Scaling Historical Text Re-use

Marco Büchler
Göttingen Centre for Digital Humanities
Georg-August-University Göttingen
Göttingen, Germany
Email: mbuechler@gcdh.de

Greta Franzini, Emily Franzini, Maria Moritz
Computer Science Department
University of Leipzig
Leipzig, Germany
Email: [franzini|efranzini|moritz]@informatik.uni-leipzig.de

***Abstract*—*Text re-use* describes the spoken and written repetition of information. *Historical text re-use*, with its longer time span, embraces a larger set of morphological, linguistic, syntactic, semantic and copying variations, thus adding complication to *text-reuse* detection. Furthermore, it increases the chances of redundancy in a digital library. In *Natural Language Processing* it is crucial to remove these redundancies before we can apply any kind of machine learning techniques to the text. In Humanities, these redundancies foreground textual criticism and allow scholars to identify lines of transmission. Identification can be accomplished by way of automatic or semi-automatic methods. Text re-use algorithms, however, are of squared complexity and call for higher computational power. The present paper addresses this issue of complexity, with a particular focus on its algorithmic implications and solutions.**

***Keywords*-text re-use, performance, scalability**

I. Introduction

Recent research in Computer Science has witnessed an increased interest in Big Data. Ulrike Rieß, editor of *Speicherguide*, defined *Big Data* as: i) large volumes of data that cannot be processed manually, ii) less structured data when compared to that of warehouse systems, and iii) linked data between heterogeneous and distributed sets [1].

There are numerous definitions and interpretations of Big Data. Most of them, however, do not necessarily focus on digital data but, rather, describe, in the words of Rieß, man-made data and its properties.

In order to better understand the meaning for scholarship, we first need to look at the emergence of Big Data in Humanities. Libraries[1], archaeological finds[2], card catalogues[3], manuscript editions (*Codex Sinaiticus*[4], *Codex Suprasliensis*[5], the *Newton Project*[6], the *Darwin Correspondence* project[7] and the *Aurelii Augustini De Civitate Dei* project[8]) and numismatics[9] champion years', even centuries', are worth of manual Big Data.

By gradually providing and presenting scholarly Big Data digitally, we are now able to investigate historical texts in broader and more comprehensive ways than ever before. At the same time, the bigger the data, the more difficult it becomes to search and browse large collections. The Digital Humanities contribution to this growth is its support towards the creation of those tools, visualisations and user interfaces, which have now become instrumental in the exploration of mass data, as well as an integral part of digital ecosystems.

In recent years research methodologies in the Humanities have been gradually changing. As recently as thirty years ago, access restrictions to libraries posed numerous challenges to humanists working with printed books and manuscripts. Today, the efforts of mass digitisation provide broader access to these items in digital form. The increasing availability of digitally encoded texts expedites and facilitates the exploration of text patterns. Google's mass digitisation effort, for example, is driving the improvement of close reading methods. The question "*What do you do with a million books?*" [2] did not only kickstart the *Digging into Data*[10] programme[11], but also addressed the potential use of distant reading methods in virtually any form of text mining.

Methodologically, this growing electronic environment catapults Digital Humanities research into a seemingly paradoxical situation of concurrent *Information Overload* and *Information Poverty* [3]. *Information Overload* may be understood as multiple data waves that simultaneously increase entropy and slow down academic decision processes. *Information Poverty*, on the other hand, describes the wealth of material which has not survived historical, natural and man-made disasters, such as the World War II bombings, the destruction of the library of Alexandria and the eruption of Mount Vesuvius in 79 AD. It follows that despite the *Information Overload* at our disposal, we in the 21st century look back at antiquity and at the Middle Ages with a very fragmentary and "information-poor" view of our textual heritage.

[1] http://dp.la/

[2] http://arachne.uni-koeln.de/drupal/

[3] http://catalog.loc.gov/

[4] http://www.codexsinaiticus.org/en/

[5] http://suprasliensis.obdurodon.org/

[6] http://www.newtonproject.sussex.ac.uk/prism.php?id=1

[7] http://www.darwinproject.ac.uk/

[8] https://sites.google.com/site/gretafranzini/

[9] http://vcrc.austincollege.edu/

[10] http://www.diggingintodata.org/

[11] supported by the National Endowment for the Humanities (NEH)http://ww.neh.gov/

This scattered legacy underpins the field of *textual criticism*, whereby scholars compare manuscripts in an attempt to, amongst other things, trace the origin of a text and reconstruct lost works. Quotations can be investigated in order to identify editorial contamination or fragments of lost authors who survive through other texts [4], [5].

Today, this long tradition of scholarly activity can be supported in a digital ecosystem by way of text re-use techniques that automatically find data parallels in bigger collections [6], [7]. One scholarly criterion that is often used is the property of *completeness*, which seeks to find all occurrences of a re-used instance of a text chunk.

The *completeness* of text re-use leads us to assume that Digital Humanities should focus on, and weigh, *recall* over *precision* [8]. Empirical research in the last six years, however, has shown that *completeness* challenges algorithms with different *re-use types*, such as *Idiom*, *Definition*, *Wisdom*, *Battle Cry*, and *Summary*, as well as different *re-use styles* like *literal quotation*, *paraphrase* and *allusion*. *Re-use styles* differ in the length of the re-use chunk, in the domain in which it has been re-used and in the intent of the re-use - a *Definition* being a deliberate re-use and an *Idiom* or *Winged Word* arbitrary re-uses. This is also the reason why we employ the term *text re-use* instead of *quotation detection*, in that we would otherwise need to deal with plagiarism as well. The *re-use diversity* that emerges from unknown distributions of *re-use types* and *re-use styles* has been extensively investigated in [9] and will not, therefore, be further discussed here.

Our research addresses the following question: If Big Data provides us with additional parallel texts –what every textual scholar dreams of– will we still be able to deal with the complexity and the computational needs of those big scale text re-use analyses?

II. Complexity of Text Re-use Detection

Text re-use implies the pre-computation of all possible links between two aligned text chunks. In Big Data, pre-computation is key to a smooth user-interface experience insomuch as it minimises the computational burden produced by hundreds of real-time requests.

Formally, the process of text re-use detection can be understood as the creation of a hyperlink structure based on a minimum similarity of two text chunks. Therefore, we can define the re-use graph $G = (V, E)$ with V as the number of vertices, i.e. lines, sentences, paragraphs or pages. The choice of this window size strongly depends on the size of the aforementioned *re-use chunks*. $(v_i, v_j) \in E$ represents the set of links between the two text chunks v_i and v_j with $v_i \in V$ and $v_j \in V$. When comparing two digital libraries with the vertices V_1 and V_2 the text re-use graph G is represented by $G = (V_1, V_2, E)$, whereas $(v_i^1, v_j^2) \in E$ describes the number of edges in this bipartite graph.

The naive *Brute Force* method used to compute text re-use checks every *re-use unit* v_i in a digital library with every other *re-use unit* v_j for similarities. If a pair of *re-use units* passes the similarity threshold, then the units are considered *re-use candidates*. This *Brute Force Linking (BFL)* can be computationally expressed as equation 1.

$$BFL = \big|V\big| \cdot \big(\big|V\big| - 1\big) \tag{1}$$

BFL links the naive *Brute Force* to a problem of squared computational complexity $O(n^2)$ with a doubling of data size, thus requiring four times as much computational power.

An experiment may be of use in clarifying the point. Let us assume that we are interested in a digital library of one million books. Each book has 50 pages. Every page contains 20 sentences, each representing a *re-use unit* v_i. In total, $|V|$ corresponds to one billion *re-use units* v_i. Comparing each and every *re-use unit* v_i with each and every *re-use unit* v_j produces $(10^9)^2 = 10^{18}$ different correlations. If we were to compare 1000 pairs of re-use units every second to form $(v_i, v_j) \in E$, we would need 10^{15} seconds or roughly 31.7 million years of single-threaded processing alone.

In Computer Science, one rectifies these complex problems by means of the *Divide & Conquer Strategy*, such as with *Merge Sort* [10]. In other words, problems are firstly split into subproblems, which are then, in turn, either recursively broken into further subproblems or solved using a smaller number of items (*Conquer*). The solved subproblems are finally merged into one working whole.

One *Divide & Conquer Strategy* for text re-use detection is parallelisation in an Apache Hadoop[12] cluster. To bring our experiment to an end, let us assume that we have a Hadoop cluster of 31.7 million processors and that no time is spent on scheduling and merging the data: the detection would still require one year of computation with massive parallelisation. As of August 2014, Big Data content providers such as the *Internet Archive*[13] or *Google Books*[14] contain about 6 million and 15 million books respectively, thus increasing the aforementioned computational power by 36 and 225 times.

Even if this experiment is imbued with optimistic assumptions, it is nevertheless clear that a *Brute Force* method, often used on small amounts of data, cannot find *all* parallel texts in Big Data collections. The two questions we ask ourselves at this point are: is text re-use computable on Big Data? If so, how do we compute it?

III. State of the Art

As per the previous section, *Brute Force* techniques only work on small data. For this reason, recent approaches adopt *Feature-Based Linking (FBL)* strategies [11]–[13]. While the

[12] http://hadoop.apache.org/

[13] https://archive.org/details/texts

[14] http://books.google.de/

Table I

Seven-level architecture OF *Historical Text Re-use* DETECTION. EACH LEVEL IS EXPRESSED BY ITS SYMBOL, FOLLOWED BY THE EXPECTED INPUT AND OUTPUT.

Level	Symbol	Input	Output
Segmentation	ξ_Θ	Digital Library D_S	Re-use Units V_{D_S}
Preprocessing	ψ_Θ	Re-use Units V_{D_S}	Cleaned Re-use Units
Featuring	μ_Θ	Cleaned Re-use Units	Digital Fingerprint
Selection	σ_Θ	Digital Fingerprint	Digital Signature
Linking	λ_Θ	Digital Signature	Candidate List
Scoring	θ_Θ	Candidate List	Result List $E^H_{D_S,\phi_\Theta}$
Postprocessing	π_Θ	Result List $E^H_{D_S,\phi_\Theta}$	Reduced Result List

Brute Force method checks all entries A_{ij} in an adjacency matrix A, the *Feature-Based Linking* only checks those entries in A which have one or more features in common. The latter method can reduce performance by ignoring function words such as *and* or *the* and by comparing those $(v_i, v_j) \in E$ that have valuable re-use candidates.

Historical texts often contain a large number of spelling mistakes, linguistic variations, dialectal variations or scribal errors. For this reason, we may understand the process of text re-use detection as an instance of a *Locality Sensitive Hashing (LSH) h* [14] (cf. eq. 2). Unlike *md5* or *crc32*, an *LSH* hash function does not aim at flipping 50% of all output bits if one input bit changes, but at mapping similar inputs to the same, or at least similar, representation. This is shown in equation 2:

$$\mathbf{Pr}_{h\in\mathcal{F}}[h(x) = h(y)] = sim(x, y), \tag{2}$$

where $sim(x, y)$ is the *Min-wise Independent Permutation* [15] computed by equation 3. A and B represent the sets of features of the two *re-use units* v_i and v_j.

$$sim(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

Like the *Brute Force* method, the feature-based linking *FBL* is also of squared complexity. While the *BFL* depends on the number of elements in V, the *FBL* depends on the frequency f of all features. The linking costs for *FBL* can be computed by equation 4:

$$fbl = \sum |f_i| \cdot \left(|f_i| - 1\right) \tag{4}$$

The Zipfian Law [16] as a power law on word-type distribution states that only function words appear frequently in a text, whereas roughly half of all words occur only once. The Zipfian law can be used for *max pruning* and *min pruning* [3]. *Max pruning* removes all frequent features such as function words. Even if it includes only a few hundred words, it significantly speeds up performance (cf. section V). *Min pruning* of rare features occurring only once helps to reduce the feature index and often makes data structures faster. Features with a frequency of 1 can be ignored for text re-use detection since re-use implies that a word needs to occur at least twice.

A *LSH* function h includes and needs, on the basis of the aforementioned *re-use diversity*, numerous parameters. We thus propose to break *LSH* for the text re-use function ϕ_Θ down into the seven levels described in table I.

A digital library can be processed by the *overlapping* or *disjoint segmentation* ξ_Θ. An *overlapping segmentation* is beneficial whenever we wish to detect text re-use on small sentence sections. An overlapping segmentation test on the *Perseus Digital Library* revealed that 80% of the identified text re-uses contained four or fewer common words. *Disjoint segmentation*, a commonly used method, splits re-use units into sentences [17], paragraphs [13] or even documents [18]. The size of the re-use units strongly depends on the research question and the underlying task.

The pre-processing function ψ_Θ is one of the most complex. The objective of this function is to form equivalence classes of similar words, which [19] calls *concepts*. Stein maps different techniques to these conceptual classes [20]. Richly annotated data such as WordNet [21], [22], supports the use of linguistic and semantic relations, including synonyms or co-hyponyms. Co-hyponyms in historical corpora are especially interesting insomuch as synonyms are more likely to vary over time. Hence, co-hyponyms provide essential data for paraphrase detection [9]. Automatic methods of forming semantically similar concept classes are known as *pLSA* [23] and *Topic Modeling* [24]. [25] describes the usage of word *co-occurrence* profiles as a means of comparing their semantic similarities. The underlying *distributional hypothesis* [26] assumes that words being used in similar contexts are semantically close.

Historical documents heavily fraught with linguistic variation lend themselves well to string processing techniques, including the *Levenshtein distance* [27] and the *FastSS* approach [28], both of which can also be applied to noisy data such as OCR output.

The featuring function μ_Θ transforms the re-use unit into features that can be compared, like *word-types* [9], [17] and *word n-grams* [18]. The *word-type feature* technique can be deployed to detect paraphrase or allusion, whereas the *n-gram* approach identifies duplicates and near-duplicates. *N-gram* techniques can be further divided into *shingling* and *hash-breaking* featuring methods [3].

The selection function σ_Θ seeks to remove from the digital fingerprint features that are considered to be irrelevant. Alongside the aforementioned *min* and *max pruning*, selection can also be achieved by adapting information retrieval techniques such as the *tf.idf* measure of term-weighting [29]. Talavera [30] describes another approach that involves feature dependencies, which occasionally lead to clusters. As per Schleimer's recommendation [31], clusters can be avoided by virtue of the winnowing algorithm, which selects features all over the re-use unit rather than solely over a local cluster. This latter avenue, however, is best suited to bigger re-use units such as *pages* or *books*.

There are two classes of linking functions λ_Θ. The *Intra Digital Library Linking* [3] maps re-use units to other units within the same database, leading to a graph $G = (V, E)$. The advantage of the *Intra Digital Library Linking* is that the feature index can be compressed by removing all features with a frequency of 1 as these do not contribute to the process. The *Inter Digital Library Linking* [3] detects re-use between at least two different textual databases resulting in a bipartite graph $G = (V_1, V_2, E)$. Features with a frequency of 1 cannot be removed from the feature index owing to the fact that the same feature could appear in the other database or digital library.

The scoring function θ_Θ investigates the *re-use overlap* of two linked re-use units. The *vector space model* [29] considers not only the size of the overlap but also the weight of its features [11], [17], [32]. [33] reports that easier methods such as *resemblance* and *containment* already provide commensurable results. Following these latter methods, all existing features of an overlap have the same weight (or are weightless). This indicates that the output of the previous processed feature selection function σ_Θ has a higher influence on the scoring. A broader overview of scoring metrics is provided in [3].

The optional post-processing π_Θ function operates on top of the re-use graph $G = (V, E)$, eliminating, for instance, noisy links. Post-processing is advantageous if the research intent is to identify single links or clusters of links that indicate a passage has been copied over from one work to the other. For these use-cases, one can algorithmically pinpoint linear sequences in the manner of dot plot view visualisations [34], [35].

IV. Methodology

Text re-use is a complex process and a catalyst for research questions that largely relate to the *quality* of algorithms. While primarily focusing on algorithm *performance*, the present paper also touches upon previously published *qualitative* experiments [9] on the *re-use style* of seven versions of the English Bible. The *n-gram shingling* approach is particularly suited to analyse minor linguistic changes, such as those introduced by Webster's revision of the King James version. For other comparisons, including for example the Young Literal Translation or the Bible in Basic English, *word-based featuring* generated acceptable results. This study was performed on 200,424 verses.

Having looked at *quality*, our attention then shifts to *performance* with a keen eye for Big Data. In practice, this entails the implementation of the featuring function μ_Θ and of the selection function σ_Θ, as well as the study of the implications to the linking function λ_Θ.

The text is split into verses. Data is pre-processed by way of lemmatisation and synonym replacement. Both synonyms [21], [22] and the *morphy* function that lemmatises English texts are taken from WordNet. Our research question on linking performance, the most time-critical component in text re-use analysis, does not call for the *scoring* and *post-processing* functions θ_Θ and π_Θ.

Here, we discuss the application of three different featuring techniques: *trigram shingling* (*tri*), *bigram shingling* (*bi*), and *word-based featuring* (*word*). Before doing so, let us introduce the three selection functions σ_Θ used. Firstly, *min pruning* removes all features with a frequency of 1; for *word-based featuring* this implies compressing the feature index by 50%; for *bigram* and *trigram shingling*, this means that approximately 75% and 87.5% of all features can be removed as they occur only once. Secondly, *max pruning* is used to progressively remove frequent words. To this end, the *Feature Density* $\mathcal{F}$ needs to be defined –as in equation 5– as describing the ratio of selected features to all features of a *re-use unit* v_i. Thirdly, we use *part of speech* (PoS) tags to single out features of higher linguistic relevance, including nouns and verbs.

$$\mathcal{F}_i = \frac{\sigma_\Theta(\mu_\Theta(v_i \in V_{D_S}))}{\mu_\Theta(v_i \in V_{D_S})} \tag{5}$$

For this PoS selection strategy[15], we selected the *Perseus Tag System* (cf. table II), which is also being used to treebank Ancient Greek and Latin.

Table II
Perseus Tag System. List of all 14 *part of speech* tags.

Part of Speech-Tag	Wortartklasse
n	noun
v	verb
t	participle
a	adjective
d	adverb
l	article
g	particle
c	conjunction
r	preposition
p	pronoun
m	numeral
i	interjection
e	exclamation
u	punctuation

[15] For more details, see http://nlp.perseus.tufts.edu/syntax/treebank/agdt/1.7/docs/README.txt

All PoS tags in table II are weighted and numerically incremented, where 1 is punctuation and 14 a noun. If multi-word features are used such as those generated by *bigram* and *trigram shingling*, we calculate the average weight.

V. Results

In Biometry and other forensic sciences, the *performance* of a method is not only a property but one possessing eight quality criteria [36]. For this reason, we set up a couple of experiments to investigate parameters and their influence on performance. In practice, however, we run four different experiments: in section V-A, we compare the linking costs in relation to different feature densities $\mathcal{F}$; section V-B describes the process of manual feature selection carried out by 24 testers; lessons learnt from this test are automatically applied on a larger scale to all investigated Bible data and described in section V-C; finally, we explore the usage of multi-word features in relation to linking costs.

A fundamental parameter of these evaluations is the *Feature Density* $\mathcal{F}$, which numerically describes how features are selected. As high *feature density* correlates to high *recall*, we also notice a progressive increase in computational time. In order to complete all experiments in a short period of time –computation would require weeks– we opt for simulation. Within the simulation, we do not process every re-use unit but, rather, count the number of processed links. In detail, we define three different metrics to measure the linking.

Firstly, we define the *Unique Linking* metric *(UL)*. *UL* measures the uniquely identified links in a digital library. Formally, it is identical to the degree or size of the set of links E in a text re-use graph $G = (V, E)$ and can therefore be expressed as $UL = |E|$. Secondly, we introduce the *Linked Links* metric *(LL)*. The idea behind *LL* is that a *re-use overlap* almost never contains a single common feature but many. On the grounds that we are investigating *feature-based linking*, it may be inferred that linking costs correlate with the size of the *re-use overlap* that justifies *LL*. For comparative purposes, we also define the *Average Links* metric (*AL*) described by $AL = \frac{LL}{UL}$ and representing the average number of links per *re-use unit*. Finally, we define the *Brute Force Score* $BFS = \frac{LL}{BFL}$ (BFL is defined in eq. 1). The rationale of the BFS is to compare *feature-based linking* with *Brute Force Linking* by way of different *Feature Densities* $\mathcal{F}_i$, as described in the introduction of this paper. The BFL of 200,424 verses belonging to seven bible editions [9] amounts to $200424^2 - 200424$ or 40.169 billion.

The decision to express the complexity via links rather than time was also dictated by the reproducibility of the following computed numbers. As different computers have different throughputs, costs cannot be effectively compared. To do so, one only need the throughput of processed number of links per second; assuming we are working with 1000 links/sec, this should be a feasible approximation. Lastly, it is important to stress that we only investigate the *featuring*, *selection*, and *linking* functions, not the *scoring* function. Henceforth, all numbers used in the following sections will include a long tail of links with a *re-use overlap* of 1 - an acceptable scenario for this type of study but one that would be avoided in a full analysis.

A. Linking by parameterisation through Feature Density $\mathcal{F}$

In a first experiment, we are interested in the dependency of the linking costs –expressed by *UL*, *LL*, and *AL*– on the *Feature Density* $\mathcal{F}$. We run *word-based featuring* on lemmatised texts, normalised at the hand of a synonym heuristic. (cf. section IV). We use *max pruning* as a selection criterion to obtain different *Feature Densities* $\mathcal{F}$. The results are presented in table III.

If we use a *Feature Density* of $\mathcal{F} = 1.0$, thus preserving all features, the *feature-based linking* generates 40 billion unique links *(UL)*. On an average of 5.11974 features per *re-use unit*, the total number of generated links is 204 billion. As the *BFL* in our data is 40,169 billion, 99.3% of all possible links are generated. This almost full linking is caused just by the function words. Working under the assumption that there are about five features in every *re-use unit*, the *BFS* amounts to 5.087 (cf. table III, last column) with the result that the linking costs are about five times higher than those sustained by the naive *Brute Force* method.

Table III
Linking analysis depending on *Feature Density* $\mathcal{F}$ for Unique Links (UL), Linked Links (LL), Average Links (AL) and the Brute Force Score (BFS). *UL* and *LL* are in million links.

	UL	**LL**	**AL**	**BFS**
$\mathcal{F} = 0.1$	161 M	165 M	1.02350	0.00411
$\mathcal{F} = 0.2$	1,099 M	1,152 M	1.04786	0.02868
$\mathcal{F} = 0.3$	3,778 M	4,198 M	1.11109	0.10452
$\mathcal{F} = 0.4$	8,347 M	10,246 M	1.22754	0.25508
$\mathcal{F} = 0.5$	15,130 M	21,558 M	1.42489	0.53669
$\mathcal{F} = 0.6$	21,687 M	37,003 M	1.70621	0.92117
$\mathcal{F} = 0.7$	29,224 M	64,521 M	2.20779	1.60623
$\mathcal{F} = 0.8$	35,294 M	106,211 M	3.00930	2.64408
$\mathcal{F} = 0.9$	38,685 M	157,160 M	4.06248	3.91241
$\mathcal{F} = 1.0$	39,914 M	204,354 M	5.11974	5.08729

Furthermore, table III shows that a *Feature Density* of $\mathcal{F} = 0.63$ is necessary to match the linking costs of the naive *Brute Force* method. If we wish to speed up the process by a factor of 10, we must reduce the *Feature Density* to $\mathcal{F} \approx 0.3$. This, however, means that the *re-use overlap*, expressed by the *AL* score, is much smaller than 2 and hence impractical.

This first experiment brings us to the conclusion that removing function words does not boost performance as significant as needed. And while it might be easy to detect or compute frequent words, sayings subsuming function or common words, like *to be, or not be*, invariably run into problems. This is a known computational problem when it comes to *re-use types* such as *Wisdoms*, *Winged Words* and *Sayings*.

Table IV
Feature Density FOR ALL *part of speech* TAGS FROM TABLE II SELECTED MANUALLY BY 24 TEST PEOPLE.

	n	v	t	a	d	l	g	c	r	p	m	u
Bible	0.98	0.86	0.81	0.95	0.69	0.39	0.71	0.70	0.72	0.56	0.80	0.58
Middle Ages	0.98	0.88	0.93	0.95	0.79	0.42	0.81	0.71	0.79	0.49	0.84	0.52

B. Manual selection of features by PoS tags

Following the results of the previous section, we set up a second set of experiments. For this purpose we asked 24 native speakers of German from different academic backgrounds, ages and social groups, to manually select the words or linguistic features forming Biblical idioms and allusions. The list of idioms was taken from [37]. The 200 datasets of idioms were processed by all 24 candidates. In total, we collected 4,800 manually analysed datasets. To form a comparison with another dataset, we took medieval idioms from [38], which contains 202 datasets, also analysed by our 24 testers. The 402 (200+202) datasets had been manually tagged beforehand by the *Perseus Tag System*, described in table II.

Bearing in mind that the average word length for the Biblical and Medieval *Idioms* is 3.99 and 3.77, candidates were initially asked to select and delete those words that clearly do not define an idiom. We then proceeded to compile a list of all words that were systematically deleted, with their corresponding frequency of deletion across all testees:

- **Bible**: *ein* (563), *die* (276), *das* (193), *sein* (176), *den* (170), *der* (169), *wie* (131), *und* (127), *im* (107), *ist* (105), *etwas* (94), *einen* (93), *in* (92), *eine* (88), *auf* (78), *sich* (76), *sein* (73), *jemanden* (71), *haben* (58), *!* (55), *,* (50), *von* (46), *vom* (43), *jemandem* (42), *gehen* (41), *das* (38), *machen* (38), *werden* (38), *dem* (37), *mit* (37)
- **Middle Ages**: *ein* (563), *die* (276), *das* (193), *sein* (186), *einen* (172), *ein* (140), *und* (117), *sich* (111), *haben* (107), *auf* (98), *dem* (93), *!* (85), *der* (77), *,* (75), *eine* (64), *mit* (64), *jemandem* (59), *jemanden* (46), *in* (40), *ins* (40), *am* (38), *kommen* (37), *einer* (35), *machen* (35), *wie* (34), *aus* (33), *es* (31), *das* (30), *legen* (29)

These lists contain mainly function words, as well as some common German verbs. This indicates that these are the words that could be deleted first even in an automatic system.

Our second topic of interest concerns the *Feature Density* $\mathcal{F}$ score describing how many words were manually selected. We inherit this from observing the results of the previous section, where it is evident that a performance boost would result in a very low *Feature Density* of $\mathcal{F} \leq 0.3$. The manually annotated data of all 200 Biblical and 202 Medieval *idioms* across all 24 candidates shows an average *Feature Density* of $\mathcal{F}^B = 0.7585$ on a word level for the Bible *idioms* and of $\mathcal{F}^M = 0.7699$ for the Medieval *idioms*. The standard deviations are $\sigma_B = 0.1367$ and $\sigma_M = 0.1435$. These manual results lead us to assume that human selection is not sufficiently strict to speed up the text re-use process (more in section V-C).

Thirdly, we asked ourselves which *part of speech* tags would be consistently selected by human reviewers and which would be considered irrelevant. For this purpose, all 402 *idioms* had been manually tagged beforehand by the *Perseus Tag System* (cf. table II) and the probability of a *PoS* tag $P_c(tag)$ occurring in these datasets computed together with the probability of the reduced *idioms* $P_s(tag)$. Finally, the manual *Feature Density*, defined by $\mathcal{F}_{tag} = P_s(tag)/P_c(tag)$, is also computed. As a baseline, we took the *Feature Densities* $\mathcal{F}^B = 0.7585$ and $\mathcal{F}^M = 0.7699$ on a word level from the first experiments in this section. The results of these experiments are shown in table IV. Cells with a black background show a high density of *part of speech* tags, a grey background displays weak density and a white background no relevant density at all.

In both the biblical and medieval datasets, nouns (*n*), adjectives (*a*), and verbs (*v*) are considered by our 24 testers to be strongly significant for the creation of idioms (cf. table IV). The results for verbs could be improved if *auxiliary verbs*, such as *have*, *be*, and *will*, were considered as a separate class. If auxiliary verbs are removed, verbs on both test-sets reach values close to 99%. Surprisingly, testers also identified numerals (*m*) as particularly significant.

C. Automatic selection of features by PoS tags

A manual selection of features based on *idioms* computes the *Feature Densities* $\mathcal{F}^B = 0.7585$ and $\mathcal{F}^M = 0.7699$ with an average word length of 3.99 and 3.77 per *idiom*. That is, every idiom loses one word.

At this point, there arises the question: What will the *Feature Density* $\mathcal{F}$ be –on a *part of speech* tag level– if we apply the results to the frame of a verse containing more than the average four words of an idiom? Both medieval and biblical *idioms* bear the same density results in terms of *part of speech* tags for nouns (*n*), verbs (*v*), and adjectives (*a*). For this reason, we focus on the automatic analysis of these *PoS* tags. A second run additionally includes the participle (*t*) and numeral (*m*) word classes.

In the first run where we only select nouns (*n*), verbs (*v*) and adjectives (*a*), we can compute a *Feature Density* of $\mathcal{F} = 0.354$ close enough to the expected *Feature Density* of $\mathcal{F} \approx 0.3$ needed to increase the speed by a factor 10. Table V, however, shows that the *BFS* for these word classes is still larger than 1: on the one hand this leads to the

Table V
LINKING ANALYSIS ON THE BASIS OF *part of speech* TAGS. *UL* AND *LL* ARE IN MILLION LINKS.

	UL	LL	AL	BFS
n, a, v	26,732 M	45,260 M	1.69305	1.12673
n, a, v, t, m	27,606 M	51,091 M	1.85072	1.27189

selection of a smaller set of features but on the other it slows the analysis more than the *Brute Force method* would. The *Feature Density* of the second row of the table, when also including numerals (*m*) and participles (*t*), is $\mathcal{F} = 0.438$.

To sum up our research on including *part of speech* tags as part of feature selection, it seems fair to conclude that *PoS* tags greatly help to reduce the number of relevant features, while also improving the *precision* by about 4.5%. The speed, measured by the *BFS* score, is too slow even if the *Feature Density* is considerably smaller.

D. Use of multi-word features

Something that we noticed is that neither by reducing the *Feature Density* by removing frequent words, or by using the linguistic knowledge given to us by the *part of speech* tags, does the performance improve well enough. Given $O(n^2)$ for text re-use, it would be our aim to improve speed not by 30%, but, if possible, by several orders of magnitude.

We thus asked ourselves why it appears impossible to increase the speed by the aforementioned method. When using *feature-based linking*, the outcome of the analysis strongly depends on the word frequency. (cf. equation 4). Pre-processing is crucial to decontaminate the text from language variants and the changes that come with the inevitable evolution of language. The processes of normalising the language and that of lemmatising different semantic variants (for example, synonyms) reduce the number of word-types in a digital library, thus increasing the word frequency. This is where the problem lies. Often there are word tokens that appear only five or six times (cf. Zipfian Law) that are mapped by an *LSH* function, such as lemmatisation, to a concept in the text whose frequency is in the hundreds. This means that the cost for processing just the one feature is now as high as 10,000 or more.

It is clear that pre-processing, while necessary, has a negative impact on the speed of text re-use detection, conducted through *feature-based linking*. Since using weaker pre-processing techniques is not an option, we must seek other solutions in order to reduce feature frequency.

It is for this reason that we investigate *multi-word features*. Words follow a *Power Law*, namely Zipfian's Law, described by equation 6.

$$f = \frac{c}{r} \tag{6}$$

f is the frequency, r is the rank, and c is a constant. When working with bigrams, trigrams or co-occurrences, all known as *multi-word features* of n words, the *Power Law* can be described by equation 7.

$$f = \frac{c}{r^n} \tag{7}$$

For bigrams and co-occurrences n is 2, for trigrams it is 3. Equation 7 implies that n and f are inversely proportional. *Multi-word featuring* responds well since text re-use typically contains more than just two or three words in *re-use overlaps*. For this reason, *multi-word featuring* can be understood as clustering words as "bigger" features, which are less frequent and are more descriptive.

An initial *multi-word featuring* examines bi- and trigrams. When using n-grams, "clustering" recognises neighbouring words as one feature. For this test series, we considered a *Feature Density* of $\mathcal{F} = 0.8$ (compare results in table III with the *word* row in table VI). Taking this as a performance baseline, we also computed bi- and trigrams. The power of n in equation 7 illustrates in table VI the costs of *Linked Links* and, therefore, that the *BFS* score is logarithmically inversely proportional to the number of words in a *multi-word feature*. More specifically, table VI shows that $n = 2$ increases the speed by a factor of 35 (*BFS* score) and that $n = 3$ increases the speed by a factor of 662. The drawback of n-grams, however, is that the featuring in question is only feasible for duplicate or near-duplicate detection. Allusions or paraphrases, in fact, are excluded. *Co-occurrences* work well as *multi-word features* for them even if they were not included in the aforementioned test.

Table VI
LINKING ANALYSIS ON THE BASIS OF *n-grams*. *UL* AND *LL* ARE IN MILLION LINKS (M).

	UL	**LL**	**AL**	**BFS**
Tri	123 M	160 M	1.303	0.00399
Bi	2,531 M	3,030 M	1.197	0.07543
Word	35,294 M	10,6211 M	3.009	2.64408

Our last test combines the performance improvements brought about by *multi-word featuring* with the well-performing attributes of *part of speech* tags to reduce the *Feature Density* $\mathcal{F}$. In particular, we asked ourselves two questions: First, what are relevant *PoS* patterns? And second, what is a good length for *multi-word features* or, in other words, what is a good n? To answer both these questions, we re-analysed the manually selected data described in sections V-B and V-C and focused on *multi-word features* with $n \in [2, 4]$. Furthermore, we not only considered n-gram patterns but also looked at complex linguistic patterns that do not occur side-by-side.

On the whole, nine patterns were identified as significant. No pattern is available for $n = 4$ due to the dependency on idiom length. On an $n = 2$ level, we extracted the following five patterns: **n v** and **[a|l|r|m] n**. The first pattern **n v** is semantic, the other four syntactic. For $n = 3$, four patterns emerged: **[l|r] n v** and **n [r|c] n**.

With the exception of **n v** relations, all other patterns can be easily detected with bi- or trigrams. The **n v**, however, is the most dominant pattern of all in that it already converts 40% of all patterns. As previously shown, seeing as a higher n greatly increases performance, we also propose to use non-static featuring methods, including n of 2 or 3. Results show, however, that features with a non-static length better fulfill the needs of accuracy and performance.

VI. Conclusion

During our research, our focus was particularly on the *precision* and *recall* of algorithms of a particular dataset. We also focused the often easily ignored matter of performance. When dealing with *Big Data*, performance issues progressively increase with the increase of data. With this paper we aim to contribute to the topic of mining algorithms as a whole without neglecting performance. We believe it is not merely enough to develop methods that reach high levels of *precision* and *recall* values - acceptable performance metrics must also be achieved. The importance of performance is also something seen in Forensic biometry, where good performance levels are one of its fundamental criteria [36].

Furthermore, this paper can be concluded by several experiments that aim at the improvement of performance without losing quality of outcome. Though simple, the pruning of algorithms on word frequencies has unsatisfactory results. *Part of speech* tags reduce the *Feature Density*, but not the performance. *Multi-word features*, however, show that performance boosts are possible while still allowing the data to be increased by several orders of magnitude.

The results of this paper show that, when dealing with large sets of data, it is crucial to have a clear understanding of the re-use process. This paper is just the beginning of research efforts on the topic of the primitives of text re-use, the *Minutiae*. This would include research in the length of linguistic features and their stability. The extracted linguistic patterns are not unexpected since we assimilate most during childhood.

Acknowledgments

This work has been made available by the early career research group *eTRAP* (No. 01UG1409) of the German Ministry of Education and Research.

References

[1] U. Rieß, *Big Data*. CT 15/2013 Sonderbeilage des Heise Zeitschriften Verlages, 2013, ch. Big Data bestimmt die IT-Welt - Das verbirgt sich hinter dem Hype.

[2] G. Crane, “What do you do with a million books?” *D-Lib Magazine*, vol. 12, no. 3, March 2006. [Online]. Available: http://www.dlib.org/dlib/march06/crane/03crane.html

[3] M. Büchler, “Informationstechnische Aspekte des Historischen Text Re-use (Engl. Computational Aspects of Historical Text Re-use Detection,” Ph.D. dissertation, Universität Leipzig, 2013.

[4] M. Berti, M. Romanello, A. Babeu, and G. Crane, “Collecting fragmentary authors in a digital library,” in *Fred Heath, Mary Lynn, Rice-Lively, Richard Furuta: Proceedings of the 2009 Joint International Conference on Digital Libraries, JCDL 2009, Austin, TX, USA, June 15-19, 2009*, 2009, pp. 259–262.

[5] A. Trachsel, “Collecting fragments today: What status will a fragment have in the era of digital philology ?” in *Lire demain - Reading tomorrow*, C. C. et al., Ed. PPUR Presses polytechniques, 2012, pp. 415–429.

[6] N. Coffee, J.-P. Koenig, S. Poornima, C. Forstall, R. Ossewaarde, and S. Jacobson, “The tesserae project: Intertextual analysis of latin poetry,” in *Poster presented at Digital Humanities 2011*. Stanford University, 2011.

[7] M. Büchler, G. Crane, and G. Heyer, “Historical relevance feedback detection by text re-use mining,” in *In Maximilian Schich et al.: Arts, Humanities, and Complex Networks Living Companion at Arts, Humanities, and Complex Networks - 3rd Leonardo satellite symposium hosted by NetSci2012*, Evanston, IL, USA, 2012.

[8] M. Büchler, G. Crane, M. Moritz, and A. Babeu, “Increasing recall for text re-use in historical documents to support research in the humanities,” in *Theory and Practice of Digital Libraries 2012*, G. Buchanan, E. Rasmussen, and F. Loizides, Eds., 09 2012.

[9] M. Büchler, P. R. Burns, G. Crane, M. Mueller, and G. Heyer, “One Step Closer To Paraphrase Detection On Historical Texts: About The Quality of Text Re-use Techniques and the Ability to Learn Paradigmatic Relations,” in *Proceedings of the 2011 Chicago Colloquium on Digital Humanities and Computer Science. Chicago, 2012*, 2011.

[10] D. E. Knuth, *Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition)*, 3rd ed. Addison-Wesley Professional, Jul. 1997. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0201896834

[11] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel, “Similarity measures for tracking information flow,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, ser. CIKM ’05. New York, NY, USA: ACM, 2005, pp. 517–524. [Online]. Available: http://doi.acm.org/10.1145/1099554.1099695

[12] M. Bendersky and W. B. Croft, “Finding text reuse on the web,” in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, ser. WSDM ’09. New York, NY, USA: ACM, 2009, pp. 262–271. [Online]. Available: http://doi.acm.org/10.1145/1498759.1498835

[13] J. Seo and W. B. Croft, “Local text reuse detection,” in *SIGIR ’08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2008, pp. 571–578.

[14] M. Charikar, “Similarity estimation techniques from rounding algorithms,” in *STOC*, J. H. Reif, Ed. ACM, 2002, pp. 380–388.

[15] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, “Min-wise Independent Permutations,” *Journal of Computer and System Sciences*, vol. 60, pp. 327–336, 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.8215

[16] G. Zipf, “Human behaviour and the principle of least-effort.” Cambridge, MA: Addison-Wesley, 1949. [Online]. Available: /brokenurl#http://publication.wilsonwong.me/load.php?id=233281783

[17] R. Hose, “Cs490 final report: Investigation of sentence level text reuse algorithms,” 2004.

[18] D. Smith, R. Cordell, and E. Dillon, “Infectious texts: Modeling text reuse in nineteenth-century newspapers,” in *Big Data, 2013 IEEE International Conference on*, Oct 2013, pp. 86–94.

[19] G. Heyer, “Analyse von Bedeutungsveränderungen in diachronen Textkorpora,” Natural Language Processing Group, University of Leipzig, Germany, Tech. Rep., Februar 2009, vortrag im Forschungsseminar, Leipzig, Germany, February, 2009.

[20] B. Stein, “Principles of hash-based text retrieval,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’07. New York, NY, USA: ACM, 2007, pp. 527–534. [Online]. Available: http://doi.acm.org/10.1145/1277741.1277832

[21] G. A. Miller, “Wordnet: a lexical database for english,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[22] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[23] T. Hofmann, “Probabilistic latent semantic analysis,” in *UAI*, K. B. Laskey and H. Prade, Eds. Morgan Kaufmann, 1999, pp. 289–296.

[24] D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in *Proceedings of the 23rd international conference on Machine learning*, ser. ICML ’06. New York, NY, USA: ACM, 2006, pp. 113–120. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143859

[25] S. Bordag, “Elements of Knowledge-free and Unsupervised Lexical Acquisition,” Ph.D. dissertation, Universität Leipzig, 2007.

[26] Z. Harris, “Distributional structure,” *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[27] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” Tech. Rep. 8, 1966.

[28] T. Bocek, E. Hunt, and B. Stiller, “Fast Similarity Search in Large Dictionaries,” Department of Informatics, University of Zurich, Tech. Rep. ifi-2007.02, April 2007, http://fastss.csg.uzh.ch/.

[29] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[30] L. Talavera, C. Nord, and J. Girona, “Dependency-based feature selection for clustering symbolic data,” 2000.

[31] S. Schleimer, D. S. Wilkerson, and A. Aiken, “Winnowing: local algorithms for document fingerprinting,” in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’03. New York, NY, USA: ACM, 2003, pp. 76–85. [Online]. Available: http://doi.acm.org/10.1145/872757.872770

[32] A. Barrón-Cedeño, C. Basile, M. Degli Esposti, and P. Rosso, “Word length n-grams for text re-use detection,” in *Proceedings of the 11th international conference on Computational Linguistics and Intelligent Text Processing*, ser. CICLing’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 687–699. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12116-6_58

[33] A. Z. Broder, “On the resemblance and containment of documents,” in *In Compression and Complexity of Sequences (SEQUENCES97*. IEEE Computer Society, 1997, pp. 21–29.

[34] J. Lee, “A computational model of text reuse in ancient literary texts,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 472–479. [Online]. Available: http://www.aclweb.org/anthology/P07-1060

[35] S. Jänicke, M. Büchler, and G. Scheuermann, “Visualizations for text re-use,” in *In Proceedings of the 5th International Conference on Information Visualization Theory and Applications*, ser. IVAPP 2014, 2014.

[36] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2009.

[37] G. Wagner, *Wer’s glaubt wird selig! Redewendungen aus der Bibel.* WBG, Wissenschaftliche Buchgesellschaft, Darmstadt, Germany, 2011. [Online]. Available: http://d-nb.info/1011706148/04

[38] G. Wagner, *Das geht auf keine Kuhhaut: Redewendungen aus dem Mittelalter*. WBG, Wissenschaftliche Buchgesellschaft, Darmstadt, Germany, 2011. [Online]. Available: http://d-nb.info/1011706148/04