DHQ: Digital Humanities Quarterly
2021
Volume 15 Number 1

Using an Advanced Text Index Structure for Corpus Exploration in Digital Humanities

Tobias Englmeier <englmeier_at_cis_dot_uni-muenchen_dot_de>, CIS, Ludwig-Maximilians University, Munich, Germany
Marco Büchler <mbuechler_at_etrap_dot_eu>, Institute of Computer Science, University of Göttingen, Göttingen, Germany
Stefan Gerdjikov <st_gerdjikov_at_abv_dot_bg>, FMI, University of Sofia "St. Kliment Ohridski", Sofia, Bulgaria
Klaus U. Schulz <schulz_at_cis_dot_uni-muenchen_dot_de>, CIS, Ludwig-Maximilians University, Munich, Germany

Abstract

With suitable index structures many corpus exploration tasks can be solved in an efficient way without rescanning the text repository in an online manner. In this paper we show that symmetric compacted directed acyclic word graphs (SCDAWGs) - a refinement of suffix trees - offer an ideal basis for corpus exploration, helping to answer many of the questions raised in DH research in an elegant way. From a simplified point of view, the advantages of SCDAWGs rely on two properties. First, needing linear computation time, the index offers a joint view on the similarities (in terms of common substrings) and differences between all text. Second, structural regularities of the index help to mine interesting portions of texts (such as phrases and concept names) and their relationship in a language independent way without using prior linguistic knowledge. As a demonstration of the power of these principles we look at text alignment, text reuse in distinct texts or between distinct authors, automated detection of concepts, temporal distribution of phrases in diachronic corpora, and related problems.

Introduction

A text/corpus index is a kind of table that, given a string w, stores the positions of all occurrences of w in the given text/corpus. The computation of the index is a preprocessing step to be applied only once. Corpus index structures considerably simplify corpus analysis since they help to avoid rescanning the complete texts for each task. The index helps to directly answer many interesting questions, hence index based methods in general are much faster and more elegant. In this paper we look at an advanced corpus index structure and explain its potential use in Digital Humanities. This index, called symmetric compacted directed acyclid word graph (SCDAWG), is used to represent a collection of texts, each text is considered as a flat sequence of symbols [Inenaga et al. 2005]. SCDAWGs can be considered as a refinement of suffix trees [Weiner 1973], [McCreight 1976], [Ukkonen 1995], [Gusfield 1997]. The SCDAWG for a given text collection, as the suffix tree, can be computed in time and space linear in the size of the corpus. As a first advantage, the SCDAWG of a corpus is much smaller than the corresponding suffix tree. Even more importantly, SCDAWGs have two special features that make them very attractive for various corpus exploration tasks. 1

Finding co-occurrences. Using the SCDAWG it is simple to find all co-occurrences of the same portions of text in two, several, or all texts of the corpus. The index enables a joint search in all texts, there is no need for comparing all pairs of texts individually. This joint search, which leads to all co-occurrences, can be realized using a single scan of the index. The importance of this feature in our context should be obvious: research in Digital Humanities is often concentrated on the problem of finding phrases and text portions that occur in distinct (two, several, or all) texts of a collection. For all these problems the use of SCDAWGs offers an elegant solution. As a demonstration we look at alignment of two or several texts, detection of text reuses in distinct texts, and detection of text reuses between distinct authors. 2

Mining interesting concepts and their relationship. The nodes of the SCDAWG represent portions of text (sequences of symbols, infixes). In general, the number of all distinct infixes of a corpus C is quadratic in size $|C|$ of the 3 corpus. In contrast, the number of nodes of the SCDAWG is always linear in $|C|$. Yet, in a sense to be explained below, it contains all infixes that are interesting from a linguistic point of view.[1] For example, names, concepts and phrases etc. in general correspond to nodes of the SCDAWG. Compared to suffix trees, SCDAWGs yield a much smaller set of nodes/infixes with this property. Using structural regularities of the index, the set of all nodes can be even further filtered to approximate the subset of linguistically relevant infixes. Also hierarchical relationships and dependencies between these concepts and phrases can be detected. From the perspective of Digital Humanities, these mining techniques are relevant since they are completely language independent and do not use any kind of prior linguistic knowledge. They point to interesting concepts and bricks of text, the index helps to study their use, structure and relationship.

In this paper we first formally introduce SCDAWGs in Section 2. We compare three related index structures, SCDAWGs, suffix trees and DAWGs (directed acyclic graphs [Blumer et al. 1987]), compare the sizes of these index structures and explain the advantages of the SCDAWG for corpus exploration tasks as those mentioned above. In Section 3 we give a brief overview of the corpora used for the experiments described in the paper. In Section 4 we show how SCDAWGs can be utilized to solve tasks in the fields of text alignment and text reuse detection [Büchler et al. 2012]. For these problems, the full set of nodes of the SCDAWG is used. In Section 5 we sketch the afore-mentioned filtering of nodes/infixes based on structural regularities (in Gerdjikov and Schulz [2016] the filtering method is described in full detail). Using the example corpora we illustrate how concepts and their relationships can be mined in a language independent way. In Section 6 we look at extensions of the SCDAWG index, adding metadata information (e.g., on authors, temporal periods) from the texts. Examples from our corpora show how to detect text reuses, e.g., between distinct authors and to reveal the temporal flow of phrases across texts/authors. Before we come to a short conclusion, Section 7 provides loose ends for future research, indicating how SCDAWGs might help to treat diachronic language variation and various text classification tasks. 4

SCDAWGs as a corpus index structure

In the introduction we explained the general benefits that can be obtained for corpus analysis when using an index structure. If we are only interested in the distribution and occurrences of single words, a simple index structure is sufficient that represents the corpus as a “bag of words”. Usually an “inverted file” then stores for each word w occurring in the corpus the texts and positions where w occurs. However, corpus analysis in Digital Humanities and other fields is often focussed on other pieces of text: re-used portions of text, names, terminological and other multi-word expressions, phrases that express semantic relationships, syllables, morphems, to name a few. In this situation advanced index structures are preferable that give direct access to the occurrences of arbitrary infixes[2]. This explains why corpus analysis tools in Digital Humanities or Computational Biology are often based on the latter, more powerful type of index structures. Among the latter type of index structure, directed acyclic word graphs (DAWGs) and suffix trees have often been used for research, but refinements have not received proper attention. In this paper we describe compacted directed acyclic word graphs (CDAWGs) and a symmetric variant (SCDAWGs) and argue that these index structures are preferable for many corpus analysis tasks. 5

The common ideas behind DAWGS, suffix trees, CDAWGs and SCDAWGs are summarized in the following way. 6

- Each index represents a graph, nodes representing specific infixes of the corpus.
- Each infix of the corpus is represented at most once in the index.
- The total size of the index structure is linear in the size of the corpus.
- Given any string v we may use the index to decide in time $O(|v|)$ if v is an infix of the corpus.[3] It is also possible to use the index “as a guide” for finding all texts and positions where v occurs.

The nodes of a DAWG represent the infixes v that are “left-closed” in the sense that there does *not* exist a unique symbol σ directly in front of each occurrence of v in the corpus. The nodes of a suffix tree represent the infixes v that are “right-closed” in the sense that there does *not* exist a unique symbol σ directly after each occurrence of v in the corpus. The CDAWG and the SCDAWG for a corpus have the same set of nodes representing the infixes v that are “left- and right-closed” at the same time. Edges of a suffix tree, DAWG, or CDAWG represent extensions of infixes in reading order (i.e., to the right). SCDAWGs have two type of edges, respectively representing right extensions in standard reading order and left-extensions in left-to-right order. **Example 2.1** As an example, consider the corpus with the two toy “texts”:

- `op 1 in A`
- `op 2 in B`

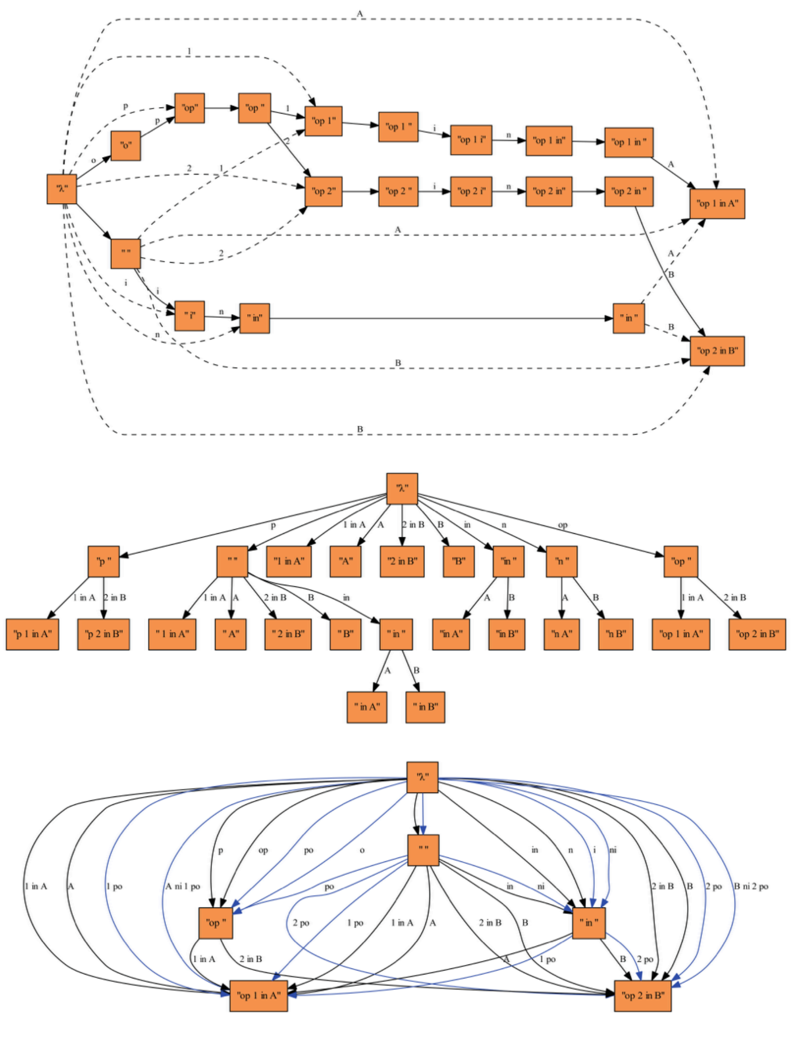


Figure 1. DAWG, suffix tree, and symmetric compact DAWG (SCDAWG) for the corpus with two toy texts `op 1 in A` and `op 2 in B`.

Infixes `n` and `in` are *not* left-closed since we find the symbol `i` directly in front of each occurrence of `n` and the blank directly in front of each occurence of `in` in the corpus. In contrast, infixes `in` , `i` and `op 1` are left-closed. Each prefix of each text is left-closed, the corpus contains 20 left-closed infixes, its DAWG has 20 nodes (cf. Figure 1). Infixes `o` and `op` are *not* right-closed since we find the symbol `p` directly after each occurrence of `o` and the blank directly after each occurence of `op` in the corpus. In contrast, infixes `op` and `p`, `1 in A` are right-closed. Each suffix of each text is right-closed, the corpus contains 25 right-closed infixes, accordingly its suffix tree has 25 nodes (cf. Figure 1). The only infixes that are left- and right-closed are the empty string, the blank, the strings `op` and `in` and the two texts. Hence the (S)CDAWG only has 6 nodes (cf. Figure 1). Lines/transitions in the figures represent extensions of nodes/infixes. In the DAWG in Figure 1, node `in` has two extensions with A and B, respectively leading to the left-closed infixes `op 1 in A` and `op 2 in B`. In the suffix tree, node `op` has two extensions with symbols 1 and 2, respectively leading to the right-closed infixes `op 1 in A` and `op 2 in B`. In the SCDAWG in Figure 1, black (blue) links represent right (left) extensions. Labels of left extensions are written in right-to-left order.

As we mentioned above, each index can be used to check if a string v represents an infix of the corpus in time linear in $|v|$. We sketch the procedure for the SCDAWG in Figure 1. Assume we want to check if `p 1` is an infix. Starting from the top node (empty string) we look for an extension to the right where the label starts with the letter `p`. We find an extension with label `p` leading to `op`. We continue to look for a right extension of the new node starting with `1`. We find the extension with label `1 in A` leading to `op 1 in A`. Hence `p 1` is an infix.

Advantages of the SCDAWG index

The strength of SCDAWGs compared to DAWGs and suffix trees relies on three features.

1. **Number of nodes.** In general, the number of nodes in the (S)CDAWG of a corpus is much smaller than the number of nodes in a suffix tree or DAWG. This is seen in Figure 1. A more informative picture is found in Table 1 where we compare the size of these index structures for a small collection of corpora of distinct sizes. For the sake of completeness, the table also includes data for suffix tries, a simplified version of suffix trees where each suffix represents a node.
2. **Naturalness of nodes.** Most of the infixes representing the nodes of a (S)CDAWG are very natural from an intuitive point of view[4]: imagine we index the German version of Goethe's Faust. Among the nodes of the suffix tree, we will find right-closed infixes such as `ephisto`, `phisto`, `retchen`, and `etchen`. In the DAWG we find left-closed infixes such as `Mephist`, `Mephis`, `Gretche` and `Gretch`. In the (S)CDAWG we only have the two-sided closures `Mephisto` and `Gretchen`. The example indicates that two-sided closure is a very natural property when looking for “interesting” and “meaningful” infixes of a corpus.
3. **Node Extension Property.** Among the infixes that represent the nodes of the index (suffix tree, DAWG, or (S)CDAWG) in general there are many inclusion relations where one infix v is a substring of another one w. In a suffix tree, DAWG, or CDAWG extension steps departing from node v in many cases do not lead to node w. For example, in the suffix tree in Figure 1 we cannot reach `op 1 in A` from `in` following the transitions. In contrast, the two-directional edge structure of SCDAWGs completely covers infix containment in the sense that *whenever node v is a substring of w there exists a chain of left and right extensions leading from v to w*. In other words, starting from a node v and traversing the index we find each context containing v. For the sake of reference, this property will be called the “Node Extension Property”. It can be used, for example, to find in a very elegant way all texts where an infix v occurs: starting from the node that represents (the two-sided closure of) v, compute all maximal chains of left extensions followed by all maximal chains of right extensions. The nodes reached at the end represent the texts in which v occurs. For example, in Figure 1 starting from `op` or `in` we reach the nodes representing the two input texts.

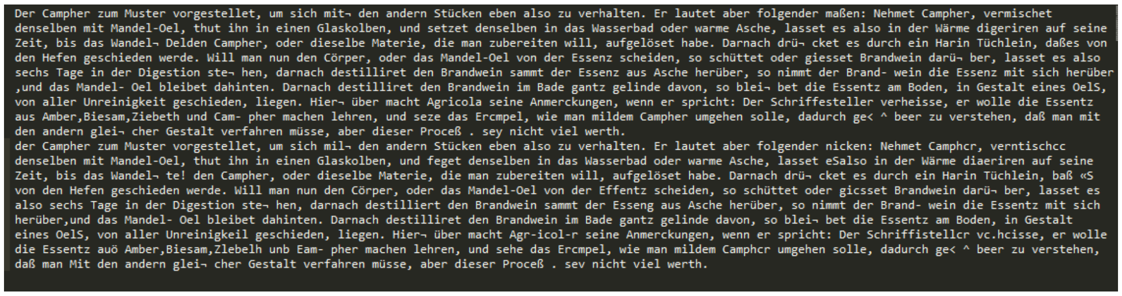
	Suffix Trie	**Suffix Tree**	**DAWG**	**CDAWG**	**SCDAWG**
	abcbc$abcab$: 12 Bytes; 12 symbols				
Nodes	66	17	16	6	6
Edges	65	16	23	13	21
	`OCR page`: 7 KB; 6.945 symbols				
Nodes	21.418.626	7.355	11.995	1.163	1.163
Edges	21.418.625	7.354	14.915	4.083	8.094
	`Excerpt EU-Corpus`: 106 KB; ca. 106.000 symbols				
Nodes	-	161.001	165.962	25.273	25.273
Edges	-	161.000	227.515	86.826	170.358
	`Small Corpus`: 1.2 MB; ca. 1.250.000 symbols				
Nodes	-	1.921.704	1.922.811	366.070	366.070
Edges	-	1.921.703	2.730.597	1.173.856	2.355.669

Table 1. Comparing the size of distinct index structures for four input texts.

Node-documents function. Using the Node Extension Property in one scan of the SCDAWG we may compute and store a function T that assigns to each node μ (represented as a number) the set of texts where μ occurs as an infix [Blumer et al. 1987]. T is called the *node-documents function*. Let us assume that each text comes with a number. We define a procedure FindTexts, the input is a node number μ, the output is a set of text numbers. The procedure is called in the form FindTexts(root). As a result, the function T is computed.

```
Procedure FindTexts($\mu$)
if $T(\mu)$ is already defined, then
    output $T(\mu)$;
else if $\mu$ is a leaf (text), then
    let $n$ be the number of the text;
    define $T(\mu) := \{n\}$;
    output $\{n\}$;
otherwise
    let $\nu_1,...,\nu_k$ be the (left or right) immediate successors of $\mu$.
    call FindTexts($\nu_1$), ...., FindTexts($\nu_k$);
    define \(T(\mu) :=\bigcup_{i=1,\ldots, k} T(\nu_i);\)
    output $T(\mu)$;
```

The procedure first visits all successor nodes of the input node μ and then assigns the union of the sets of text numbers of the direct successors to μ. It is simple to see that each link of the index is used only once. Hence, if the number of all distinct texts is treated as a constant, the procedure is linear in the size of the index.



```
Der Campher zum Muster vorgestellet, um sich mit¬ den andern Stücken eben also zu verhalten. Er lautet aber folgender maßen: Nehmet Campher, vermischet
denselben mit Mandel-Oel, thut ihn in einen Glaskolben, und setzet denselben in das Wasserbad oder warme Asche, lasset es also in der Wärme digeriren auf seine
Zeit, bis das Wandel¬ Delden Campher, oder dieselbe Materie, die man zubereiten will, aufgelöset habe. Darnach drü¬ cket es durch ein Harin Tüchlein, daßes von
den Hefen geschieden werde. Will man nun den Cörper, oder das Mandel-Oel von der Essenz scheiden, so schüttet oder giesset Brandwein darü¬ ber, lasset es also
sechs Tage in der Digestion ste¬ hen, darnach destilliret den Brandwein sammt der Essenz aus Asche herüber, so nimmt der Brand- wein die Essenz mit sich herüber
,und das Mandel- Oel bleibet dahinten. Darnach destilliret den Brandwein im Bade gantz gelinde davon, so blei¬ bet die Essentz am Boden, in Gestalt eines OelS,
von aller Unreinigkeit geschieden, liegen. Hier¬ über macht Agricola seine Anmerckungen, wenn er spricht: Der Schriffesteller verheisse, er wolle die Essentz
aus Amber,Biesam,Ziebeth und Cam- pher machen lehren, und seze das Ercmpel, wie man mildem Campher umgehen solle, dadurch ge< ^ beer zu verstehen, daß man mit
den andern glei¬ cher Gestalt verfahren müsse, aber dieser Proceß . sey nicht viel werth.
der Campher zum Muster vorgestellet, um sich mil¬ den andern Stücken eben also zu verhalten. Er lautet aber folgender nicken: Nehmet Camphcr, verntischcc
denselben mit Mandel-Oel, thut ihn in einen Glaskolben, und feget denselben in das Wasserbad oder warme Asche, lasset eSalso in der Wärme diaeriren auf seine
Zeit, bis das Wandel¬ te! den Campher, oder dieselbe Materie, die man zubereiten will, aufgelöset habe. Darnach drü¬ cket es durch ein Harin Tüchlein, baß «S
von den Hefen geschieden werde. Will man nun den Cörper, oder das Mandel-Oel von der Effentz scheiden, so schüttet oder gicsset Brandwein darü¬ ber, lasset es
also sechs Tage in der Digestion ste¬ hen, darnach destilliert den Brandwein sammt der Esseng aus Asche herüber, so nimmt der Brand- wein die Essentz mit sich
herüber,und das Mandel- Oel bleibet dahinten. Darnach destilliret den Brandwein im Bade gantz gelinde davon, so blei¬ bet die Essentz am Boden, in Gestalt
eines OelS, von aller Unreinigkeil geschieden, liegen. Hier¬ über macht Agr-icol-r seine Anmerckungen, wenn er spricht: Der Schriffistellcr vc.hcisse, er wolle
die Essentz auö Amber,Biesam,Zlebelh unb Eam- pher machen lehren, und sehe das Ercmpel, wie man mildem Camphcr umgehen solle, dadurch ge< ^ beer zu verstehen,
daß man Mit den andern glei¬ cher Gestalt verfahren müsse, aber dieser Proceß . sev nicht viel werth.
```

Figure 2. A text A leading to a value $r = 130,000$ for the LIS problem if compared with a similar parallel text B.

Corpora used for experiments

For the experiments we selected corpora of different periods, languages, and genres. Languages covered are Latin, English, German, French and Italian. Domains vary from political texts to poetry, lyrics and literature. Metadata include author, temporal data, and others. 9

Poems Corpus. The Poems Corpus consists of roughly 40k poems which had been compiled from TEI – XML annotated documents from the TextGrid Repository1. It contains poems written by German poets from about the time when printing was invented to the beginning of the 20th century. Amongst these authors are well known names like Johann Wolfgang v. Goethe, Friedrich Schiller, Joseph v. Eichendorff, or Georg Trakl. All poems are written in German language. In addition to the author each poem is furnished with the volume in which the poem was published as well as dates referring to the timespan in which the volume had been created. If this is not known the lifespan of the author is used for the dates. 10

Lyrics Corpus. The Lyrics Corpus contains about 15k of popular lyrics from songs, which had been present in the single charts of the German-speaking world, reaching from the mid nineteen fifties to nowadays. This corpus had been created by means of webcrawling. Aside from the interpreter it stores the year, in which the song had been in the charts, the chart position and whether the chart position had been a top 10 position, as well as the language, in which the song was performed, as its metadata. The language had been assigned automatically via n-gram based language classification. Since the manner of appearance of the considered single charts has changed over time, songs of the 80s and 90s are prevailing. 11

Parallel OCR Corpus. A smaller corpus of historic books, which also had been digitized by means of OCR. All documents in this collection had been created by two different OCR engines, leading to a parallel corpus of the same content. The corpus was only used in the alignments, and there no metadata had been considered. 12

Wittgenstein Corpus. The Wittgenstein Corpus consists of 3,871 original remarks of the German philosopher Ludwig Wittgenstein. Although the general language is German, the corpus also contains a few remarks in English. 13

Medline Corpus. The Medline Corpus is a collection of about 15 Million medical abstracts in English. Each abstract consists of about 10 sentences. We considered two subsets of this corpus. They were obtained by randomly selecting 0.5% and 10%, respectively, of the abstracts and gathering all the sentences in the resulting abstracts. The 0.5%-Medline Corpus consists of 800 K sentences, whereas the 10%-Medline Corpus consists of about 15 M sentences. 14

Using the index for alignment and text reuse

In this section we use the node-documents function defined in Section 2 to find large strings that co-occur in several texts of the collection. Such strings point to some form of text parallelism/reuse. Two applications are considered. In the first subsection we show how to efficiently align two or more parallel texts in a linear manner. In the second subsection we show how to explore large collections, looking at maximal strings that co-occur in several texts of the corpus 15

Index-based text alignment

In our context, to “align” two strings means to find a common linear representation where “common parts” of the two strings are shown, defining a linear sceleton, and where “deviations” or regions where the two strings do not agree are associated with the “holes” of the sceleton. The classical method for this task is the Needleman-Wunsch Algorithm [Needleman and Wunsch 1970]: The input are two texts A and B of length $m = |A|$, $n = |B|$. The algorithm finds an optimal alignment, given a scoring function. Using dynamic programming it fills the cells of a $|A| \times |B|$ matrix, cells come with “backward pointers”. The backward pointers are then used to find an optimal alignment. The complexity is $O(mn)$, which is problematic for large texts. E.g., comparing two texts with 10^6 symbols each, we need several terrabytes to store the full matrix. For a better solution we look at a strategy to translate the problem 16

Finding longest common subsequences and longest increasing subsequences. Such a translation can be found in the longest common subsequence or the longest increasing subsequence problem: Assume the scoring function for an alignment assigns value 1 to each pair of matching symbols and value 0 to each disagreement (here: insertion, deletion, or substitution). Then the matching symbols of the alignment are given by the longest common subsequence (LCS) of the two strings. A subsequence of a string $A = a_1 \ldots a_m$ is any string of the form $A = a_{i_1} \ldots a_{j_k}$ where $k \leq m$ and $i_1 < \ldots < i_k$. For example, “bdf” is a subsequence of “abcdef”. 17

This means that for a particular and natural scoring function, finding the LCS of two strings A and B helps to solve the global alignment problem for A and B. To avoid the above matrix computation with its large complexity we may use an alternative approach to solve the LCS, utilizing the longest increasing subsequence (LIS) problem. Given a sequence of natural numbers, we look for the longest increasing subsequence. For example, the LIS for $1-5-2-7-3-4$ is $1-2-3-4$. Given $A = a_1 \ldots a_m$ and $B = b_1 \ldots b_n$, the LCS problem for A and B can be reduced to the LIS problem [Gusfield 1997]. We ignore the details of the translation, but look at the complexity of the new procedure for solving the LCS. Let $r(i)$ denote the number of occurrences of the letter a_i in B, let $r = \sum_{i=1}^{m} r(i)$. Using translation into LIS, the LCS problem for A and B can be solved without dynamic programming in time $O(r\ log(n))$. 18

In many cases, using the LIS method helps to reduce the computation time for an alignment tasks. However, for large texts the value for the number r is typically very large and the computation time remains enormous. For example, the “innocent” text A shown in Figure 2, which is taken from a historical document, already leads to a value of $r \sim 130{,}000$ when compared with a similar text B. For more details on the LCS- or LIS- problem see [Gusfield 1997]. **Improved index-based alignment.** Consider the SCDAWG of two texts A and B. Applying the Node-documents function T it can be determined whether the infix represented by each node occurs in both texts. The main observation now is that using this information we immediately find large portions of the two texts that have a perfect alignment. We use the following definition. 19

Definition 4.1 An index node μ of the SCDAWG for two texts A and B is called *quasi maximal* if 20

1. the infix associated with μ occurs in both texts A and B,
2. μ does not have any transition that leads to another node μ’ with the Property 1.

As an illustration we refer to Figure 1. Our two example “texts” lead to two quasi maximal nodes `op` and `in` . Based on the *Node-documents function* there exists a simple procedure to find all quasi-maximal nodes: a queue is initialized with the root of the SCDAWG (empty string). We treat all entries of the queue in the order they are added to the queue. The treatment of a node μ involves two steps:

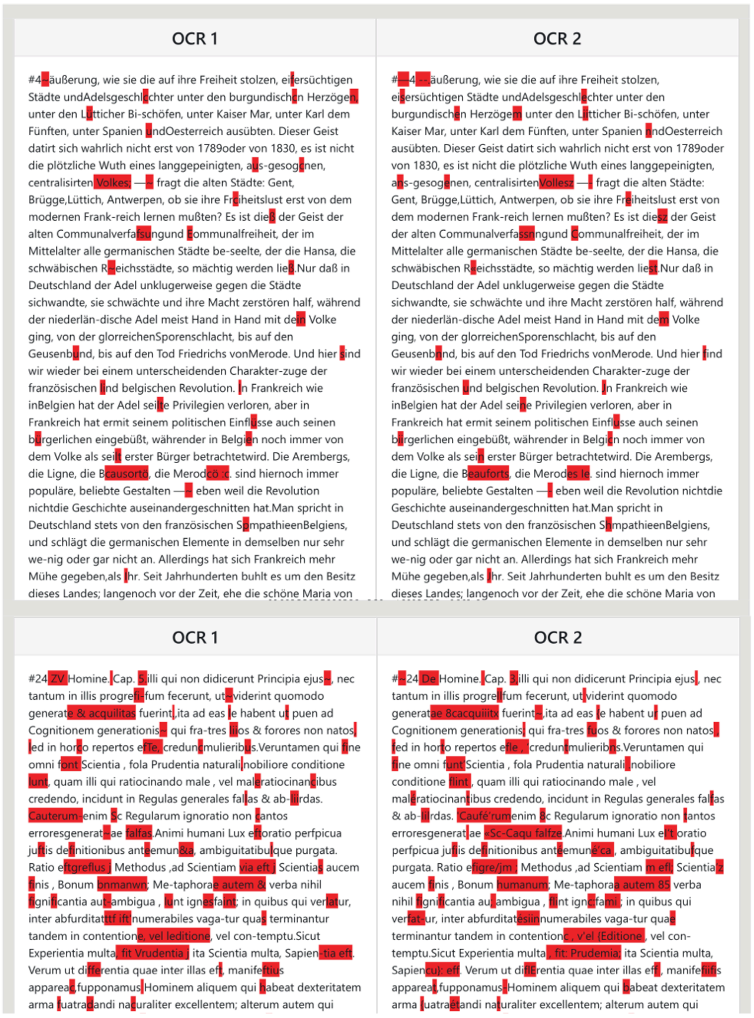
- We consider each node reached from μ with a single left or right transition in the SCDAWG. Each such node representing an infix occurring both in A and B is added to the end of the queue.
- If μ does not have any extension representing an infix occurring both in A and B, then μ is added to the list of quasi maximal nodes.

Note that each index node is added to the queue and treated at most once. At the end we obtain the full list of all quasi maximal nodes. Simple additional calculations provide the positions of the occurrences of each quasi maximal nodes.

Technical details are omitted.

The full alignment method proceeds as follows: We compute the SCDAWG index for the texts A and B (linear time). We traverse the index to find all quasi maximal nodes and the end positions of occurrences in A and B. We use end positions and the LIS method to define the linear sceleton of the alignment. When using the LIS method, the important new point to note is that the strings corresponding to the quasi maximal nodes are treated as single "symbols". In this way, the factor r mentioned in the above formula is drastically reduced. In the experiment from Figure 2, the usual procedure leads to $r \sim 130.000$, using the SCDAWG and quasi maximal nodes the value is $r = 29$. When aligning two OCR output texts of 5.500 symbols, the standard method yields $r \sim 2.300.000$, using the SCDAWG and quasi maximal nodes the value is only $r = 169$. 21

In a project related to improving OCR on historical documents we use this technique to align the outputs of distinct OCR engines for historical texts. "Holes" obtained from the alignment based on quasi maximal nodes sometimes cover portions of texts where a finer subanalysis helps to improve results. Figure 3 shows an alignment result for two OCR outputs and two texts, with distinct levels of disagreement. 22



OCR 1

#4~äußerung, wie sie die auf ihre Freiheit stolzen, eifersüchtigen
Städte undAdelsgeschlcchter unter den burgundischcn Herzögen,
unter den Lütticher Bi-schöfen, unter Kaiser Mar, unter Karl dem
Fünften, unter Spanien undOesterreich ausübten. Dieser Geist
datirt sich wahrlich nicht erst von 1789oder von 1830, es ist nicht
die plötzliche Wuth eines langgepeinigten, aus-gesogcnen,
centralisirten Volkes; —~ fragt die alten Städte: Gent,
Brügge,Lüttich, Antwerpen, ob sie ihre Frciheitslust erst von dem
modernen Frank-reich lernen mußten? Es ist dieß der Geist der
alten Communalverfafsungund Eommunalfreiheit, der im
Mittelalter alle germanischen Städte be-seelte, der die Hansa, die
schwäbischen R~eichsstädte, so mächtig werden ließ.Nur daß in
Deutschland der Adel unklugerweise gegen die Städte
sichwandte, sie schwächte und ihre Macht zerstören half, während
der niederlän-dische Adel meist Hand in Hand mit dein Volke
ging, von der glorreichenSporenschlacht, bis auf den
Geusenbund, bis auf den Tod Friedrichs vonMerode. Und hier sind
wir wieder bei einem unterscheidenden Charakter-zuge der
französischen lind belgischen Revolution. In Frankreich wie
inBelgien hat der Adel seilte Privilegien verloren, aber in
Frankreich hat ermit seinem politischen Einflüsse auch seinen
bürgerlichen eingebüßt, währender in Belgien noch immer von
dem Volke als seilt erster Bürger betrachtetwird. Die Arembergs,
die Ligne, die Bcausortö, die Merodcö :c. sind hiernoch immer
populäre, beliebte Gestalten —~ eben weil die Revolution
nichtdie Geschichte auseinandergeschnitten hat.Man spricht in
Deutschland stets von den französischen SpmpathieenBelgiens,
und schlägt die germanischen Elemente in demselben nur sehr
we-nig oder gar nicht an. Allerdings hat sich Frankreich mehr
Mühe gegeben,als Ihr. Seit Jahrhunderten buhlt es um den Besitz
dieses Landes; langenoch vor der Zeit, ehe die schöne Maria von

OCR 2

#—4 --.äußerung, wie sie die auf ihre Freiheit stolzen,
eisersüchtigen Städte undAdelsgeschlechter unter den
burgundischen Herzögem unter den Lütticher Bi-schöfen, unter
Kaiser Mar, unter Karl dem Fünften, unter Spanien nndOesterreich
ausübten. Dieser Geist datirt sich wahrlich nicht erst von 1789oder
von 1830, es ist nicht die plötzliche Wuth eines langgepeinigten,
ans-gesogenen, centralisirtenVollesz —- fragt die alten Städte:
Gent, Brügge,Lüttich, Antwerpen, ob sie ihre Freiheitslust erst von
dem modernen Frank-reich lernen mußten? Es ist diesz der Geist
der alten Communalverfassnngund Communalfreiheit, der im
Mittelalter alle germanischen Städte be-seelte, der die Hansa, die
schwäbischen R«eichsstädte, so mächtig werden liest.Nur daß in
Deutschland der Adel unklugerweise gegen die Städte
sichwandte, sie schwächte und ihre Macht zerstören half, während
der niederlän-dische Adel meist Hand in Hand mit dem Volke
ging, von der glorreichenSporenschlacht, bis auf den
Geusenbnnd, bis auf den Tod Friedrichs vonMerode. Und hier find
wir wieder bei einem unterscheidenden Charakter-zuge der
französischen und belgischen Revolution. Jn Frankreich wie
inBelgien hat der Adel seine Privilegien verloren, aber in
Frankreich hat ermit seinem politischen Einflusse auch seinen
biirgerlichen eingebüßt, währender in Belgicn noch immer von
dem Volke als sein erster Bürger betrachtetwird. Die Arembergs,
die Ligne, die Beauforts, die Merodes le. sind hiernoch immer
populäre, beliebte Gestalten —- eben weil die Revolution nichtdie
Geschichte auseinandergeschnitten hat.Man spricht in
Deutschland stets von den französischen ShmpathieenBelgiens,
und schlägt die germanischen Elemente in demselben nur sehr
we-nig oder gar nicht an. Allerdings hat sich Frankreich mehr
Mühe gegeben,als Jhr. Seit Jahrhunderten buhlt es um den Besitz
dieses Landes; langenoch vor der Zeit, ehe die schöne Maria von

OCR 1

#24 ZV Homine.|Cap. 5.illi qui non didicerunt Principia ejus~, nec
tantum in illis progrefi-fum fecerunt, ut~viderint quomodo
generate & acquilitas fuerint|,ita ad eas le habent ut puen ad
Cognitionem generationis~ qui fra-tres liios & forores non natos,
led in horco repertos efTe, creduncmulieribus.Veruntamen qui fine
omni font Scientia , fola Prudentia naturali|nobiliore conditione
lunt, quam illi qui ratiocinando male , vel maleratiocinancibus
credendo, incidunt in Regulas generales fallas & ab-liirdas.
Cauterum-enim Sc Regularum ignoratio non cantos
erroresgenerat~ae falfas.Animi humani Lux eftoratio perfpicua
juftis definitionibus anteemun&a, ambiguitatibulque purgata.
Ratio eftgreflus j Methodus ,ad Scientiam via eft j Scientias aucem
finis , Bonum bnmanwn; Me-taphorae autem & verba nihil
fignificantia aut-ambigua , lunt ignesfaint; in quibus qui verlatur,
inter abfurditatttf ift'numerabiles vaga-tur quas terminantur
tandem in contentione, vel leditione, vel con-temptu.Sicut
Experientia multa, fit Vrudentia j ita Scientia multa, Sapien-tia eft.
Verum ut differentia quae inter illas eft, manifeftius
appareac,fupponamus|Hominem aliquem qui habeat dexteritatem
arma fuatradandi nacuraliter excellentem; alterum autem qui

OCR 2

#~24 De Homine.|Cap. 3,illi qui non didicerunt Principia ejus|, nec
tantum in illis progrellfum fecerunt, ut|viderint quomodo
generatae 8cacquiiitx fuerint~,ita ad eas {e habent ur puen ad
Cognitionem generationis; qui fra-tres fuos & forores non natos ,
fed in horto repertos efle , 'creduntmulieribns.Veruntamen qui
fine omni funt'Scientia , fola Prudentia naturali .nobiliore
conditione flint , quam illi qui ratiocinando male , vel
malératiocinantibus credendo, incidunt in Regulas generales falfas
& ab-lilrdas. 'Caufé'rumenim 8c Regularum ignoratio non tantos
erroresgenerat|ae «Sc-Caqu falfze.Animi humani Lux el't oratio
perfpicua jufiis definitionibus antéemuné'ca , ambiguitatibufque
purgata. Ratio efigre/jm ; Methodus ,ad Scientiam m efl; Scientia'z
aucem finis , Bonum humanum; Me-taphoraa autem 85 verba
nihil fignificantia au; ambigua , flint ignc:fami ; in quibus qui
verfat-ur, inter abfurditatésiinnumerabiles vaga-tur quae
terminantur tandem in contentionc , v'el {Editione , vel con-
temptu.Sicut Experientia multa , fit: Prudemia; ita Scientia multa,
Sapiencu}: eff. Verum ut diflErentia quae inter illas eff , manifefiifis
appareat,fupponamus-Hominem aliquem qui babeat dexteritatem
arma {uatraétandi naturaliter excellentem; alterum autem qui

Figure 3. Two OCR outputs for two pages aligned using index technology. Black parts represent quasi maximal nodes and thus matches, red parts are disagreement regions.

Multiple string alignment. When dealing with $n > 2$ texts $A_1, \ldots, A_n$, the notion of a quasi maximal node is generalized. **Definition 4.2** An index node μ of the SCDAWG for $A_1, \ldots, A_n$ is called *quasi maximal* if 23

1. the infix associated with μ occurs in all texts $A_1, \ldots, A_n$,
2. μ does not have any transition that leads to another node μ' with Property 1.

For technical reasons we actually have the additional restriction that quasi maximal nodes used for alignment only occur once in each text. A generalization of the LIS algorithm can be used to find a maximal linear sequence of quasi maximal nodes of all texts, which gives the desired alignment sceleton. As an illustration, Figure 4 shows the simultaneous alignment of three texts. Two sequences represent parallel parts of the OCR-outputs of two OCR engines on a historical document, the third sequence shows the corresponding part of the ground truth file.

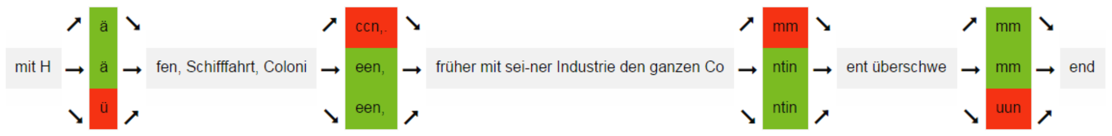
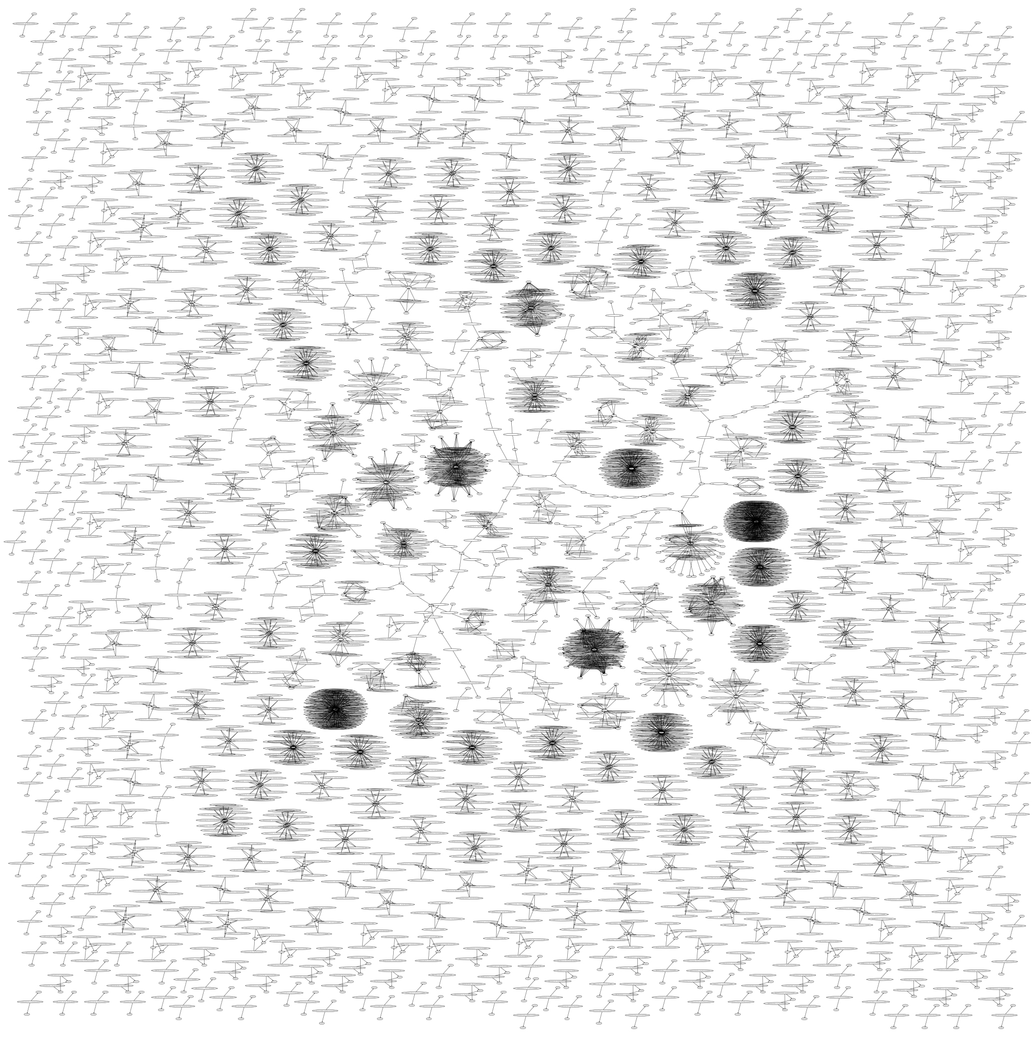


Figure 4. Subsegment of the multi-alignment of three texts (OCR outputs and ground truth) using quasi-maximal nodes. Grey parts represent quasi-maximal nodes, for the coloured regions at least two texts show a disagreement.

Detecting text reuse and similarities across collections}

24

When comparing large collections of texts $A_1, \dots A_n$, in most cases it is not natural to ask for substrings that co-occur in all texts. For the following experiment a node of the SCDAWG index for $A_1, \dots A_n$ is called quasi maximal if it is quasi maximal for two texts in sense of Definition 4.1. We have seen that we can extract the set of all quasi maximal nodes from the SCDAWG index in this sense, and for each such node μ with string v_μ we immediately get the information in which texts A_i the string v_μ occurs. In Figure 5 this information is used to compute “survey” graphs for text reuses in two complete collections (Poems Corpus and Lyrics Corpus). Each graph contains two types of nodes. Large and small ellipses respectively represent quasi maximal nodes and texts numbers of the collection. For each node μ a prefix and a suffix of the node text v_μ is shown. A link from a quasi maximal node μ to a text number k indicates that v_μ is a substring of A_k.



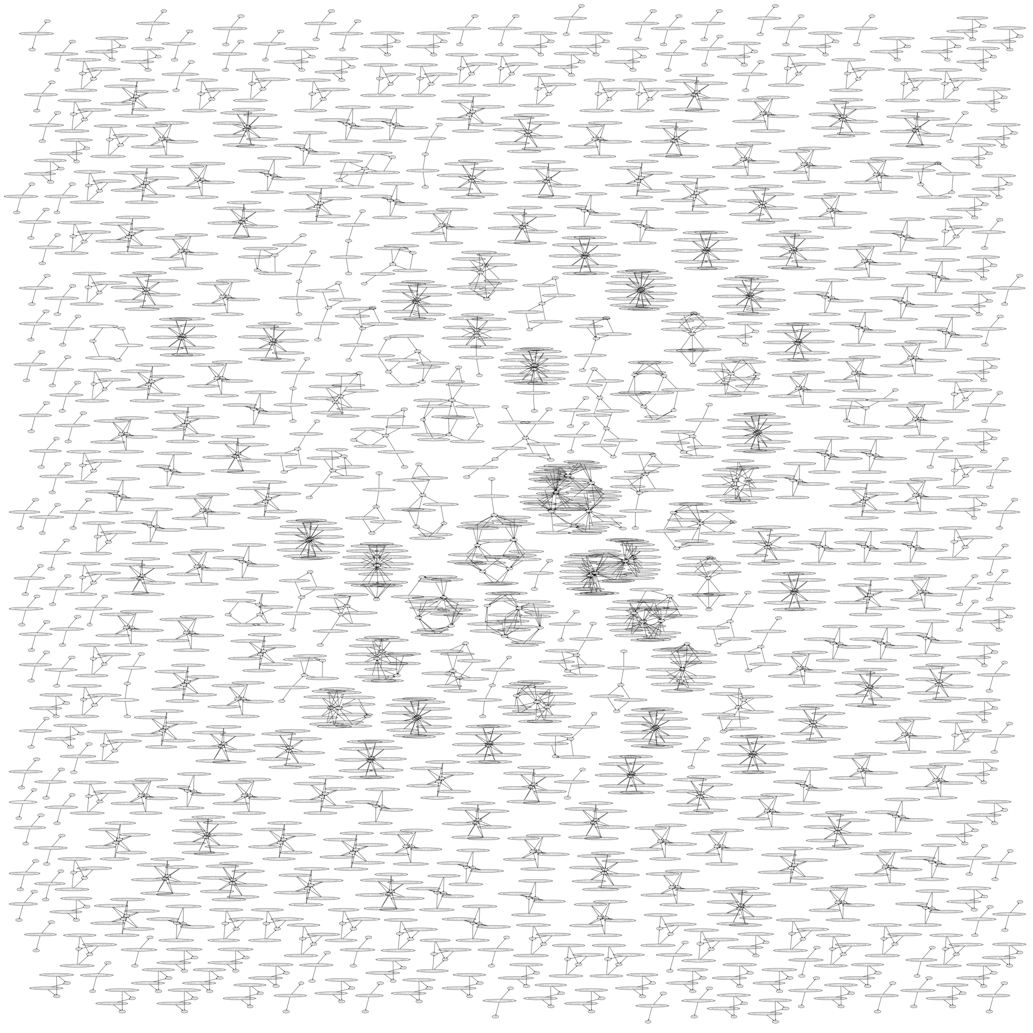


Figure 5. Text reuses in the Poems Corpus (top) and Lyrics Corpus (bottom) - quasi maximal nodes and pointers to poems/lyrics. This bird's eye perspective can be used to find interesting regions for closer inspection using zooming techniques. To view these images in more detail, download the high-resolution PDFs (5.1, 5.2).

Depending on the text reuses in the collection, the graph becomes very large. Still it is very useful to find interesting subregions with eye-catching multiple text reuses. In the lyrics corpus, many reuse effects can be traced back to covered songs. Since the song texts in the corpus were collected in a community based way, texts differ in details. In the survey graph eye-catching regions are mainly caused by songs that have been covered many times. It is then possible to zoom to these regions and to see reused text portions and the texts where these findings occur. Figures 6and 7 show such zoomed regions. In Figure 6 it can be seen that five poems in the collection have many text reuse connections. Some examples for reused text are (cf. Figure 6): 25

- `laut und leise, Unterric... träumt, Angenehme zu hören`
- `wälzend kam die Sturmesf ... der Chor vom ganzen Haine`

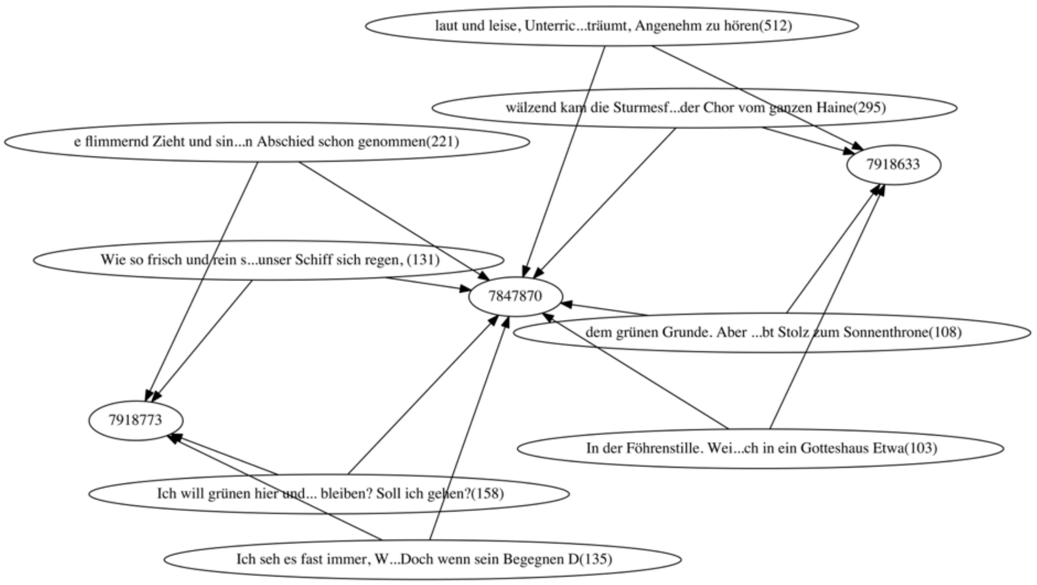

laut und leise, Unterric...träumt, Angenehm zu hören(512)
wälzend kam die Sturmesf...der Chor vom ganzen Haine(295)
e flimmernd Zieht und sin...n Abschied schon genommen(221)
7918633
Wie so frisch und rein s...unser Schiff sich regen, (131)
7847870
dem grünen Grunde. Aber ...bt Stolz zum Sonnenthrone(108)
7918773
In der Föhrenstille. Wei...ch in ein Gotteshaus Etwa(103)
Ich will grünen hier und... bleiben? Soll ich gehen?(158)
Ich seh es fast immer, W...Doch wenn sein Begegnen D(135)

Figure 6. Text reuses in the Poems Corpus - quasi maximal nodes and pointers to 5 poems, zoomed subregion of upper graph in Figure 5.

In Figure 7 many test reuses are centered around the word “Hallelujah”. Examples are

- `She tied you to a kitche .. Hallelujah, Hallelojah}`
- `s seen the light It's a -- ah Hallelujah, Hallelujah}`

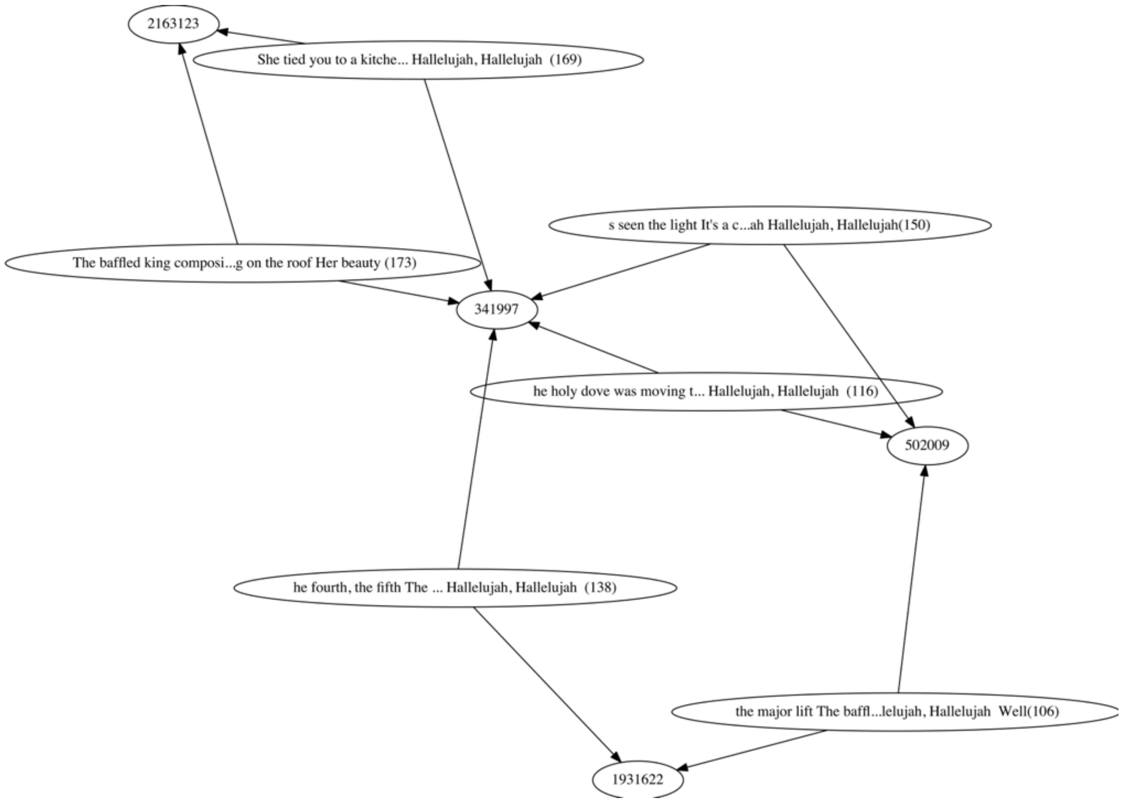

2163123
She tied you to a kitche... Hallelujah, Hallelujah (169)
s seen the light It's a c...ah Hallelujah, Hallelujah(150)
The baffled king composi...g on the roof Her beauty (173)
341997
he holy dove was moving t... Hallelujah, Hallelujah (116)
502009
he fourth, the fifth The ... Hallelujah, Hallelujah (138)
the major lift The baffl...lelujah, Hallelujah Well(106)
1931622

Figure 7. Text reuses in the Lyrics Corpus - quasi maximal nodes and pointers to lyrics, zoomed subregion of lower graph in Figure 5.

Refined “linguistic” view on a corpus

We have seen that the set of nodes of the (S)CDAWG for a corpus represents a relatively small set of infixes that often are “natural” portions of text. Experiments in Gerdjikov and Schulz (2016) show that many of the phrases corpus analyzers would look at (names, terminological expressions, stylistic phrases, ...) are typically left- and right-closed and appear among the nodes of the (S)CDAWG. This means that when just using the index nodes for corpus exploration we do not miss important portions of text. 26

From this point, the (S)CDAWG can be considered as a first step towards a new kind of “linguistic corpus index” where nodes - in the ideal case - *exactly* represent the linguistic units of interest (morphems, words, phrases, sentences, ...) of the corpus and their relationships. In a way, such an index would be the optimal basis for all corpus analysis tasks focussed on linguistic units [5]. However, for arbitrary texts there is not even a useful formal definition of a “phrase” or “linguistic unit”, let alone a procedure for finding this strings. 27

There are two possible main paths how to come closer to a “linguistic refinement” of the (S)CDAWG. On the one hand side, we can take any available NLP prodecure (parser, lexical analyser, phrase detector,..). Assuming that the units found represent nodes of the (S)CDAWG, we arrive at a (S)CDAWG substructure that directly points to units of interest and their occurrences in the corpus. On the other hand we may try to further analyze regularities found in the corpus to find a restricted set of nodes that comes closer to the idea of a linguistic phrase. In Gerdjikov and Schulz (2016), we followed the second path, which is completely language independent. Crucial assumptions are: 28

- phrases appear in distinct contexts,
- phrases are combined using function words as connectors, and
- sentences have natural decompositions into phrases, overlaps representing function words.

We then describe a bootstrapping method for finding function words, phrases and sentence decompositions into phrases.[6] Using the same kind of techniques, phrases can be further decomposed into *sub(sub)phrases*. As a matter of fact, for this form of decomposition also given lists of function words could be used. The main point is that the bootstrapping method in a completely unsupervised and language independent way leads to an interesting set of phrases and subphrases that helps to considerably reduce the set of nodes considered, thus coming closer to a “linguistic” index.

Finding important concepts of the corpus. When ignoring bordering function words, *atomic subphrases* obtained in this way typically represent content words or “concepts”. Analyzing the role of these concepts in phrase decomposition it is possible to compute a ranked list of characteristic concepts of the corpus. See Gerdjikov and Schulz (2016) for details. Below some examples for *most characteristic atomic phrases* in the example corpora are given. For example, in the Wittgenstein Corpus, central concepts found using the above methods are (cf. Figure 9) `Bedeutung` (meaning), `Farbe`(colour), `Sprache` (language), and `Erklärung` (explanation). 29

Exploring the use of concepts. After computing characteristic concepts, the aforementioned decomposition of phrases helps to find all larger phrases where a given concept occurs. In this way the use of the concept in the corpus can be studied. Figure 8 shows the hierarchical structure of phrases of the Wittgenstein Corpus extending `Bedeutung`. 30

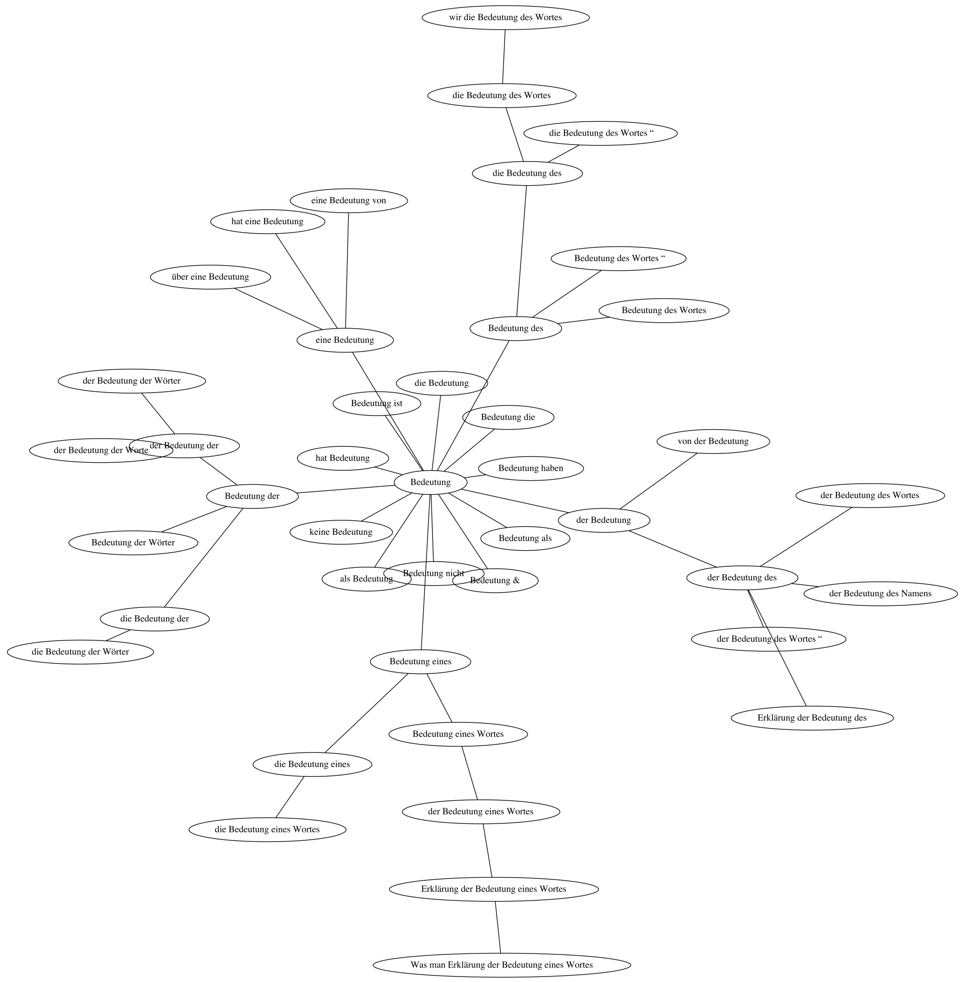


Figure 8. Exploring the use of concepts. The hierarchical structure of phrases extending the concept `Bedeutung` in the Wittgenstein Corpus detected by our approach.

Exploring the relationship of concepts in a visual way. A third goal is to explore interesting relationships between the concepts. Existing corpus exploration tools and automated approaches for exploring the paradigmatics of lexical units typically look at the co-occurrence of terms in documents, paragraphs, sentences or fixed size neighbourhoods [Schütze and Pedersen 1993],[Storjohann 2010]. Another, more “syntactic” view is obtained when looking at the co-occurrence of concepts in *phrases*. In Figures 9 and 10 we see the relationship of concepts (characteristic atomic kernels) in terms of co-occurrences in phrases. For example, in the Wittgenstein Corpus, `Bedeutung` (meaning) is strongly related to `Erklärung` (explanation), as can be seen from phrases like `Erklärung der Bedeutung eines Worts` (explanation of the meaning of a word). In the Medline Corpus (cf. Figure 10), concepts found to be related to `tumor` are, e.g., `risk` (witness phrase `risk for tumor`), `chemotherapy` (witness phrase `tumor cells to chemotherapy`), and `vaccines` (witness phrase `tumor vaccines`).

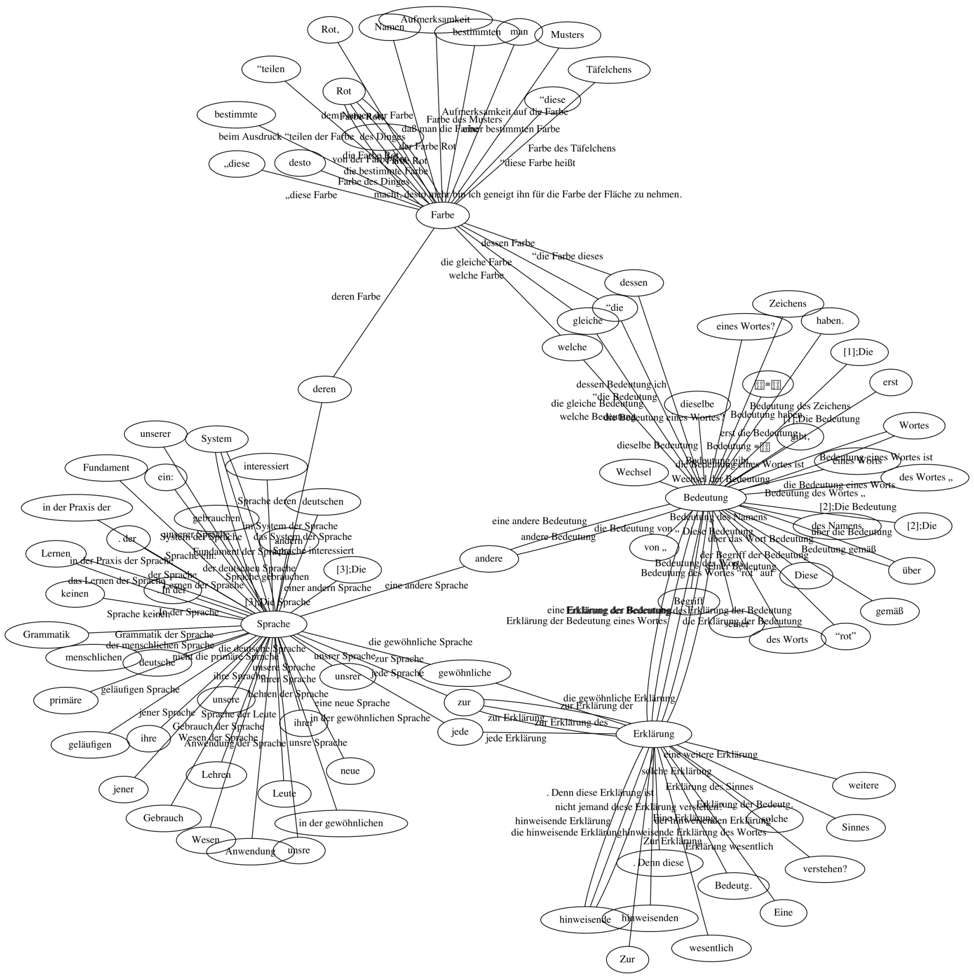


Figure 9. Exploring the relationship of concepts in a visual way. Network with "witness phrases" for the relationship between concepts derived from Wittgenstein Corpus.

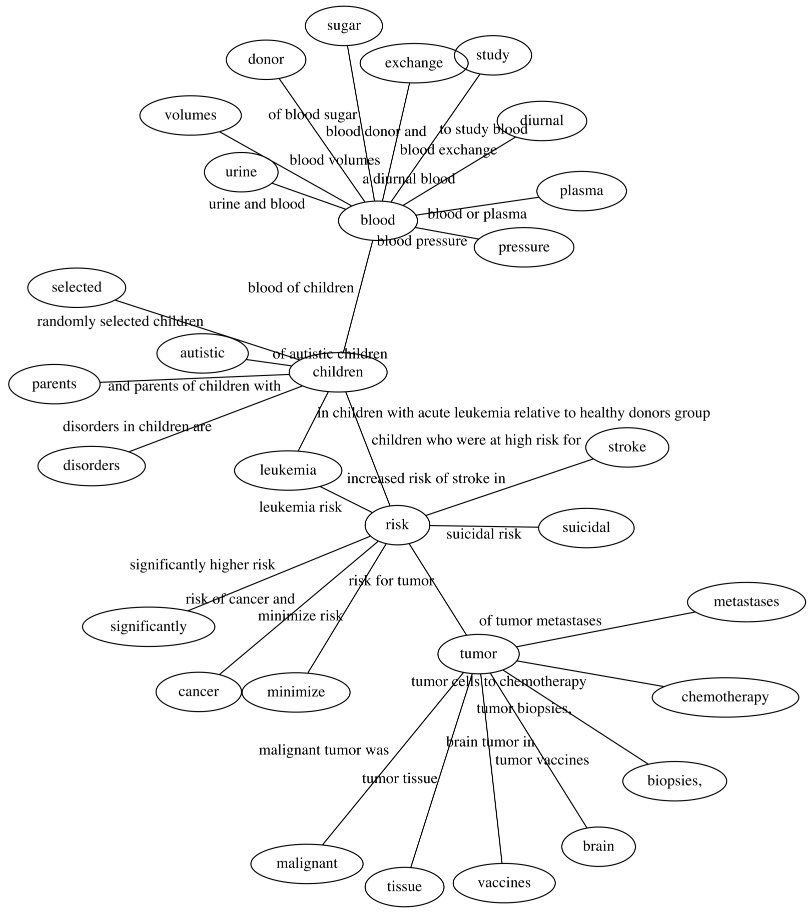


Figure 10. Exploring the relationship of concepts in a visual way. Network with "witness phrases" for the relationship between concepts derived from the Medline.

Corpus exploration using metadata

When metadata are available, many of the aforementioned techniques can be refined for revealing similarities in the space of metadata. Technically, we assign the metadata for the original documents to the nodes in the SCDAWG that represent those documents. For the rest of the nodes we can retrieve this information on demand by simple traversal of small parts of the SCDAWG. 32

Comparing authors. Our first illustration uses the Poems Corpus, where we have authorship and temporal metadata for texts. In Figure 11 we use the aforementioned refined "linguistic" view. After adding authorship information to the maximal nodes of the index we first adapt the notion of a quasi maximal node: a node μ with text v_μ is called *quasi maximal with respect to authors* if v_μ occurs in texts of two distinct authors, while any phrase that properly extends v_μ only occurs in the works of one author. 33

The nodes in Figure 11 represent important concepts in the corpus - some of the most characteristic atomic subphrases for the Poems Corpus in the sense considered above. Our language independent techniques revealed that `Welt` (world), `Sonne` (sun), `Himmel` (sky/heaven), `Gold` (gold), and `Schnee` (snow) are important concepts in the Poems Corpus. In Figure 11, each link between two concepts stands for a phrase that 34

- contains both concepts, and
- is quasi maximal with respect to authors.

The authors that have used the phrase are annotated with the link. For example it is seen that

- `Schnee zur Erde` (snow to earth) was both used by Nikolaus Lenau and Hermann Löns, and

- `der Himmel mit der Erde` (the heaven/sky with the earth) was both used by Betty Paoli and Friedrich Rückert.

This gives a basis for comparing two authors, now asking if both authors used similar (quasi-maximal) phrases with the central concepts of the corpus.

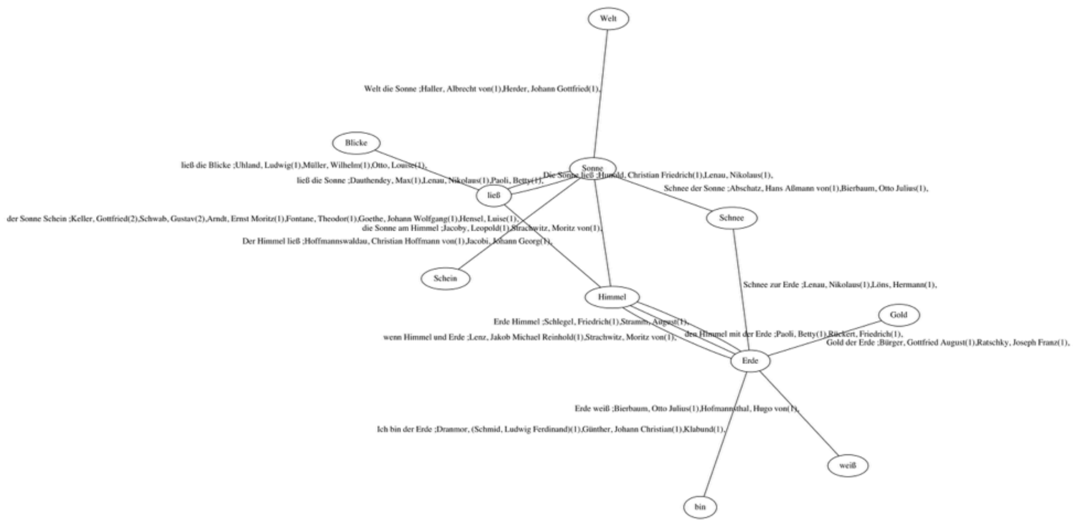

Welt
Welt die Sonne ;Haller, Albrecht von(1),Herder, Johann Gottfried(1),
Blicke
ließ die Blicke ;Uhland, Ludwig(1),Müller, Wilhelm(1),Otto, Louise(1),
Sonne
ließ die Sonne ;Dauthendey, Max(1),Lenau, Nikolaus(1),Paoli, Betty(1),
Die Sonne ließ ;Herold, Christian Friedrich(1),Lenau, Nikolaus(1),
Schnee der Sonne ;Abschatz, Hans Aßmann von(1),Bierbaum, Otto Julius(1),
ließ
der Sonne Schein ;Keller, Gottfried(2),Schwab, Gustav(2),Arndt, Ernst Moritz(1),Fontane, Theodor(1),Goethe, Johann Wolfgang(1),Hensel, Luise(1),
die Sonne am Himmel ;Jacoby, Leopold(1),Strachwitz, Moritz von(1),
Der Himmel ließ ;Hoffmannswaldau, Christian Hoffmann von(1),Jacobi, Johann Georg(1),
Schnee
Schein
Schnee zur Erde ;Lenau, Nikolaus(1),Löns, Hermann(1),
Himmel
Gold
Erde Himmel ;Schlegel, Friedrich(1),Stramm, August(1),
den Himmel mit der Erde ;Paoli, Betty(1),Rückert, Friedrich(1),
wenn Himmel und Erde ;Lenz, Jakob Michael Reinhold(1),Strachwitz, Moritz von(1),
Gold der Erde ;Bürger, Gottfried August(1),Ratschky, Joseph Franz(1),
Erde
Erde weiß ;Bierbaum, Otto Julius(1),Hofmannsthal, Hugo von(1),
Ich bin der Erde ;Dranmor, (Schmid, Ludwig Ferdinand)(1),Günther, Johann Christian(1),Klabund(1),
weiß
bin

Figure 11. Use of Metadata. Connecting concepts (nodes) by means of phrases that are maximal with respect to the property of being used by two poets. Edges represent phrases plus the two poets that used the phrase.

Temporal development of similarities between authors. In Figure 12 we see a dual variant of the graph where the role of nodes and edges is changed. Nodes now represent authors, authors are linked if they have used identical quasi maximal (w.r.t authors) phrases. Using colouring of nodes, also temporal information about authors is added - we simply used the year defining the middle point in the life of an author. The spectral ordering of colours red-green-blue-purple represents the temporal timeline from earlier periods to later periods. Links always point from earlier authors to later authors. 35

In the lower part of the graph we find a path from Anna Louisa Karsch (1756) Figure 12, circle marker (1)) $\longrightarrow$ Friedrich Gottlieb Klopstock (1763) $\longrightarrow$ Ernst Schulze (1803) Figure 12, circle marker (2)). The latter is also reached with a link from Johann Wolfgang von Goethe (1794) Figure 12, circle marker (3)). Goethe has several successors. In this graph, an even more central person is Ludwig Achim von Arnim (1806) Figure 12, circle marker (4)), who builds the center of the large cluster in the middle of the figure. The temporal spectrum of this cluster corresponds to the first half of the 19th century. Authors of the 17th and 18th century are found in the second cluster on the top right part. An early “influential” author of this period is Simon Dach (1632) Figure 12, circle marker (5)). 36

The two clusters indicate a change of literary concepts covered between the Baroque/Pre-Romantic period and the Romantic period. The three authors (John Brinckman (1842), Fritz Reuter (1842), and Klaus Groth (1859) Figure 12, circle marker (6)), who form the cluster right to Romantic period cluster however are not connected to the cluster consisting of other authors of their period due to the fact that all three were writing poems in the german dialect of niederdeutsch. 37

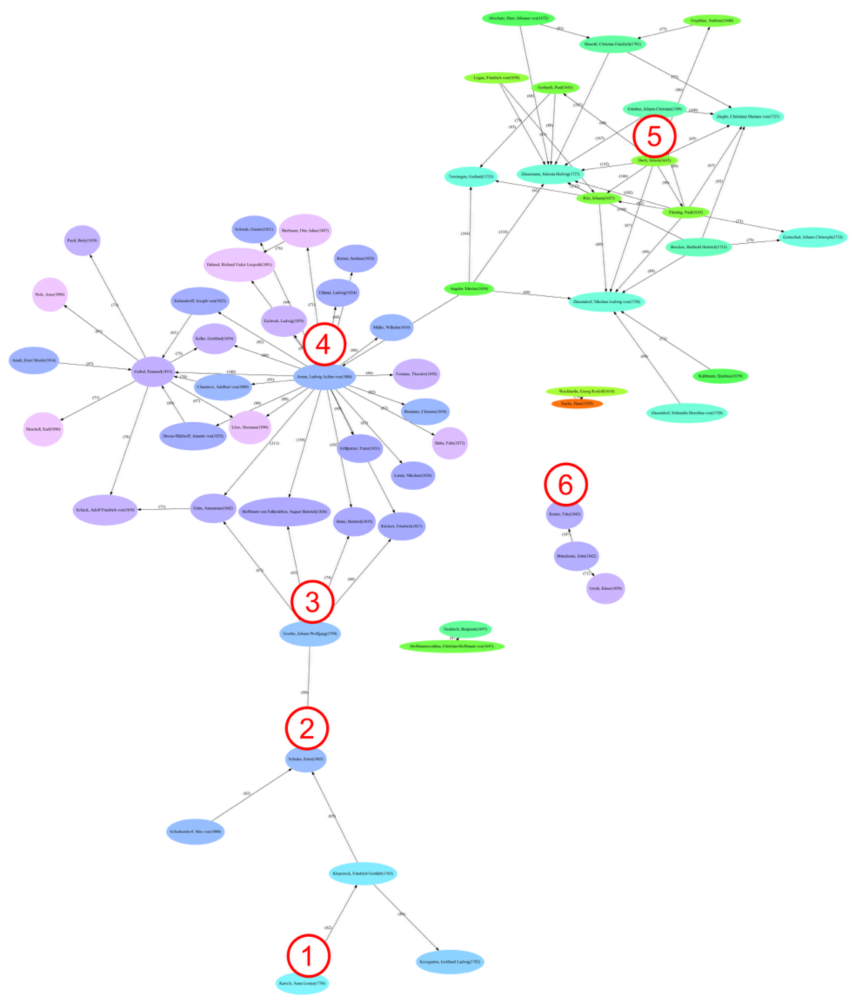


Figure 12. Use of Metadata in the Poems Corpus. The graph is obtained as the dual variant of the graph in Figure 11. Links between two poets mean that both have used similar maximal phrases around the concepts shown in Figure11. Colours correspond to temporal information for authors. Links point from earlier authors to later authors.

When constructing this kind of graph for the Lyrics Corpus similar observations can be made. The nodes of Figure 13 again are connected by co-occurring quasi maximal phrases. Instead of the author now band names are used together with the mean of the release years of their corresponding songs. The colour coding is the same as used in the previous experiment. One of the first things that can be observed is a cluster of Italian singers centered around Eros Ramazotti (1998) (See Figure 13, circle marker (1)), located at the right to the center of the graph. Also lots of subgraphs, which only contain of two or three interpreters are seen. They often vizualize relations between cover songs, which are contained in the corpus. Therefore for instance Right Said Fred (1998) and Peter Sarstedt (1969) (See Figure 13, circle marker (2)) can be found at the bottom left. Another example of this is a three node cluster at the top right with edges leading from Julie Covington (1977) and Don McLean (1972) to Madonna (1996) (See Figure 13, circle marker (3)), who famously released covers of the songs “Don't Cry For Me Argentina” and “American Pie”. Other subgraphs connect interpreters who obviously belong to the same genre, e.g. at little to the right of the center a path can be found leading from 2Pac (2000) $\longrightarrow$ Busta Rhymes (2001) $\longrightarrow$ Kanye West (2007) (See Figure 13, circle marker (4)), with the interpreters representing famous hip-hop artists.

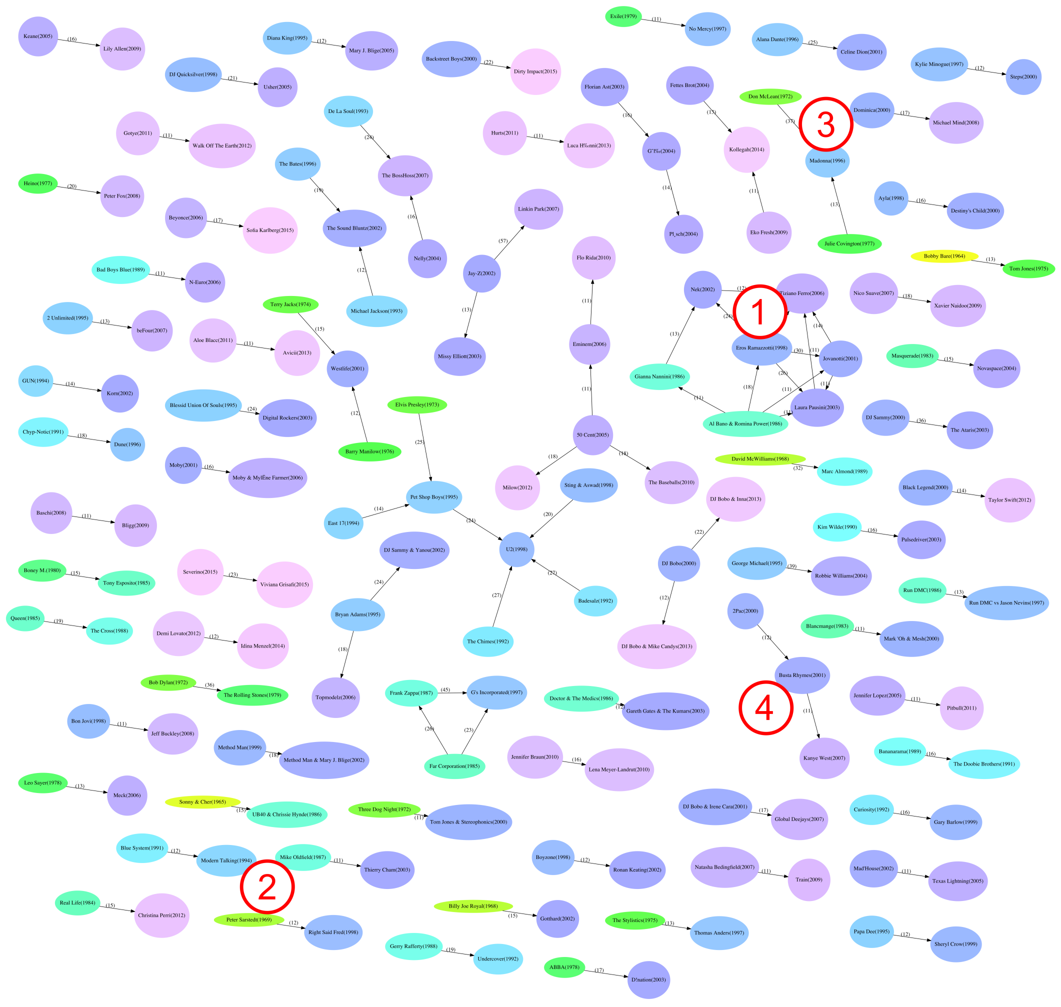


Figure 13. Use of Metadata in the Lyrics Corpus. The graph is obtained by connecting interpreters by their commonly used phrases. Node colours again encode temporal metadata. Links point from older interpreters to younger ones.

Loose ends - diachronic language variation and classification tasks

Even though the focus of the current paper are the problems of text-reuse and mining of relationships based on it, we suggest that our approach can be extended to other related problems, e.g. diachronic language variation and text classification. 39

In Gerdjikov et al. (2013) and Sariev et al. (2014), the authors develop a historical text normalization system. Based on a modern dictionary and several thousands of pairs, historical word and its modern variant, they show that the DAWG structure can be used to automatically learn and extract spelling variations. The main benefit of the infix structure is that: (i) it does not restrict the spelling variation to any particular predefined patterns and (ii) it reflects the entire structure of the language (on word level). In Sariev et al. (2014) the combination of this technique with a language model for the modern language yields a complete historical text normalization system. 40

Of course, the requirement for training data sets practical limitations -- training data is often unavailable and its production is expensive and time-consuming. In Mitankin et al. (2014) the authors try to address this problem and suggest a completely automatic historical text normalization approach. The main idea is to automatically generate pairs of historical word and its modern variant. Mitankin et al. (2014) proposes to use Levenshtein edit distance and approximate search in the modern dictionary in order to generate for each historical word its most relevant modern variant(s). Afterwards we are in the situation to run the historical text normalization system from Mitankin et al. (2014) 41 and Sariev et al. (2014). Unfortunately, the Levenshtein edit distance, as a generator of pairs, turns out to introduce a lot of noise in the system. In particular, the accuracy of the normalization system drops from 94% to 82%.

42 The results from the current paper suggest two straightforward approaches in order to reduce the noise introduced by the unsupervised approximate search. The first approach is the following. Instead of searching for *historical words* and their *modern word variants*, to search for *historical phrases* and their *modern phrase variants*, automatically extracted from the SCDAWG structure for the historical texts and modern texts, respectively. In this way we plug in additional language context in the query, which in general will exclude orthographically similar but semantically irrelevant candidates. On the other hand, if the modern text corpus covers the domain of the historical texts, it is possible that the basic phrases in the historic language have indeed been preserved in the modern language and do have their spelling variants in the modern language. Thus, without seriously reducing the recall, we could increase the quality of the automatically generated pairs *historical phrases* and their *modern phrase var*. The production of pairs of words is then straightforward[7].

43 The second approach is the following. Instead of modifying the searching space, we can modify the edit-distance and use more relevant edit-distance mechanism. As described in Section 4, we can use the SCDAWG structure in order to align different editions of the same source. In particular, if the editions belong to close time periods, the most disagreements in the alignment will be due to uncertainties in the spelling that have been typical in this time period. This phenomenon can be used in order to constrain the elementary edit-operations and the contexts in which they are applicable. As a result we can expect that the generator of pairs, historic word and its modern variant, will be more relevant with respect to the structure of the historical language.

44 We should stress that the arguments raised in the previous two paragraphs require further research. It is also a challenging open problem to directly map the SCDAWG structure of a historical corpus to a substructure of the SCDAWG structure for a large modern language. The latter, of course, would resolve the normalization problem.

45 The substrings of a SCDAWG index could also be used to train models for text classification tasks. A similiar approach, which uses maximal substrings of a suffix tree and led to promising results, had been persued in Okanohara and Tsujii (2009). As stated in Section 2 the two-sided closures, which form the nodes of the SCDAWG, represent more natural infixes than those of a suffix tree or DAWG. Therefore they may render descriptive features which still are not restricted to fixed word boundaries. To find optimal features in the substrings of a SCDAWG of a corpus again metadata can be used to find longest or shortest substrings which occur only in a certain instance of a metadata attribute. For example the longest/shortest substrings which belong to the author “Goethe” can be used as features to train a text classification model with regard of this metadata attribute.

Conclusion

46 In this paper we have shown how symmetric compact acyclic word graphs (SCDAWGs) can help to efficiently mine interesting portions of texts in corpora, which serve as a basis for many ways of comparing texts like alignment and the detection of text reuse (Section 4). These findings can be refined by adding a form of “linguistic analysis” where “phrases”, “subphrases”, “function words”, and “important content words” (characteristic atomic kernels) are computed in an unsupervised way without using prior linguistic knowledge (Section 5). If metadata for the texts in the collection are available, additional information stored in the index and the above techniques give a basis for finding similarities in the space of metadata (Section 6).

47 Of course many of the here mentioned techniques would require more detailed analysis and comparsion with other existing methods. However this was not the purpose of this paper, since its focus didn't lie on the development and description of a certain method but rather on showing how manifold and versatile the application of such an index can be to the task of large scale corpus exploration. In many cases, various variants exist for the concrete methods used in our examples. When moving to real applications, experts from Digital Humanities are needed to find those variants that are most adequate and lead to real insights.

Acknowledgements

We express our gratitude to: Uwe Springmann for his help with OCR recognition of historic documents and the corpus derived from them. We also thank Stefanie Schneider for the provision of the lyrics corpus. Part of the results described in this paper were obtained during the stay of the third author at LMU for which he has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007--2013) under REA grant agreement 625160.

Works Cited

Blumer et al. 1987 Blumer, A., Blumer, J., Haussler, D., McConnell, R., and Ehrenfeucht, A.“Complete inverted files for efficient text retrieval and analysis”. *Journal of the ACM (JACM)* 34 (1987): 578-595.

Büchler et al. 2012 Büchler, M., Crane, G., Moritz, M., and Babeu, A. “ Increasing recall for text re-use in historical documents to support research in the humanities ”. In *Proceedings of 16th International Conference on Theory and Practice of Digital Libraries, (tpdl 2012):* pp. 95–100 *Springer Berlin Heidelberg*.

Gerdjikov and Schulz 2016 Gerdjikov, S., and Schulz, K. U. “Corpus analysis without prior linguistic knowledge-unsupervised mining of phrases and subphrase structure”. *ArXiv e-prints* (2016): 1602.05772.

Gerdjikov et al. 2013 Gerdjikov, S. and Mihov, S. and Nenchev, V. “Extraction of spelling variations from language structure for noisy text correction”. In *Proc. Int. Conf. for Document Analysis and Recognition* (2013): 324-328.

Gusfield 1997 Gusfield, D. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press, Cambridge, 1997.

Inenaga et al. 2005 Inenaga, S., Hoshino, H., Shinohara, A., Takeda, M., Arikawa, S., Mauri, G., and Pavesi, G.“On-line construction of compact directed acyclic word graphs”. *Discrete Applied Mathematics* 146 (2005): 156-179.

McCreight 1976 McCreight, Edward M.“A space-economical suffix tree construction algorithm”. *Journal of the ACM (JACM)* 23 (1976): 262-272.

Mitankin et al. 2014 Mitankin, P. and Gerdjikov, S. and Mihov, S. “ An approach to unsupervised historical text normalization ”. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage:* (2014) 29-34.

Needleman and Wunsch 1970 Needleman, S. B., and Wunsch, C. D.“A general method applicable to the search for similarities in the amino acid sequence of two proteins.”. *Journal of molecular biology* 48 (1970): 443-453.

Okanohara and Tsujii 2009 Okanohara, Daisuke and Tsujii, Jun'ichi“Text categorization with all substring features”. In *Proceedings of the 2009 SIAM International Conference on Data Mining:* (2009) 839-846.

Sariev et al. 2014 Sariev, Andrei and Nenchev, Vladislav and Gerdjikov, Stefan and Mitankin, Petar and Ganchev, Hristo and Mihov, Stoyan and Tinchev, Tinko“Flexible noisy text correction”. In *Proceedings of Document Analysis Systems:* (2014)

Schütze and Pedersen 1993 Schütze, H., and Pedersen, J.“ A vector model for syntagmatic and paradigmatic relatedness ”. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research, (oed 1993):* pp. 104–113.

Storjohann 2010 Storjohann, P. (Ed.) *Lexical-semantic relations: theoretical and practical perspectives (Vol. 28)*. John Benjamins Publishing, Cambridge, 2010.

Ukkonen 1995 Ukkonen, Esko. “On-line construction of suffix trees”. *Algorithmica*14 (1995): 249-260.

Weiner 1973 Weiner, P. “Linear pattern matching algorithms”. In *Proceedings of 14th Annual Symposium on Switching and Automata Theory, (swat 1973):* pp. 1–11 *IEEE*.