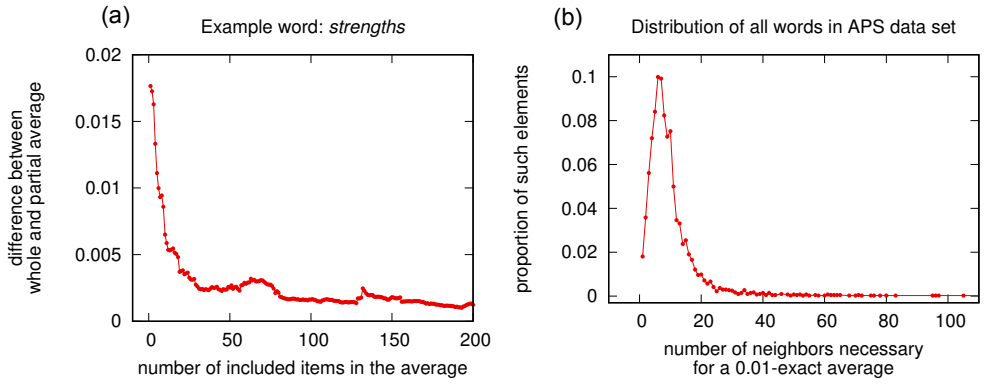


Figure 5.2: *Choosing the threshold value of included neighbors in the analysis.* In order to make the algorithm more efficient, not all neighbors are considered in the prediction input. Rather, the neighbors are ordered according to the distance measure $sim(X, Y)$ introduced above. The first n neighbors are summed and normalized. For every possible $n = 1 \ldots N$, the resulting curves are compared to the last one, for which all neighbors are included. This comparison involves calculating the absolute distance between the current curve and the last one for every year. This process results the function $d_w(n)$, shown on part **(a)** of the plot, which was run for the keyword *"strengths"*, in the APS data set. This word reaches the 0.01 limit for the first time by including the first $n_{strengths} = 6$ neighbors. For every word w, this n_w number is collected. Part **(b)** of the plot contains the probability density function of this n_w number. This PDF has a clear cut: it is apparent that by including the first 30 neighbors, one reaches the 0.01 distance of the final curve for most of the data set (95%, in the present case). This analysis is applied subsequently for all data sets.

5.2b. It turns out to be a concentrated distribution, with most of the words residing in the interval $[0; 30]$. Therefore, we can fix the neighbor threshold for the APS data set to $n_w = 30$, and the same procedure provides all the other values that can be found in Table 1.1, in column "Neighbor limit". By doing so, it is guaranteed that in most of the cases, by taking only the first n_w neighbors, we get essentially the same result as if we took all the words.

5.2.3 Topic codes in different phases

The prediction itself works by using three input variables and producing one output. The three input variables are:

A. Past (first half of time range) of the word.

B. Past of the neighbors of the word (their average).

C. The relation between the two above.

The output variable is D, which corresponds to the second half of time range of the word. It is obtained with the same procedure as A, just its definition applies to the next time frame.

At this point, variable A and B is in the format of a topic diagram, while variable C is not yet exactly defined. The output is the future of the word, which is the second half of the time range of the word, the continuation of variable A. This is also a topic diagram. Above, in Section 1.4.2 we introduced the notion of topic codes, which is the representation that we will use for the actual prediction.

We have already referred to the fact that the outcome of the interaction between topics can often be stagnation (at the end of Section 5.2.1). If the topic itself is increasing, but the similar topics are decreasing, the result might very well be some noisy fluctuation around the current point, since these two forces can extinguish one another. This behavior can be finely described by using $h = 4$. For example, if the topic diagram is described by 012333, it means increasing and then stagnation. On the other hand, 012322 mean increasing and then decreasing. These two patterns coincide when projected by the parameter choice $h = 2$ to 000111. Therefore, from now on we will assume $h = 4$. On the other hand, the subdivision parameter l is subject to further analysis, since it may influence the outcome of the prediction algorithm for the better or worse.

5.2.4 Comparing words with their neighbors

Now we have successfully assigned codes to the input variables A and B of the prediction. Let us now see the remaining variable, C, which describes the relation of these two. As we will see later on from the results of the input variable analysis, in Section 5.4.3, that despite the apparent redundancy, this notion advances the prediction in a palpable manner.

For the sake of comparison of the past topic diagram of the word and the average of its neighbors, we use the sum normalization introduced in Section 1.4.1, 1.(b), since this involves less noise than the other possible option, the min-max normalization. The relation of the two curves are also described by a similar code which was introduced in the section above. The purpose of this code is to show whether a significant draft can be expected from the part of the neighbors. This is why in this case we work with two coordinates only ($h = 2$, using the recently introduced notion), instead of $h = 4$ which is

more useful by a word or an average of several words.

$$c_i = \begin{cases} 1, & \text{if } freq_{s_w}(i) \geq freq_w(i) \\ 0, & \text{otherwise,} \end{cases} \tag{5.4}$$

where s_w stands for the average of the neighbors of word w. The code obtained by putting together these $c_i, i = y_1 \dots y_n$ coordinates is then resembles an increasingly ordered sequence of 0's and 1's if the neighbors are expected to influence w in a positive manner. Conversely, if it resembles a decreasingly ordered sequence, a negative influence can be expected.

5.2.5 Increasing, decreasing and noisy classes

This idea of "orderedness" is also applicable to the two further input variables A and B (see Section 5.2.3 for definition). This application will help us to classify all three inputs, A, B, and C, into one of three classes: increasing, decreasing and noisy (or stagnating). Clearly, if a code described above for A and B is increasing in coordinates, this implies that the corresponding topic diagram is also increasing. This effect can be measured very simply by comparing such a code $c_i, i = y_1 \dots y_n$ to its increasingly ordered variation $\bar{c_i}$. If c_i itself is increasingly ordered, then the two coincides. An appropriate measure of its increasing orderedness is therefore the proportion $p(c_i)$ of coordinates for which $c_i = \bar{c_i}$ stands true, which gives 1 exactly if c_i is ordered already.

If, however, c_i is ordered decreasingly, it has to be compared to its decreasingly ordered variant, $\tilde{c_i}$. It is not enough to use simply $1-p(c_i)$, since being increasingly ordered, decreasingly ordered and unordered are three, well-separated cases that have to be measured by separate means. The function $n(c_i)$ returns the proportion of coordinates i for which $c_i = \tilde{c_i}$ occurs. If for a specific c_i coordinate $p(c_i) \geq n(c_i)$, then it is more increasing than decreasing. For a generalized orderedness measure $g(c_i)$, we can take therefore the larger of the two. We also want to use the sign of the measurement to reflect which is the larger, therefore if $n(c_i) \geq p(c_i)$, then $g(c_i) = -n(c_i)$ by definition. Furthermore, classes which are neither increasing nor decreasing can be detected easily by having a low value for both orderedness functions, therefore, $g(c_i)$ can be defined as:

$$g(c_i) = \begin{cases} 0, & \text{if } p(c_i) \leq 0.5, n(c_i) \leq 0.5 \\ p(c_i), & \text{if } p(c_i) \geq n(c_i), p(c_i) \geq 0.5 \\ -n(c_i), & \text{if } n(c_i) \geq p(c_i), n(c_i) \geq 0.5 \end{cases} \tag{5.5}$$

The classification of all topic diagrams into increasing, decreasing and noisy classes follows simply by taking the *signum* function of $g(c_i)$: return 1 on positive result, -1 on negative, and 0 for 0.

Note that in the $h = 2$ special case $g(c_i) = 0$ is possible only if exactly the half of the coordinates is 0. Otherwise, it is unavoidable for any two orderings of the coordinates to have an overlap for the value (either 0 or 1) which is represented in more than half of the cases. The problem is that even the comparison of two completely unrelated curves could result non-zero classification, which contradicts the intention of the definition (see Figure 5.3). Therefore, it is advisable to adjust the input variable C in order to fulfill this criteria, otherwise it might result misleading classification, often showing completely unordered sequences as ordered. This can be solved simply by shifting the curves $freq_{s_w}(i)$ and $freq_w(i)$ with the median of the difference of the two:

$$c_i = \begin{cases} 1, & \text{if } freq_{s_w}(i) - freq_w(i) \geq Median_i(freq_{s_w}(i) - freq_w(i)) \\ 0, & \text{otherwise,} \end{cases} \tag{5.6}$$

The complete process of generating the three-valued variables from the topic diagrams is summarized on Figure 5.4.

5.3 Methods of prediction

What comes out from the last section is a well-prepared set of variables A, B, C, and D for each word in every data set, where all four variables can take up three values: -1, 0, or +1, and D is the variable to predict using A, B and C. At this point, there are two possible approaches which we will apply in a parallel fashion. The first is to conjecture the possible output based on a given combination of input values, that is, setting the rules manually. The second is to analyze the data and try to extract the rules. This latter is very similar to what is known in data science as *association rule learning* (see [38] for

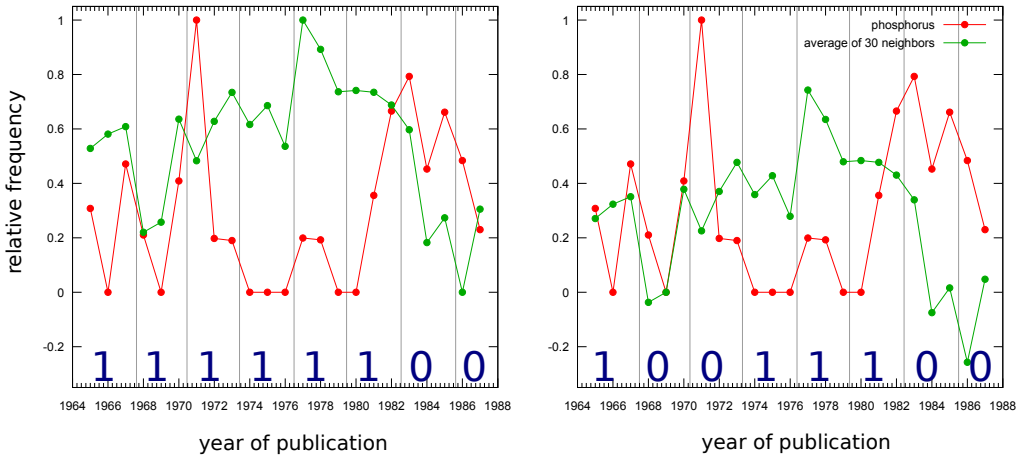


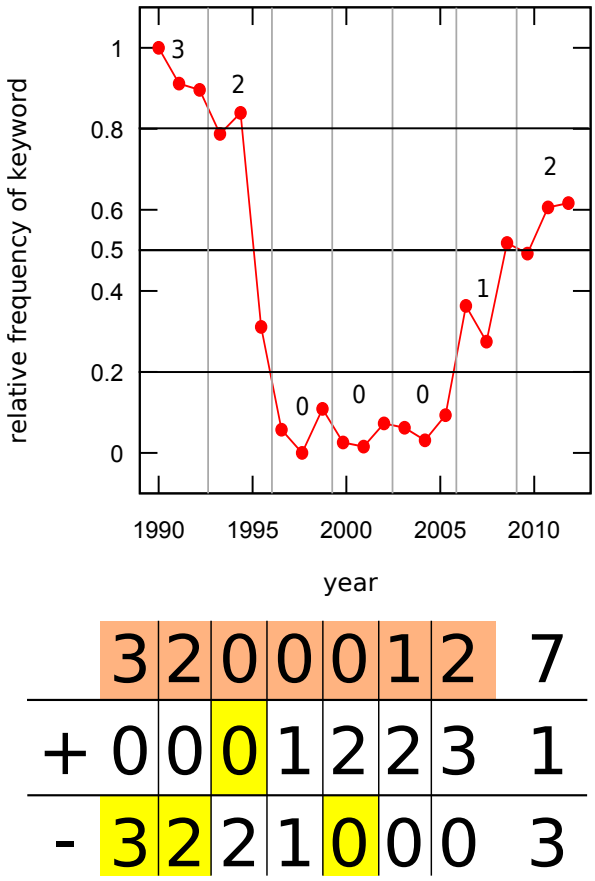
Figure 5.3: *Comparing word traffic with its neighbors.* One of the most important input variables is C, which is calculated as the difference between the topic diagram of the word and that of the average of its neighbors. In this figure the word *phosphorus* was chosen as an example from the APS data set. The figure on the left uses the original values, on the right, the curve of the neighbors are shifted with median of their difference, as described in Section 5.2.5. The 0-1 numbers on the bottom of the figures show the resulting codes. As explained there, the shifting results that exactly the half of the numbers are 0, the other is 1. This allows us to classify the relation of the word to its neighbors as neutral, the neighbors are not expected neither to raise nor to lower the original word (which happens to be the case in this specific example). Without the shift, the assumption would be that the neighbors are decreasing, therefore the word *phosphorus* is also expected to decrease in the following years.

the original model). The combination of the two is to use the second approach to obtain the rules and see whether they fit to what a human agent would expect or not.

5.3.1 Association rules

In order not to restrict the generality of the evaluation, we allow possibility for a prediction with any number of input variables between 0 and 3 (using the input variables A, B and C, introduced in Section 5.2.3). A prediction with 0 variable simply returns the most frequent output for any given input. For example, if in a data set the 60% of the records are increasing (that is, they have $D = +1$), 30% is decreasing ($D = -1$), 10% is noisy ($D = 0$), then the prediction always returns +1. The effectiveness of this "prediction" equals the largest proportion between the possible D values (60%, in our case).

For 1 input variable the procedure is similar: for all possible input variables we choose the largest possible set and the corresponding output value. We apply the same idea for 2



	3	2	0	0	0	1	2	7
+	0	0	0	1	2	2	3	1
-	3	2	2	1	0	0	0	3

Figure 5.4: *Classification process of topic diagrams.* Continuing the example started on Figure 1.3, using parameters $l = 3, h = 4$, the resulting code of the topic diagram is compared to its sorted forms $\bar{c}_i$ and $\tilde{c}_i$. The first row is the original c_i coordinates readable from the top figure, as well. The second row is its increasing order, the third is its decreasing. Overlapping elements are emphasized with yellow background. Based on the proportions of the overlaps, one can read the values $p(c_i) = 1/7$ and $n(c_i) = 3/7$. Since both of them are less than 1/2, consequently, $g(c_i) = 0$. In a prediction setting, the first half of the time range and its second half is analyzed separately (see terms *past* and *future* described in Section 5.2.3), which would result $g(c_i) = -1$ for the first half and $g(c_i) = 1$ for the second half.

and 3 input variables, as well. This results an effectiveness value (proportion of correctly classified keywords) for every possible input variable combination, denoted with 0, for 0 input variable, A, B, and C, for 1 input variable, AB, AC, and BC for 2 input variables and ABC for 3 input variables. In sum, this results 8 values for every data set. The input variables are useful for the prediction if they can successfully increase its effectiveness. In Figure 5.5, all these combinations are shown together, for each data set (for a specific choice of parameters). From this we can see that the prediction is successful in increasing

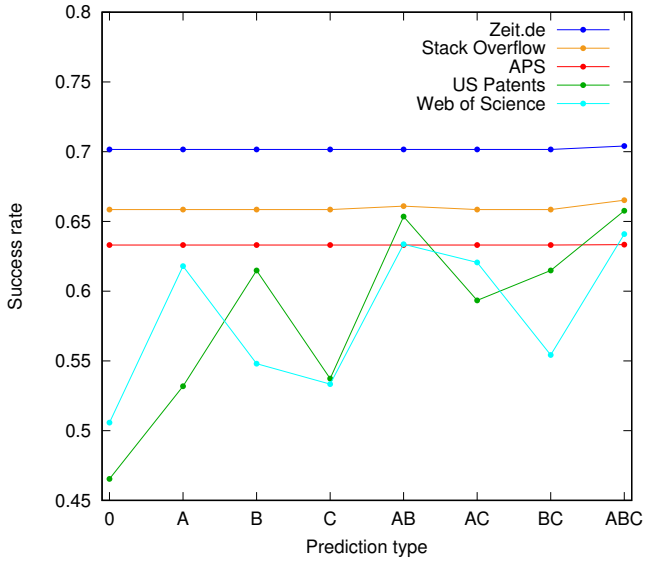


Figure 5.5: *Success rate for different input variables, for fixed parameters:* $ZT = 0.3$, $l = 2$. Prediction type is based on the combination of the following possible input variables: **A:** Past (first half of time range) of a word. **B:** Past of the neighbors of a word (yearly average). **C:** The difference between A and B (shifted with median value, as described in Section 5.2.5). All three input variables are one of the values $\{-1, 0, +1\}$. Apparent from the analysis, but also highly reasonable, that the consideration of input variables is able to increase the success rate only if it was low without them. By considering all three input variables, the success rate becomes remarkably reliable, residing in a small range of possible values. In the selected example, the input variables A and B together are able to produce almost the same success rate as applying them all three. This would imply that the input parameter C (which represents the difference between the past of the word and the past of its neighbors) is unnecessary for the success of the prediction. As shown in Section 5.4.3, this is generally not the case.

the effectivity of a naive prediction with no input if the latter was not successful already.

5.3.2 Expecting stagnation

The effectivity of the prediction can be further improved using a simple and intuitive idea which is reflected on Figure 5.6. In some cases, the output for an association rule is quite straightforward, for example, if the 87% of the cases are increasing. Such is the optimal case of a rule, when one output is strongly greater than the other two. The worst output is when positive, negative and stagnating are producing very similar proportions, close to 33%.

The case between these two extremes deserves more attention. In such a case, either

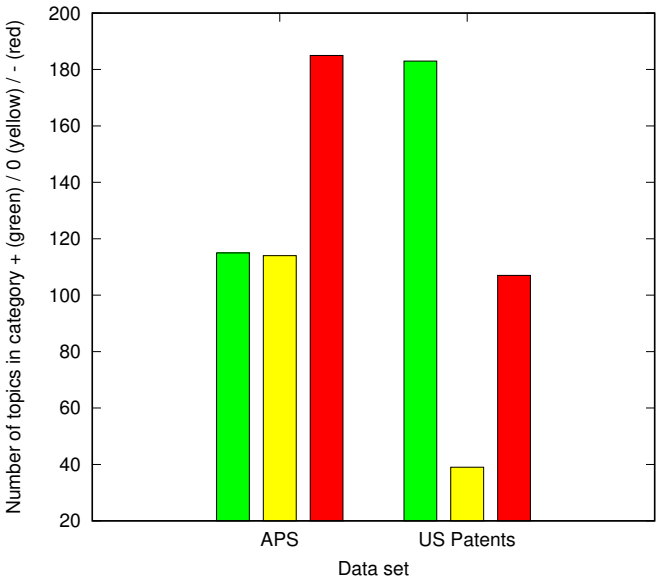


Figure 5.6: *The intuition behind the definition of ZT (zero tolerance).* Two association rules were selected from the data sets APS ($0--$) and Patent ($++-$). In both cases, the red column stands for the decreasing, the green for increasing, the yellow for the number of stagnating cases. From this we see that the rule selected for the Patent data set results a definitely increasing prediction, while regarding APS stagnation and increase are almost the same. ZT is defined as the limit for the maximal height difference of two columns, in relative terms, either between the red and yellow, or between the green and yellow columns. This example uses the latter case. A relatively small ZT would already allow the APS example to produce an *ambiguous output* of increase-or-stagnation, despite the high number of decreasing cases (45%).

1. increasing and stagnating, or 2. decreasing and stagnating output has *almost the same* level. The first case implies that the output is *not* decreasing, while the second case is *not* increasing. This is also a valuable enough piece of an information. Therefore, we extend the definition of an association rule to be able to produce such an output. Such an output is called an *ambiguous output*, since it reflects potentially two different results, and it is predicting what will *not* happen instead of predicting what will happen.

The only question is: how similar should the increasing (or decreasing) and stagnating output be in order to considered *"almost the same"*? In terms of the Figure 5.6, how far the green and yellow columns should be in order to produce such a rule? Obviously, if we allow too big distances in the two values, then it will increase the success of the prediction, at the cost of producing rules that are not so useful (for example, it will predict increase or stagnation, when stagnation is much more probable).

In order to examine the behavior of this distance, we introduce the parameter ZT (*zero tolerance*), which is the possible distance between these two values. ZT is measured with respect to the 50% expected middle value. For example, $ZT = 0.1 = 10\%$ means the proportion of both values should be between 40% and 60% of the total sum. It can be thought of as opening a scissors to width ZT which is placed in the middle, in 50%.

Hence, ZT can take up values from 0 to 0.5. The latter always results uncertain output, that is, *"+1 or 0"* instead of +1, *"-1 or 0"* instead of -1. In the present work, the value of ZT is tested for 5 different values between 0 and 0.5, using a step size of 0.1.

5.4 Results

5.4.1 Comparison with the null model

The first thing that demands verification is whether the input variables A, B, and C have any effect on the success of the prediction D at all. That is, what success rate the prediction achieves in comparison with the null model that simply always predicts the most frequent output. Figure 5.7 shows this effect, for all possible parameters and returns a positive answer. The prediction based on 3 input variables can successfully increase the success rate of all data sets, for certain parameters.

5.4.2 Effect of the input parameters: width and zero tolerance

Since in Figure 5.7 mostly the small points are residing in the important region with high success rate, which implies that the relatively uncertain parameter setting of $ZT = 0.5$ and the sort is really successful in the prediction, one would want to see in more detail, how decreasing ZT would effect the success rate. Furthermore, this figure in and of itself is not able to describe the effect the other parameter, width (l), described in Section 5.2.3. For this purpose we analyze the marginal and joint effects of these two input parameters. This is what is shown in Figure 5.8 and 5.9. The outcome: a low width and zero tolerance value together are able to produce good prediction and useful rules at the same time.

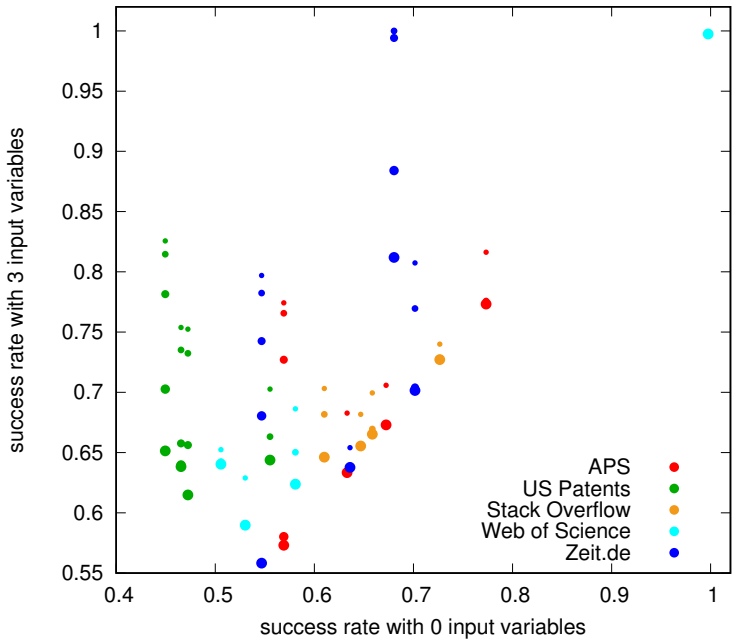


Figure 5.7: *Comparing the prediction with null model.* For 5 data sets, all possible combinations of the parameters: the number of consecutive years, $l = 1 \dots 4$ and the zero tolerance, $ZT \in [0; 0.5]$ (using step size 0.1). The latter is reflected on the figure by the point sizes. Small point sizes correspond to large zero tolerance, which makes less sense, as outlined above in Section 5.3.2. Therefore, large point sizes correspond to the more meaningful prediction contents. Regarding the relation of the number of input variables, we expect that by increasing the number should increase the success rate of the prediction, therefore, the top left part of the scatter plot is the most successful region. A slight success is produced by almost all 5 data sets. Obviously, the region below the $x = y$ line is empty, since this would imply a prediction which works better without variables than with them, which is impossible. Furthermore, note that the data sets APS and Zeit.de produce what looks like a bifurcation, which could be an interesting topic for further research.

5.4.3 Necessity of input variables

Now let us turn to the analysis of the finer details. For every parameter pair l, ZT a detailed figure can be generated, which shows the success rates for all possible variable inputs. See Figure 5.5 for an example.

This type of figure helps to analyze the effect of the individual input variables. Most importantly: are they all necessary? Which of them is the most effective in increasing the success rate of the prediction?

One expects that the more input variables should imply more accurate prediction. In the 92% of all possible parameter choices and data sets this is indeed the case: the

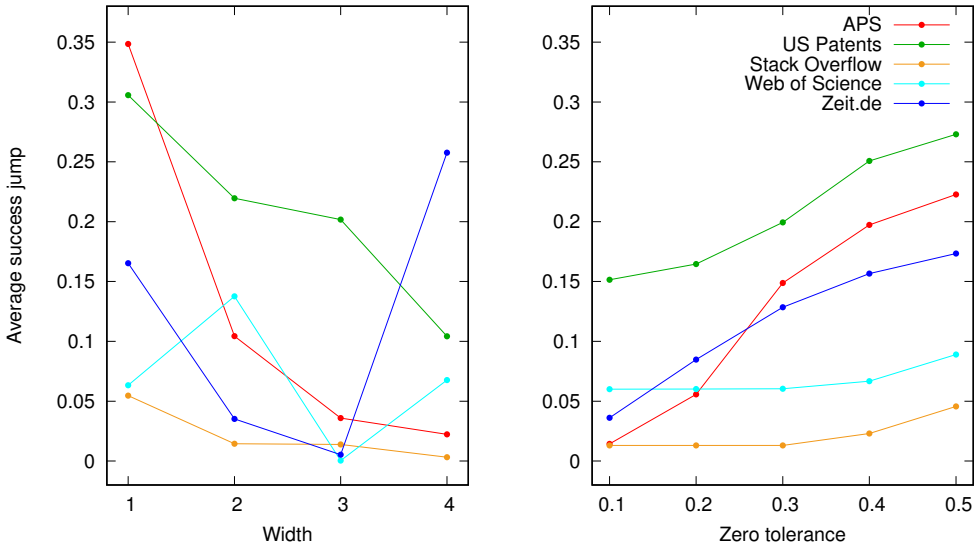


Figure 5.8: *Effect of the input parameters: width (l) and zero tolerance (ZT).* The prediction is based on two input parameters, which are defined above in Section 5.2.3 and 5.3.2, respectively. The dependence of the success on them is shown on this figure, by taking the marginal averages of the other parameter. It is apparent that usually 1 or 2 as a width produces the best results, for almost all data sets. By increasing zero tolerance, the success increases by definition, but only in a slight manner, while at the same time, the information content of the rules produced are decreasing. Even by choosing a relatively low value (like 0.2-0.3) produces useful results for most data sets.

input combination signed with ABC produces the best prediction success rate. The few exceptions occur because the fulfillment of the criteria of zero tolerance (see Section 5.3.2) depends on the way how cases are divided into different groups corresponding to possible combinations of input values. One division might fulfill it and allow zero cases in the output, thereby increasing the success rate, while another division might not, regardless of the number of input variables. In 1 case, this results a prediction that is 11% better than the one considering all input variables A, B and C, in another case, it results 4%, and all other cases are below 2%, so even this exceptional phenomenon is not significant. Therefore, in most cases a prediction that considers all three possible input variables prevails.

The next thing to clarify is the relation amongst the three input variables. Which of them is the most helpful? All of them are important? In Figure 5.10, a statistics is shown about the second most successful input combination. Since all three input variables A, B and C are represented in a significant number of cases, therefore, none of them can be

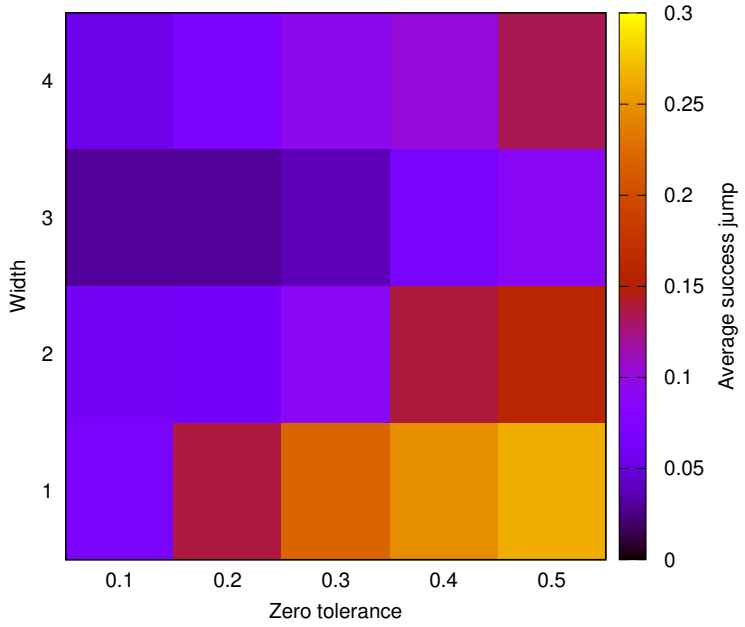


Figure 5.9: *Effect of the input parameters jointly.* This heatmap shows the joint results for the marginals from Figure 5.8. It proves that in order to produce the best results in the prediction, a low width parameter choice is the best, which also works with low zero tolerance. Such a choice is already able to increase the results significantly.

omitted. In Figure 5.5, we saw that the combination AB gets quite close to the optimal success rate. Now we see that this implies by no means that a prediction of ABC input can be substituted with AB input, since one cannot tell in advance, which of the combinations will be the most successful. Note that the two most successful inputs, AB and AC already contain all three input variables.

5.4.4 Analysis of decision rules

As already mentioned, the prediction algorithm was defined to generate the decision rules in a "blind" manner, without considering if they make any sense or not. For example, the rule $+++$ makes much sense (using the abbreviated notion introduced in Section 5.3.2), while the rule $++-$ does not. It is not expected that if the word itself was increasing in the past, as well its neighbors, then it should decrease in the future.

Generally speaking, the input variables determine an "interval" in which the output variable is expected to be. This is an interval in a very slight sense of the word, since the input variables can take up the values $\{-1, 0, +1\}$ only, while the output variables are a

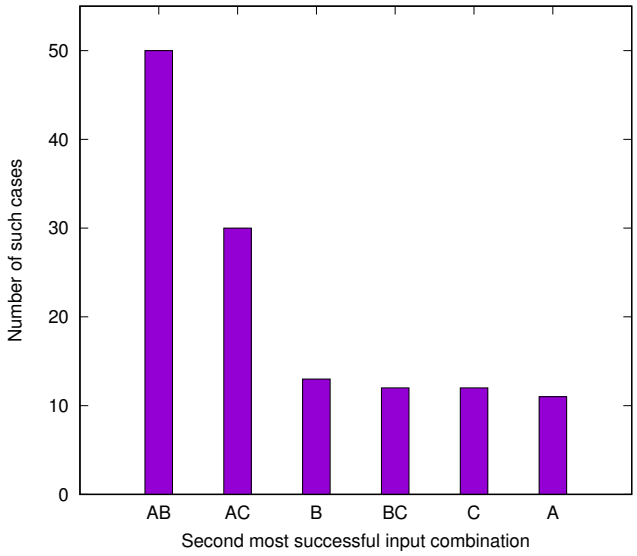


Figure 5.10: *Distribution of the second best input variable combination.* Generally, the best input variable combination that has the highest success rate is assumed to be the full choice, ABC. Often, the second best option reaches a level that is near to it. The distribution of the second best combination is variable, not concentrated to a certain set of input variables. This proves that not one of the three input variables A, B and C can be omitted from the prediction, all of them form an integral part of the success of the joint prediction. The cases were run here, as well as in all other figures, for all 5 data sets, $l = 1 \ldots 4$, $ZT \in [0; 0.5]$ (with 0.1 step sizes, 100 cases altogether).

bit more flexible. As described in Section 5.3.2, they can take up more output values. One of them – let us sign it with p, which stands for *positive* – is either +1 or 0. The other is m, which can be either -1 or 0.

Now, the set of input variables determine their interval based on their smallest and largest value, and they allow the output to be everything in between. That is, if there is both $+$ and $-$ in the input, then the output can be anything. If there is only $+$ and 0, then it cannot be $-$, neither m, since both of them reside outside of the interval defined by the input.

Using this definition we can decide for every rule if it is intuitive or not. In order to see the proportion of the intuitive rules for every data set, we need to define a *stable set of rules* for each of them. This is necessary because different parameter choices result different rules, as we have seen already. For that end, we run the prediction algorithm for every possible parameter and assemble a *histogram* of all rules produced for each individual data set, which counts the number of occurrences of every rule. Since we saw

in the last section the necessity of all three input variables, therefore now it suffices to analyze the intuitivity of the rules pertaining to the full input combination, ABC.

This will inevitably contain contradicting rules, for example, $++0+$ and $++0-$. They contradict in the sense that they would assign different outputs for the same input. (Recall that in the abbreviated form of the rules introduced in Section 5.3.2 the output variable appears directly after the input variables.) Therefore, in order to narrow the set of rules we start from the most common, which is assumed to be the most stable, which was produced most frequently, for the different parameter choices, and going in decreasing order, we add each rule to the set of stable rules, until we get to the first contradiction. Then we throw away both contradicting rules, provided that they have an equal number of occurrences, since then there is no reason to favor either of them, and what we have is the stable set of rules for the selected data set. By this procedure, we defined the set of stable rules as being the most frequent and consistent (not self-contradictory) set of association rules.

The number of stable rules and proportion of intuitive rules for every data set can be seen in Figure 5.11. The results imply that most of the rules produced by the algorithm is intuitive, although, not predictable without the assistance of the computer, since we still have multiple valid choices within the interval defined by the input variables.

5.5 Conclusion

In this paper we tested a prediction algorithm of the usage of topic keywords in titles on 5 data sets of different types, which is based on the directionality (increasing, decreasing or noisy) of the keyword and its most similar neighbors. The prediction algorithm produces rules based on the directionality of the word, the average of its neighbors, and the relation of the two.

We showed that none of these three components are omittable for a successful prediction. Furthermore, the success rate was increased by introducing uncertainty in the result in a natural manner, by accepting both increasing/decreasing and noisy result, and the measure of this necessary uncertainty was tested. At the same time, we succeeded to show and numerically measure the limitations of such a prediction, which is based on an extremely simplified input, based on the similar topics. The results can be seen online on `http://topinav.elte.hu/pred180/` (the source code is available on

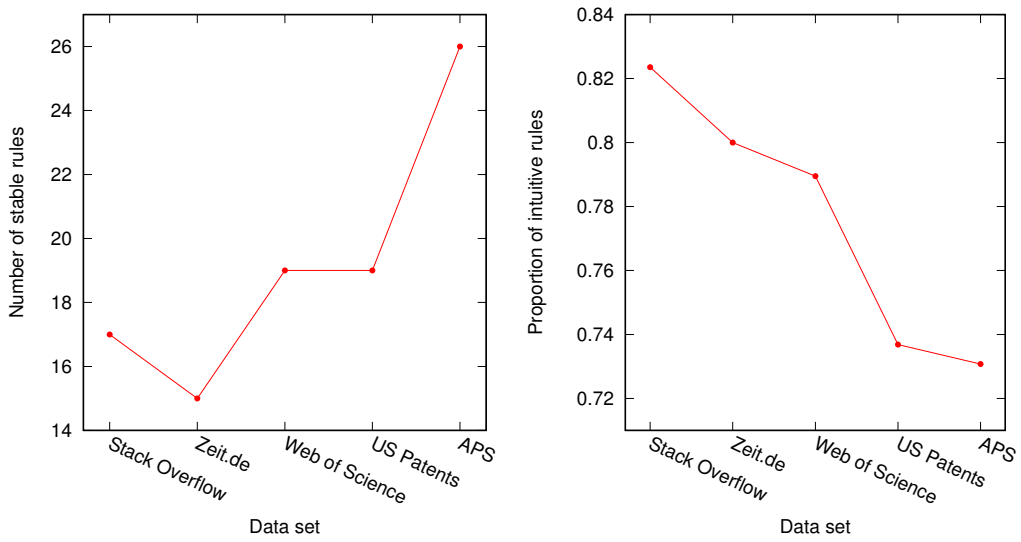


Figure 5.11: *Stability and intuitivity of the decision rules.* The prediction algorithm works by producing a number of decision rules for every possible data set and parameter choice. Within a data set, the most stable rules are those that occur in almost every possible parameter choice. In order to find a threshold value for their number of occurrence, we stop at the first contradicting pair of rules. This results the left part of the figure. The right part is the proportion of intuitive rules amongst the stable ones. A rule is *intuitive* if its output is not outside of the interval defined by its input variables, as described in Section 5.4.4. The figure shows that the overwhelming majority of the produced stable rules are intuitive.

`https://github.com/binyominzeev/pred180`).

Among the further research directions of this topic is more detailed analysis of the dependence of the success rate on the parameters, cross-validation of the prediction algorithm, verifying the apparent bifurcation effect of the success rate in Figure 5.7, and testing whether the usage of classical topic models are able to significantly improve the quality of the prediction.

Acknowledgments

In the first place, I would like to thank the Creator, Who grants wisdom to every human, and has guided me in all my efforts, often, against my will, and endowed me both with useful ideas as well as with the fascination, which is so necessary to do anything meaningful in the life. Without His help and direct intervention, I would give up way before reaching to the final goal.

I thank for my advisor, Dr. Illés Farkas, who besides investing so much time and energy in all these efforts, has demonstrated an exemplary amount of patience, which would really be a deep lesson for my whole life if I could preserve these memories well. His trademark was during all these years not only to give guidance in the area of research, rather, also prepare a young and immature fellow to the challenges of life. He never grew weary of considering always the best option for his student, under every circumstance, and really worked hard in trying to figure out the possible pitfalls and guarding me from them.

I thank to Prof. Tamás Vicsek for the outstanding opportunity that I could be part of his research group of exceptionally high quality, and to see from such closeness what it means to work and live as a scientist. I thank Dr. Péter Pollner, Dr. Gergely Palla, Dr. Imre Derényi, and everybody else in the institute for all the good advice and helpful comments, and enduring all the difficulties and oddities, which came together with me.

I thank for all governmental support which made me possible to start my research after graduation, before becoming a PhD student, and also for providing the opportunity to visit conferences, which also became a lasting and thoughtful experience.

I thank Dr. Pál Hegedűs (CEU) for his help in personal guidance as well as professional issues. I thank my father and my brother, Iván, who both have invested disproportional amount and worry in the present work. I thank for all the people who have supported me during this 9 years since I was involved in this effort, especially at the very

end, thanks for Rabbi David Keleti for the support, making me able to learn both religious and secular studies, in Hungary and in Israel, and thanks for all the people involved in the Rabbinerseminar zu Berlin, for providing the same in Germany.

I thank for my grandmother, should the peace be with her, for allowing me (later: us) to live in her apartment, and thereby significantly supporting my work and living. I thank my mother, my brothers Iván and Balázs, my wife, my son, and all my family for the constant support and encouragement.

Bibliography

[1] United States Patent and Trademark Office Google. *United States Patent and Trademark Office Bulk Downloads*. https://www.google.com/googlebooks/uspto.html. 2015.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.Jan (2003), pp. 993–1022.

[3] David M. Blei and John D. Lafferty. "Dynamic Topic Models". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 113–120. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143859. URL: http://doi.acm.org/10.1145/1143844.1143859.

[4] André Gohr et al. "Topic Evolution in a Stream of Documents". In: *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), pp. 859–870. DOI: 10.1137/1.9781611972795.74. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9781611972795.74. URL: http://epubs.siam.org/doi/abs/10.1137/1.9781611972795.74.

[5] Hoang Anh Dau et al. *The UCR Time Series Classification Archive*. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/. Oct. 2018.

[6] Hoang Anh Dau et al. *The UCR Time Series Archive*. 2018. arXiv: 1810.07758 [cs.LG].

[7] Ádám Szántó-Várnagy et al. "Scientometrics: untangling the topics". In: *National Science Review* 1.3 (2014), p. 343. DOI: 10.1093/nsr/nwu027. URL: https://academic.oup.com/nsr/article-lookup/doi/10.1093/nsr/nwu027.

[8] Emilio Delgado Lopez-Cozar, Nicolas Robinson-Garcia, and Daniel Torres-Salinas. "Manipulating Google Scholar Citations and Google Scholar Metrics: simple, easy and tempting". In: (2012). eprint: arXiv:1212.0638.

[9] P. Erdős and A Rényi. "On the Evolution of Random Graphs". In: *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*. 1960, pp. 17–61.

[10] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512. ISSN: 0036-8075. DOI: 10.1126/science.286.5439.509. eprint: http://science.sciencemag.org/content/286/5439/509.full.pdf. URL: http://science.sciencemag.org/content/286/5439/509.

[11] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas". In: *Phys. Rev.* 136 (3B Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864. URL: https://link.aps.org/doi/10.1103/PhysRev.136.B864.

[12] W. Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". In: *Phys. Rev.* 140 (4A Nov. 1965), A1133–A1138. DOI: 10.1103/PhysRev.140.A1133. URL: https://link.aps.org/doi/10.1103/PhysRev.140.A1133.

[13] John A. Hertz. "Quantum critical phenomena". In: *Phys. Rev. B* 14 (3 Aug. 1976), pp. 1165–1184. DOI: 10.1103/PhysRevB.14.1165. URL: https://link.aps.org/doi/10.1103/PhysRevB.14.1165.

[14] A. J. Millis. "Effect of a nonzero temperature on quantum critical points in itinerant fermion systems". In: *Phys. Rev. B* 48 (10 Sept. 1993), pp. 7183–7196. DOI: 10.1103/PhysRevB.48.7183. URL: https://link.aps.org/doi/10.1103/PhysRevB.48.7183.

[15] Stephen L. Adler. "Axial-Vector Vertex in Spinor Electrodynamics". In: *Phys. Rev.* 177 (5 Jan. 1969), pp. 2426–2438. DOI: 10.1103/PhysRev.177.2426. URL: https://link.aps.org/doi/10.1103/PhysRev.177.2426.

[16] J. Steinberger. "On the Use of Subtraction Fields and the Lifetimes of Some Types of Meson Decay". In: *Phys. Rev.* 76 (8 Oct. 1949), pp. 1180–1186. DOI: 10.1103/

PhysRev.76.1180. URL: https://link.aps.org/doi/10.1103/PhysRev.76.1180.

[17] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks". In: *Rev. Mod. Phys.* 74 (1 Jan. 2002), pp. 47–97. DOI: 10.1103/RevModPhys.74.47. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.47.

[18] V. Alan Kostelecký and Matthew Mewes. "Cosmological Constraints on Lorentz Violation in Electrodynamics". In: *Phys. Rev. Lett.* 87 (25 Nov. 2001), p. 251304. DOI: 10.1103/PhysRevLett.87.251304. URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.251304.

[19] Luca Gammaitoni et al. "Stochastic resonance". In: *Rev. Mod. Phys.* 70 (1 Jan. 1998), pp. 223–287. DOI: 10.1103/RevModPhys.70.223. URL: https://link.aps.org/doi/10.1103/RevModPhys.70.223.

[20] D. Colladay and V. Alan Kostelecký. "Lorentz-violating extension of the standard model". In: *Phys. Rev. D* 58 (11 Oct. 1998), p. 116002. DOI: 10.1103/PhysRevD.58.116002. URL: https://link.aps.org/doi/10.1103/PhysRevD.58.116002.

[21] David R. Nelson and V. M. Vinokur. "Boson localization and correlated pinning of superconducting vortex arrays". In: *Phys. Rev. B* 48 (17 Nov. 1993), pp. 13060–13097. DOI: 10.1103/PhysRevB.48.13060. URL: https://link.aps.org/doi/10.1103/PhysRevB.48.13060.

[22] J. Wunderlich et al. "Experimental Observation of the Spin-Hall Effect in a Two-Dimensional Spin-Orbit Coupled Semiconductor System". In: *Phys. Rev. Lett.* 94 (4 Feb. 2005), p. 047204. DOI: 10.1103/PhysRevLett.94.047204. URL: https://link.aps.org/doi/10.1103/PhysRevLett.94.047204.

[23] B. W. Harris, F. Chen, and U. Mohideen. "Precision measurement of the Casimir force using gold surfaces". In: *Phys. Rev. A* 62 (5 Oct. 2000), p. 052109. DOI: 10.1103/PhysRevA.62.052109. URL: https://link.aps.org/doi/10.1103/PhysRevA.62.052109.

[24] Michal Brhlik, Gerald J. Good, and G. L. Kane. "Electric dipole moments do not require the CP-violating phases of supersymmetry to be small". In: *Phys. Rev. D* 59

(11 Apr. 1999), p. 115004. DOI: 10.1103/PhysRevD.59.115004. URL: https://link.aps.org/doi/10.1103/PhysRevD.59.115004.

[25] MEJ Newman. "Power laws, Pareto distributions and Zipf's law". In: *Contemporary Physics* 46.5 (2005), pp. 323–351. DOI: 10.1080/00107510500052444. eprint: https://doi.org/10.1080/00107510500052444. URL: https://doi.org/10.1080/00107510500052444.

[26] J. Starlinger et al. "Layer Decomposition: An Effective Structure-Based Approach for Scientific Workflow Similarity". In: *2014 IEEE 10th International Conference on e-Science*. Vol. 1. Oct. 2014, pp. 169–176. DOI: 10.1109/eScience.2014.19.

[27] Emden R. Gansner and Stephen C. North. "An open graph visualization system and its applications to software engineering". In: *Software: Practice and Experience* 30.11 (2000), pp. 1203–1233. ISSN: 1097-024X. URL: http://www.graphviz.org/.

[28] Sulieman Bani-Ahmad, Ali Cakmak, and Abdullah Al-Hamdani. "Evaluating Publication Similarity Measures". In: *IEEE Data Engineering Bulletin* 28 (Dec. 2005), pp. 21–28.

[29] Joan Serra and Josep Ll Arcos. "An empirical evaluation of similarity measures for time series classification". In: *Knowledge-Based Systems* 67 (Sept. 2014), pp. 305–314.

[30] Benjamin Markines et al. "Evaluating similarity measures for emergent semantics of social tagging". In: *Proceedings of the 18th international conference on World wide web*. ACM. 2009, pp. 641–650.

[31] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. "Evaluating similarity measures: a large-scale study in the orkut social network". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM. 2005, pp. 678–684.

[32] Saikat Bagchi. "Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout". In: *Procedia Computer Science* 50 (2015). Big Data, Cloud and Computing Challenges, pp. 229–234. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.04.055. URL: http://www.sciencedirect.com/science/article/pii/S1877050915005566.

[33] Gabor Csardi and Tamas Nepusz. “The igraph software package for complex network research”. In: *InterJournal* Complex Systems (2006), p. 1695. URL: http://igraph.org.

[34] Gustaf Bellstam, Sanjai Bhagat, and J Anthony Cookson. “Innovation in Mature Firms: A Text-Based Analysis”. In: *SSRN* (2017). DOI: https://dx.doi.org/10.2139/ssrn.2803232.

[35] Ádám Szántó-Várnagy and Illés J. Farkas. “Forecasting turning trends in knowledge networks”. In: *Physica A: Statistical Mechanics and its Applications* 507 (2018), pp. 110–122. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2018.05.055. URL: http://www.sciencedirect.com/science/article/pii/S0378437118305995.

[36] Dashun Wang, Chaoming Song, and Albert-László Barabási. “Quantifying Long-Term Scientific Impact”. In: *Science* 342.6154 (2013), pp. 127–132. ISSN: 0036-8075. DOI: 10.1126/science.1237825. eprint: http://science.sciencemag.org/content/342/6154/127.full.pdf. URL: http://science.sciencemag.org/content/342/6154/127.

[37] Alexander Maedche and Steffen Staab. “Measuring Similarity Between Ontologies”. In: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. EKAW ’02. London, UK, UK: Springer-Verlag, 2002, pp. 251–263. ISBN: 3-540-44268-5. URL: http://dl.acm.org/citation.cfm?id=645362.650859.

[38] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. “Mining Association Rules Between Sets of Items in Large Databases”. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’93. Washington, D.C., USA: ACM, 1993, pp. 207–216. ISBN: 0-89791-592-5. DOI: 10.1145/170035.170072.

Summary

We have presented various methods to analyze the interactions of topics and records in real world data sets, and identifying their outstanding elements ("centers"), based on multiple ideas.

The first idea was to find articles boosting one another, based on the prime example of the publication presenting Barabási-Albert model, which was boosting the number of citations of the Erdős-Rényi model (Chapter 2). Using the data set of publications of the American Physical Society between 1965 and 2009, it was demonstrated that a boosting effect of the measure of the prime example does not have a matching pair. Nevertheless, three different numerical measures (boosting value, time distance, and citation count) were applied in a joint manner, in order to identify boosting effect of a lower measure, which successfully pointed out connected and important pieces of research articles and their network.

In Chapter 3, we turned our attention from article-article interactions to article-topic interactions, and a similar core concept (*burst*) was evaluated. The starting point of this research was the topic diagram, which shows the evaluation of the usage of a certain word in time, and its quick jumps. After eliminating factors which create noise, we have successfully presented a method which identifies such jumps (bursts). Subsequently, an algorithm identifying a set of articles responsible for the specific burst was presented. This latter step used the underlying network structure, which offers a unique outlook on a topic, which is available publicly, through an online interface, developed for that purpose (`http://topinav.elte.hu/burst`).

In Chapter 4, similarity measures of topics were evaluated and compared on different data sets. Three concepts (text-based, network-based and time-based), realized in four different measures were analyzed using four large data sets. Through our results, one can obtain a way to choose a similarity measure for a later research or business application

which fits mostly the data set. Furthermore, the general method of normalization leaves an open possibility to use a similar comparison to any further similarity measures.

In Chapter 5, a prediction of the topic diagrams was presented and its success was evaluated. This is based on the nearest neighbors, using the structure of the underlying network, and a simple classification of the possible future of the topic (increasing, decreasing, stagnating). The necessity of multiple input variables was effectively demonstrated, and the prediction showed a success of 60-70%. This experiment also has an online available, browsable interface at (`http://topinav.elte.hu/pred180`).

In summary, the present work has offered a number of useful and computationally effective tools for evaluating massive amount of data, as well as visual demonstrations which are aimed to guide a human agent, when confronted with a data analysis application.

Összefoglaló

Valós adatsorokra alkalmazható különböző módszereket mutattunk be, melyek a témák és rekordok kölcsönhatásait vizsgálják, valamint kiemelkedő elemeket azonosítanak be, különféle megközelítések segítségével.

Az első ezek közül a cikk-cikk kölcsönhatások vizsgálata, melynek alapötletét az adta, ahogyan a Barabási-Albert modellről szóló publikáció szignifikáns pozitív hatással volt az általa hivatkozott Erdős-Rényi modellt bemutató publikációra (2. fejezet). Ehhez az American Physical Society 1965 és 2009 közötti adatsorát használtuk, és megmutattuk, hogy ebben az adatsorban nem található ilyen mértékű effektus. Mindezzel együtt, három különböző mérték (felpörgetés, időeltérés, hivatkozottság) együttes alkalmazása segítségével sikerült beazonosítani csekélyebb mértékű hatást, mely által fontos, összefüggő publikációk csoportja, valamint azok hálózata került felfedezésre.

A 3. fejezetben a cikk-cikk hatások vizsgálatát a cikk-téma hatások váltották fel. Ebben a fejezetben is hasonló felpörgetési mértékegységet vezettünk be (*burst*). Ez a vizsgálat a téma gyakorisági diagramjából indult ki, amiről egy adott szó említésének időbeli fejlődése olvasható le, és annak jelentős ugrásai. A zajkeltő hatások kiszűrése után, sikeresen bemutattunk egy módszert, ami az ilyen ugrásokat beazonosítja. Ezután egy újabb algoritmus került bemutatásra, ami beazonosítja az ugráshoz kapcsolható cikkek halmazát. Ez a lépés felhasználta a mögöttes hálózat struktúráját is, mely egy online, nyilvános felületen keresztül vizsgálható, és új rálátást biztosít az adott témára (`http://topinav.elte.hu/burst`).

A 4. fejezetben témák különféle hasonlósági mértékegységeit értékeltük ki és hasonlítottuk össze, többféle adatsoron. Háromféle megközelítés (szöveg-, hálózat-, és időfejlődés-alapú), négyféle mérték került vizsgálatra, négy adatsoron. A kutatás eredményeként kapott módszer alkalmas arra, hogy tetszőleges adatsorhoz segítsen kiválasztani a lehetőségek közül a lehető leghatékonyabb hasonlósági mértékegységet. Ezenkívül,

a normalizálás során bemutatott általános módszer lehetővé teszi, hogy további, tetszőleges hasonlósági mértékegységeket is össze lehessen hasonlítani a későbbiekben.

A 5. fejezetben témák időfejlődési diagramjainak tendenciájának jóslásásra mutattunk be egy módszert és értékeltük ki annak sikerét. Ez a módszer a legközelebbi szomszédokkal való hasonlóságon alapszik, figyelembe veszi az elemeket összekötő hálózatot, és egy egyszerű osztályozást használ a téma várható tendenciájának predikciójára (növő, csökkenő, stagnáló). A vizsgálat során igazoltuk a többféle bemeneti változó szükségességét, és a jóslás 60-70%-os sikerrátát produkált. Ehhez a kísérlethez is tartozik egy online hozzáférhető, böngészhető felület a `http://topinav.elte.hu/pred180` címen.

Összességében, jelen munka számos hasznos és számításigényét tekintve hatékony eszközt kínál fel, melyek alkalmasak nagy mennyiségű adatsor elemzésére, valamint az emberi szemlélő számára is kézzelfoghatóvá tevő vizualizációkat, amik jelentős segítséget nyújtanak az adatok elemzése során.