
# Evaluating Document Similarity Measures on Wikipedia

**Bachelorarbeit**

vorgelegt von

**Malte Schwarzer**

Matrikel-Nr.: 340819

zur Erlangung des akademischen Grades

Bachelor of Science

im Studiengang Wirtschaftsinformatik

Gutachter:

Prof. Dr. rer. nat. Volker Markl

Jun.-Prof. Dr. Bela Gipp

Betreuer:

Moritz Schubotz

Norman Meuschke

Bearbeitungszeitraum: 14.02.–14.06.2015

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Datum Unterschrift

## Abstract

Literature recommender systems play an important role for scientists as the number of publications increases every year. Current systems depend on document similarity measures to automatically identify topically related documents. The underlying concept is that two documents are more likely to be topically related the more similar they are. For scientific literature, the approach of Co-Citation (CoCit) has proved to be useful for scientists. CoCit defines document similarity as the frequency of two documents cited together by other documents. Recently, Co-Citation Proximity Analysis (CPA) was introduced to enhance CoCit by considering also proximity of citations.

This thesis evaluates the citation-based document similarity measures CoCit and CPA as well as the well-established text-based similarity measure MoreLikeThis (MLT) from the Apache Lucene framework on the test collection of Wikipedia. First, we test how suitable CoCit and CPA are to identify related Wikipedia articles, given that the measures were designed for academic articles. Second, we quantitatively and qualitatively compare the performance of citation-based similarity measures with MLT as a baseline. Third, we investigate the appropriateness of user-generated literature recommendations from Wikipedia's "See also" sections as quasi-gold standard for topically related articles that allows performing automated large-scale evaluations.

Our large-scale quantitative evaluation proves an advantage of MLT in recommendation quality. However, when analysing samples, CPA performs qualitatively equivalent to MLT, whereas CPA offers room for technology improvement. Furthermore, we show that "See also" recommendations meet the requirements of a quasi-gold standard for this evaluation.

## Zusammenfassung

Literaturempfehlungssysteme spielen, aufgrund der stetig anwachsenden Anzahl an Publikationen, eine immer größere Rolle für Wissenschaftler. Gängige Empfehlungssysteme beruhen auf Dokumentähnlichkeitsmaßen zum automatisieren Finden von thematisch verwandten Dokumenten. Die zugrundeliegende Idee ist, dass Dokumente stärker thematisch verwandt sind, je ähnlicher sie sich sind. Für wissenschaftliche Arbeiten hat sich das Verfahren der Kozitation (CoCit) als nützlich für die Bestimmung von Dokumentähnlichkeit erwiesen. CoCit misst die Dokumentähnlichkeit eines Dokumentenpaars über die Anzahl der gemeinsamen Zitationen des Dokumentenpaares in anderen Quellen. Das kürzlich entwickelte Verfahren der Co-Citation Proximity Analysis (CPA) erweitert CoCit, indem es auch die textliche Nähe von Zitaten berücksichtigt.

Die vorliegende Arbeit evaluiert die Empfehlungsqualität der zitationsbasierten Verfahren CoCit und CPA sowie des weitverbreiteten textbasierten Verfahrens MoreLikeThis (MLT) aus dem Apache Lucene Framework anhand des Textkorpus von Wikipedia. Erstens testen wir, ob die ursprünglich für wissenschaftliche Artikel entwickelten Verfahren CoCit und CPA auch auf Wikipedia Artikel sinnvoll angewandt werden können. Zweitens, vergleichen wir die Empfehlungsqualität von CoCit und CPA quantitativ und qualitativ mit MLT als Vergleichsmaßstab. Drittens, untersuchen wir, ob sich die benutzergenerierten Literaturempfehlungen aus der „Siehe auch“ Rubrik von Wikipedia als Quasi-Goldstandard für eine groß angelegte, automatisierte Auswertung eignen.

Zwar zeigt unsere groß angelegte, quantitative Auswertung Vorteile der Empfehlungsqualität zugunsten von MLT, dennoch ergibt die Analyse von Stichproben, dass CPA qualitativ dem Niveau von MLT entspricht, aber CPA gleichzeitig technologisches Verbesserungspotential bietet. Des Weiteren konnten wir die Erfüllung der Auswertungsanforderungen an einen Quasi-Goldstandard durch „Siehe auch“-Literaturempfehlungen darlegen.

# Contents

# 1 Introduction & Goals

Literature research is an important task of scientific work. Finding relevant information and related papers is essential for scientific success. The increasing number and availability of scientific papers makes literature research even more important. In 2012 the number of publications reached 1.8 million a year [1]. Traditional methods of library research cannot handle such an amount of documents. Consequently, literature recommender systems had been developed to help researchers finding relevant papers. These systems use automated approaches to determine document relevance.

However, determining relevance is a complex challenge due to its nature. Relevance consists of two main components [2]:

1. Commonly objective topical relevance
2. Purely subjective user relevance

Domain experts can judge the topical relevance of a document. On contrary user relevance highly depends on the users information need. This can differ from user to user and makes automation challenging.

Current recommender systems identify topically relevant documents by using document similarity as approximation of relevance [3]. The idea is that documents, which are similar, are more likely to cover a related topic. There are several concepts of document similarity. One concept rests on the document text and words it contains. Two documents are considered as similar when their vocabulary overlaps [4]. A second well-established and helpful concept, namely Co-Citation (CoCit), is based on citation and references within documents [5], [6], [7]. Recently, a modification of CoCit called Co-Citation Proximity Analysis (CPA) has been introduced [8]. This method has already been evaluated and proved to enhance the traditional CoCit in the domain of scientific literature [9].

In this thesis, we continue the investigation of recommendation quality differences between CoCit and CPA by testing them with a different document type:

Besides traditional scientific publications another information source, which also demands good recommender systems, has become popular in the academic community: Wikipedia [10]. Not only its popularity makes Wikipedia an interesting research subject, but also its text corpus is fully available. Many commonly used test collections in Information Retrieval (IR) like the standard TREC [11] facing the problem of incompleteness, because they are samples. Therefore, some citation might point to papers, which are not part of the test collection. In contrast, Wikipedia is a closed test collection. Thus, a complete citation graph can be evaluated.

Furthermore, many Wikipedia articles contain user-generated literature recommendations in so-called “See also” section. We utilise these recommendations for a large-scale evaluation of

document similarity measures. Thereby, we are able to take a large number of results into account. Such a large number would be unfeasible to compare in a traditional user study.

For comparison purpose, we define Apache Lucene's MoreLikeThis (MLT) as a third and baseline similarity measure. Beside Lucene's usage in many website, e.g. Twitter [12], it is also used as baseline in other IR studies [13]–[15].

Summing up, the goals of this work are to test the following research questions:

1. Given that their primary area of application are academic articles, how suitable are the citation-based similarity measures Co-Citation (CoCit) and Co-Citation Proximity Analysis (CPA) to identify related articles in Wikipedia?
2. How does the performance of CoCit and CPA in identifying related Wikipedia articles compare to the performance of a typical text-based similarity measure applied for the same task?
3. Can "See also" sections of Wikipedia articles serve as an approximation gold standard for topically related articles that allows performing automated large-scale evaluations of document similarity measures?

This thesis is structured as follows: First, we start by introducing the background necessary for this work. We explain the concept of document similarity, how similarity can be measured, which measures we focus on and how they work. In addition, we provide a short survey of related work. The third chapter presents our methodology, e.g. the information needs of our experiment, information about Wikipedia as test collection, our gold standard and how we measure the recommendation quality. Chapter 4 explains the implementation of the document similarity measures and the technology we use. In Chapter 5, we present the results in a quantitative and qualitative evaluation. At the end, in Chapter 6 we discuss our results and propose future work.

# 2 Background & Related Work

In this section, we start with an introduction of document similarity as terminology in the field of IR. We continue with a description of concepts of the investigated document similarity measures. Lastly, we close this section with a survey of related work.

## 2.1 Dimensions of Similarity

Many IR tasks, especially those related to text mining, involve finding documents that are similar to each other. Typical applications are listed below:

1. Clustering groups sets of documents in such a way that similar documents are in the same group.
2. Plagiarism detection fights fraud in the scientific and commercial community based on the similarity of work.
3. Literature recommender systems use similarity, as well, to provide relevant documents to its users.

Defining document similarity, thereby, is essential for all these IR tasks, since similarity can be understood in different ways.

For instance, a black bird and a black cat are similar in respect of their colour. The black cat is also similar to a white cat, since they are both cats. But the black bird is not similar to the white cat. Similarity, therefore, depends on its definition. Same patterns can be recognised in document similarity. Documents might be declared as similar, when they, e.g. cover the same topic, use a common set of words or are written in the same font. In IR the dimension of similarity defines the understanding of similarity. We distinguish between the following dimensions: lexical, structural and semantic document similarity.

Moreover, similarity is not a binary decision. In many cases declaring two things as similar or not, is not suitable. Instead, the degree of similarity is measured. Even if the black bird is neither white nor a cat, it is an animal. Consequently, the black bird is in some way also similar to the white cat, but not as similar as the black cat is to the white cat. Similarity measures express therefore document similarity as normalised scalar score [16], which is within an interval of zero to one. The highest degree of similarity is measured as one. When two objects are not at all similar, the degree of similarity is zero.

### 2.1.1 Lexical Similarity

The lexical document similarity of two documents depends on the words, which occur in the document text. A total overlap between vocabularies would result in a lexical similarity of 1, whereas 0 means both documents share no words. This dimension of similarity can be calculated

by a simple word-to-word comparison. Methods like stemming or stop word removal increase the effectiveness of lexical similarity [17], [18].

### 2.1.2 Structural Similarity

Whereas lexical similarity focuses on the vocabulary, structural similarity describes the conceptual composition of two documents. Structural document similarity stretches from graphical components, like text layout, over similarities in the composition of text segments, e.g., paragraphs or sentences that are lexical similar, to the arrangement of citations and hyperlinks [19].

Structural similarity is mainly used for semi-structured document formats, such as XML or HTML. A common, yet expensive in computing time, approach is to calculate the minimum cost edit distance between any two document structures [20]. The cost edit distance is the number of actions that are required to change a document so it is identical to another document.

### 2.1.3 Semantic Similarity

Two documents are considered as semantically similar when they cover related topics or have the same semantic meaning. A proper determination of the semantic similarity is essential for many IR systems, since in many use cases the users information need is rather about the semantic meaning than the vocabulary or structure of a document. However, measuring topical relatedness is a complex task to accomplish. Therefore, lexical or structural similarity is often used to approximate semantic similarity. The example below shows that approximating semantic by lexical similarity is not always suitable.

1. Our earth is round.
2. The world is a globe.

Even if both sentences are lexically different, because they have only one out of five words in common, the semantic meaning of the sentences is synonymous.

Some approaches (e.g. WordNet [21], Latent Semantic Analysis [22]) try to overcome this issue by comparing rather concepts or word-to-word relationships than single words to measure semantic similarity. The words “earth” and “world” can be interpreted as same concept. The semantic similarity of both sentences therefore is recognisable. But in many cases a single word, on its own, has only little semantic meaning, because the meaning is created through the connection between words. Lexical similarity treats words as though they are independent of one another and thereby misses those semantic connections. Furthermore, synonymies and ambiguities need to be resolved to capture a topic based on words.

## 2.2 Document Similarity Measures

After introducing the fundamental concepts of lexical, structural and semantic document similarity, we now point out approaches to compute similarity. The document similarity measures are divided in two categories depending on the information they are based on.

### 2.2.1 Text-based Document Similarity

The most intuitive method to measure similarity of documents is to use the document text. Word-to-word similarity measures can easily determine the lexical similarity, but a simple counting of overlapping vocabulary does not necessary lead to a correct semantic similarity. Thereby, extensive research in this IR had been done to improve semantic detection in text-based similarity measures.

The Vector Space Model (VSM), introduced by Salton, Wong and Yang [23], is the state-of-the-art approach for IR. In VSM, documents are organised as vectors in a so-called term-document matrix (TDM) as illustrated in Figure 2.1, where a row represents term $\boldsymbol{t_i} = w_{i,*}$ that occurs in the document collection and a column represents a document $\boldsymbol{d_j} = w_{*,j}$. Matrix element $w_{i,j}$ is the count of term $\boldsymbol{t_i}$ in document $\boldsymbol{d_j}$.

| | $\boldsymbol{d_1}$ | $\boldsymbol{d_2}$ | ... | $\boldsymbol{d_m}$ |
|---|---|---|---|---|
| $\boldsymbol{t_1}$ | $w_{1,1}$ | $w_{1,2}$ | ... | $w_{1,m}$ |
| $\boldsymbol{t_2}$ | $w_{2,1}$ | $w_{2,2}$ | ... | $w_{2,m}$ |
| ... | ... | ... | ... | ... |
| $\boldsymbol{t_n}$ | $w_{n,1}$ | $w_{n,2}$ | ... | $w_{n,m}$ |

**Figure 2.1: Concept of TDM.**

| | $\boldsymbol{d_1}$ | $\boldsymbol{d_2}$ | $\boldsymbol{d_3}$ | $\boldsymbol{d_4}$ |
|---|---|---|---|---|
| **car** | 1 | 5 | 3 | 0 |
| **truck** | 9 | 4 | 3 | 1 |
| **flower** | 0 | 0 | 0 | 4 |

**Figure 2.2: Example of TDM.**

Figure 2.2 shows an example of TDM for the documents $\boldsymbol{d}_{1\text{-}4}$ containing the terms “car”, “truck” and “flower”.

In general more sophisticated methods exist to determine $w_{i,j}$. A well-established approach is Term Frequency – Inverse Document Frequency (TF-IDF) by Sparck Jones [24]. TF-IDF normalises the weights $\widehat{w}_{i,j}$ by the overall term frequencies within the corpus for example as shown in Equation 2.1.

$$\widehat{w}_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{m} w_{i,k}} \qquad (2.1)$$

Based on this matrix, VSM can be used to retrieve ranked results depending on a query, while a query can be both, a term or document. A term query is the typical input of search engines, e.g. keywords, whereas a document query is used to find similar documents. Both queries are also represented as vectors in the TDM. The matching of a query with documents from the collect is quantified by calculating the Cosine similarity of both vectors as the example below illustrates:

**Document vector** $\boldsymbol{d_2} = (5 \quad 4 \quad 0)$

**Query vector** $\boldsymbol{q} = (0 \quad 1 \quad 0)$

**Cosine similarity** $$\text{sim}(\boldsymbol{d_2}, \boldsymbol{q}) = \frac{\boldsymbol{d_2} \cdot \boldsymbol{q}}{|\boldsymbol{d_2}||\boldsymbol{q}|} = \frac{4}{\sqrt{25+16}} \cong 0.62$$

**Figure 2.3: Example of computing Cosine similarity of a document a query vector.**

In Figure 2.3 the term query $\boldsymbol{q}$ stands for "truck" and $\boldsymbol{d_2}$ for the document in Figure 2.2. For visualisation vectors are written as row and not as column vectors. $\cdot$ denotes the dot product and $||$ the Euclidean norm. The resulting Cosine similarity is approximately 0.62 and determines how relevant the document for this query is. The same methods can be used to compute the text-based similarity of two documents.

Implementations of this concept are used in Apache Lucene and many other modern IR systems [25], [26]. Aside query-based search functionalities, Lucene also consists of component to retrieve similar documents. The component's name is "MoreLikeThis" (MLT). We choose MLT as baseline measure of this research, because it has been proven to be success in similar use cases [13]–[15]. We explain MLT from the technology perspective in Section 4.1.4.

### 2.2.2 Citation-based Document Similarity

The second approach, to measure document similarity is built upon the structural element of citations. Scientific documents and other types of documents (e.g. websites) can be understood as objects in an network, which are connected by citations or hyperlinks. The citation network can be used to calculate similarity between documents independent from the lexical, syntactical and style characteristics of a text. Therefore, citation-based similarity measures can be applied without any knowledge about the document's language.

Before introducing the concepts of Direct Citation, Bibliographic Coupling, Co-Citation and Co-Citation Proximity Analysis, we clarify the terminology in the context of library and information science.

The descriptions in sections 2.2.2.1 - 2.2.2.6 closely follow the work of Gipp [27].

#### 2.2.2.1 Citation Terminology

Referring to related work in publications has long tradition in scientific history. These references are used to acknowledge concepts or methods that were used by the author and other reasons [28]. In this context the terms citation and reference are often used inconsistently [29]. To clarify we comply the definition by Egghe and Rousseau:

*"If paper R contains a bibliographic note using and describing paper C, then R contains a reference to C and C has a citation from R. Stated otherwise, a reference is the acknowledgement that one document gives to another, while a citation is the acknowledgment that on document receives from another."* [30]

In other words, paper R cites C, while C is cited-by R. Both papers are in a directional relationship. Furthermore, as one of the three citation-based measures is about the proximity of citation, we need to define the position, where a source is cited in the text, as a citation marker [27], [31], [32].

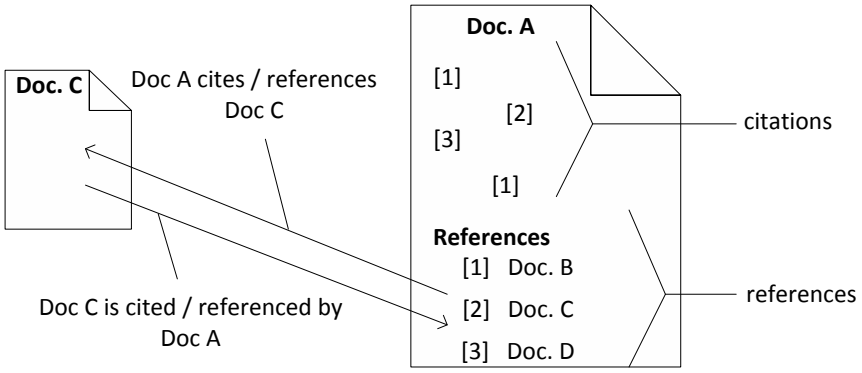



**Figure 2.4: Citations and references in scientific documents. Source [27].**

Figure 2.4 illustrates both definitions. There is a 1:1 relationship between a document and each of its references and a 1:n relationship between a single reference and corresponding citation markers.

#### 2.2.2.2 Link Terminology

The documents evaluated in this thesis are webpages, more specially Wikipedia articles. Webpages commonly do not contain academic citations. For webpages, links are the equivalent of citations in academic documents. However, the motivation of making a link on a web page is different from the motivation behind citing a scientific article, even if their concepts may seem very similar [33]. In general, links and citations serve the purpose of acknowledgment and are therefore analysed for relevance judgements by many algorithms, e.g. PageRank [34]. In the context of our test collection, the internal links of Wikipedia have mainly the purpose of creating "relevant connections to the subject of another article that will help readers understand the article more fully" [35]. Consequently, the internal Wikipedia links are as well a judgement of relevance.

As a result: When speaking about citation-based document similarity or document similarity measures, we use the terms "link" and "citation" interchangeable.

In addition, we distinguish between the directions of a link. An inbound link is a link that a webpages receives from another webpage; an outbound link is a link that a webpage gives to another webpage.

#### 2.2.2.3 Direct Citation

The concept of Direct Citation describes the straightforward relationship of two documents that are directly connected by a citation. Two documents are considered similar if one cites the other. Figure 2.5 shows that each citation relationship is bidirectional as an earlier document is cited by a new document.

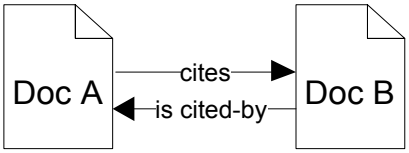



**Figure 2.5: Direct Citation: Doc A cites Doc B, while Doc B is cited-by Doc A. Source** [27]

Even if a Direct Citation clearly indicates a topical relatedness of documents, the method is inapplicable as a general document similarity measure, since it is limited to bidirectional citation relationship. Thereby, Direct Citation cannot measure the degree of document similarity, it only measures if a relationship of two documents exists or not.

Nonetheless, a document's number of "cited-by" is often used as indicator for popularity or to rank results in scientific search engines.

#### 2.2.2.4 Bibliographic Coupling

Bibliographic coupling, established by Kessler [36], uses citation analysis to determine a similarity relationship between two documents. Documents are bibliographically coupled, if they cite one or more documents in common. The basic idea is that documents that cite the same works are more likely to have to same subject.

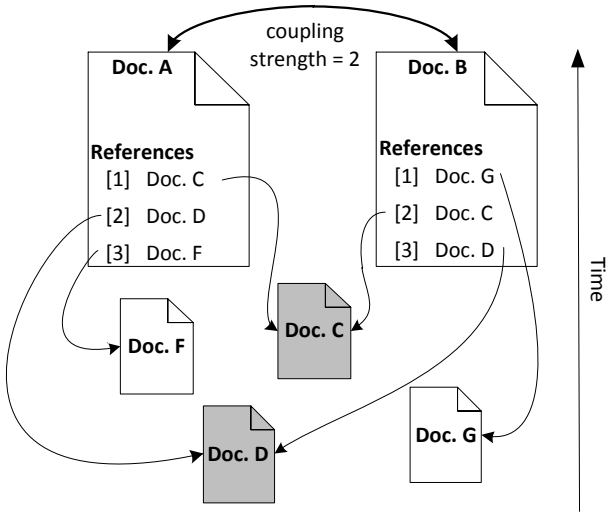



**Figure 2.6: Bibliographic Coupling. Source** [27]
**Doc A and B have a coupling strength of 2 as both cite Doc C and D.**

The degree of similarity is measured by the Bibliographic Coupling strength (BCS). In Figure 2.6 the coupling strength of document A and B is two, since they have two references (document C and D) in common. When documents do not share any references, the BCS is zero.

Bibliographic coupling has been criticised in several ways. Martyn stated that Bibliographic Coupling indicates an relationship between two documents, but not necessary their similarity [37]. Small and Marshakova-Shaikevich criticised as well the static nature of Bibliographic Coupling. The references of a document do not change an therefore, the measure does not evolve over time. Changes in the perception of concepts and ideas are overlooked by this retrospective of Bibliographic Coupling [5], [6]. Consequently, an advancement of this concept has been developed based on these critics.

#### 2.2.2.5 Co-Citation

In 1973 Small and Marshakova-Shaikevic independently developed another citation-based similarity measure named Co-Citation (CoCit) [5], [6]. Instead of focusing on what a document cites, this approach evaluates the citations a document receives. The number of papers citing two documents together equals the co-citation strength, i.e. degree of similarity. For instance,

the document A and B have the co-citation strength of two as both are co-cited by documents C and D (Figure 2.7).

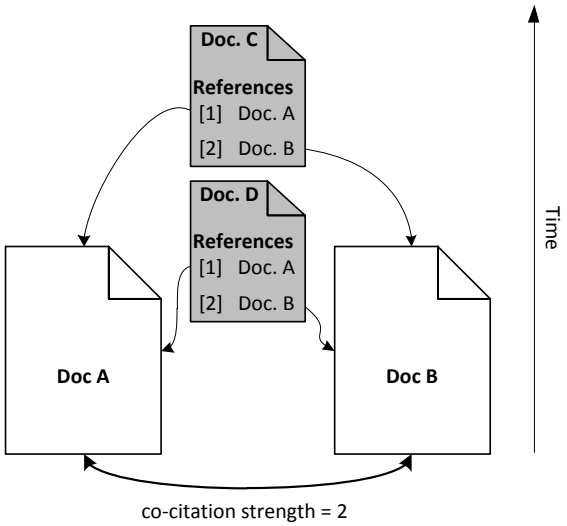



**Figure 2.7: Co-Citation Relationship between Documents. Source** [27]
**Doc A and B have a co-citation strength of 2 as both are co-cited by Doc C and D.**

As a result, Co-Citation has forward-looking perspective compared to Bibliographic coupling. The degree of document similarity measured by Co-Citation can change over time as new documents getting published, resulting in a forward-looking perspective [38].

#### 2.2.2.6 Co-Citation Proximity Analysis

In 2006 Gipp and Beel introduced an advancement of Co-Citation called Co-Citation Proximity Analysis (CPA), which utilises the additional information implied in the citation marker [8].

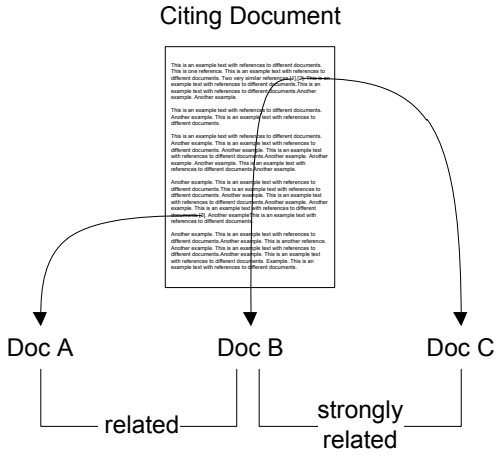



**Figure 2.8: Co-Citation Proximity Analysis. Source** [27]
**Doc B and C are stronger related than Doc B and A as their citation markers are in close proximity.**

They assume that, when citation markers of co-cited documents are in close proximity, the documents are more likely to be similar (Figure 2.8).

Gipp and Beel distinguished five levels of citation proximity based on the citation marker: same sentence, same paragraph, same chapter, same journal and same journal but different year. For instance, documents co-cited in the same sentence are more similar to each other than documents co-cited in the same paragraph. Table 2.1 illustrates the assignment of different values, named as Citation Proximity Index (CPI), to each level.

**Table 2.1: Co-Citation Proximity Index**

| **Occurrence** | **CPI value** |
|---|---|
| Sentence | 1 |
| Paragraph | 1/2 |
| Chapter | 1/4 |
| Same journal / same book | 1/8 |
| Same journal but different edition | 1/16 |

Previous studies showed that CPA provides better performance for scientific documents than Co-Citation [39]. But they also concluded that CPA cannot completely replace Co-Citation and further research is needed to identify the appropriate weighting of the CPI values, because the weighting differs from document topic and document type. Moreover, Beel and Gipp propose additional variations of the CPA algorithm to improve its performance. For example, combining CPA with other similarity measures or evaluating additional information (e.g. citation counts).

## 2.3 Related Work

In this section, we give a short survey of related work. Thereby we focus on research investigating the effect of citation proximity in Co-Citation analysis and work related to Wikipedia as a test collection.

### 2.3.1 CPA

Several publications discuss and investigate the placements of citation within the full-texts of documents as additional information in Co-Citation analysis.

Tran et al. showed the positive effect of sentence-level over paper-level citation proximity when retrieving related articles [39]. They investigated 100,000 articles from PubMed [40] and measured the effect of co-citation proximity by comparing to a text-based document similarity measure. They used the text-based document similarity measure as gold standard. The text similarity was calculated with a VSM- and TF-IDF-based approach, similar to MLT. Their outcome was that sentence-level proximity performs better than article-level proximity co-citation analysis in terms of lexical similarity. Moreover, Tran et al. proposed a generalisation of citation proximity level: Articles cited $n$ sentence apart should also be considered, while the relatedness of cited articles decreases as $n$ increases.

Liu and Chen analysed the effects of Co-Citation Proximity on the quality Co-Citation Analysis in full-text scientific publications [9]. They studied differences in four levels of Co-Citation proximity: article-, section-, paragraph- and sentence-level. They found that sentence-level and article-level Co-Citations are essential for the overall Co-Citation network and that sentence-level proximity is potentially more efficient.

### 2.3.2 Wikipedia

In 2007 Ollivier and Senellart compared Green Measure to several other methods for finding related pages in the case of the English version of Wikipedia [41]. They found out that Green Measure has both the best average results and the best robustness compared to Co-Citation, Cosine with TF-IDF, PageRank of Links and Local PageRank. A user study measured the performance of each method.

Belomi and Bonato investigated Network Analysis techniques on the hyperlinked structure of the whole English Wikipedia [42]. They used HITS and PageRank algorithm to gain understanding of the structure and content of Wikipedia. Both algorithms showed that articles in the categories of geo political spaces, historical events, famous people and abstract nouns or common words dominated Wikipedia in terms of inbound links.

# 3 Methods

In this section, we describe the research methods following the IR system evaluation schema as described by Manning [25]. First, we point out the aims of the evaluation regarding our information needs. Second, we provide essential background information about Wikipedia as our test collection. Third, we define a quasi-gold standard that is based on Wikipedia's "See also" section. Finally, we explain how the performance of the document similarity measures is quantified.

## 3.1 Information Needs

The information needs are in line with the earlier introduced research questions (Section 1):

1. Given that their primary area of application are academic articles, how suitable are the citation-based similarity measures Co-Citation (CoCit) and Co-Citation Proximity Analysis (CPA) to identify related articles in Wikipedia?
2. How does the performance of CoCit and CPA in identifying related Wikipedia articles compare to the performance of a typical text-based similarity measure applied for the same task?
3. Can "See also" sections of Wikipedia articles serve as an approximation of gold standard for topically related articles that allows performing automated large-scale evaluations of document similarity measures?

The concept of citation-based document similarity measures originates from the field of Bibliometric, a research field analysing scientific publications, e.g. journals or books. By changing the application domain to articles of an online encyclopaedia, we analyse a different document type, which contains links instead of citation. It is questionable if the change of the concept, from citation- to link-based, has any effect on the recommendation quality of the two document similarity measures.

Second, we compare the three document similarity measures CoCit, CPA and MLT in terms of their qualitative and quantitative performance to point out a best recommending method for this use case. Furthermore, we analyse the recommendation quality differences. Does the recommendation quality vary for a certain subset of documents? What are the conditions for a good performance? An overview of pros and contras for each measure should be provided.

Third, we test, if "See also" links are useful as a quasi-gold standard, i.e. an approximation of a perfect reference model, for evaluating document similarity measures at a large-scale. Instead of a user study-based evaluation, which is commonly limited to relatively low number of participants and result sets, we use Wikipedia's "See also" links as relevance judgment. "See also" links are available for a large number of articles. Thus, we are capable of evaluating a large set of results for each document similarity measure. Such a large number would be unfeasible to

compare in a traditional user study. However, the question is whether this approach affects evaluation quality? Is the relevance judgment of "See also" similar to judgments of experts in a user study? Are "See also" links missing out relevant results? And, does the large number of "See also" links compensate possible lack of quality?

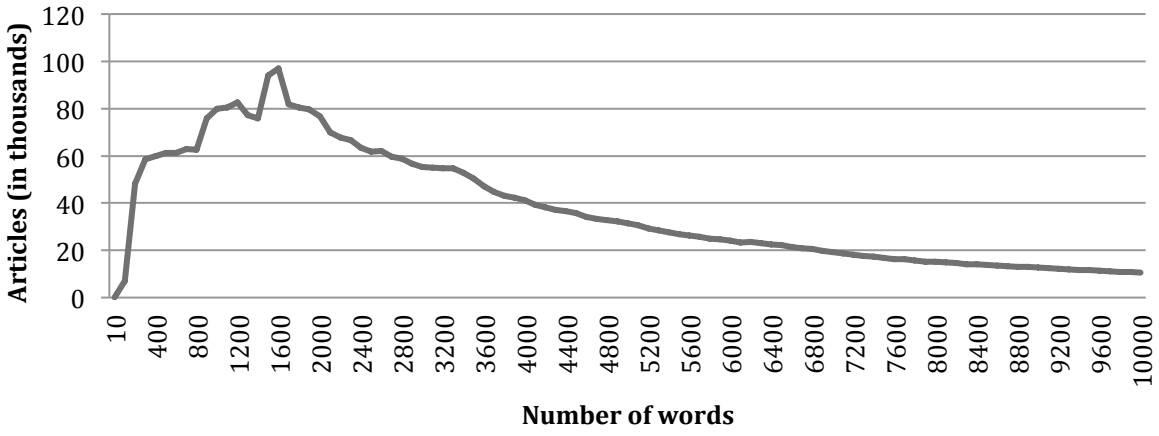
## 3.2 Test Collection

The document similarity measures are applied to the English version of Wikipedia. Wikipedia is a free online encyclopaedia, which is one of the most-frequently used websites [43], [44]. The following paragraphs highlight characteristics of Wikipedia that are crucial to the planned evaluation.

While editorial staff writes traditional encyclopaedias, Wikipedia articles are written, edited and constantly revised by a community of committed volunteers. There is virtually no restriction on the content of Wikipedia articles. Topics covered in Wikipedia range from classical art, history, and science to breaking news and urban legends. Hence, Wikipedia's vocabulary spans a large and diverse set of terms. Hence, lexical document similarity varies. Likewise, the article structure varies, too.

In the following, we present statistics to outline Wikipedia's diversity. The statistical data has been collected with a Flink job that we developed. The source code is available at GitHub[1].

Figure 3.1 and Figure 3.2 illustrate these characteristics by plotting the number of words and the number of headlines per article. The number of words represents the article length. The number of headlines indicates the article structure. Both illustrations show that there is no uniform article length or structure.




[1] https://github.com/TU-Berlin/cpa-demo/blob/master/src/main/java/de/tuberlin/dima/schubotz/cpa/stats/ArticleStats.java

**Figure 3.1: Distribution of words among articles. Avg.: 740.54 words/article. Max.: 75,178 words.**

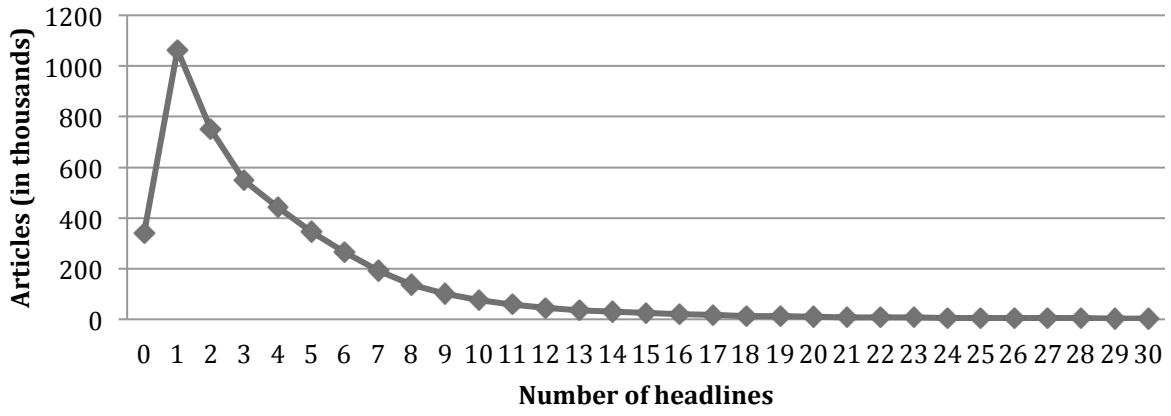



**Figure 3.2: Number of headlines per article. Avg.: 4.27 headlines/article. Max.: 766 headlines.**

Wikipedia represents a directed link graph, in which articles constitute nodes and links between articles constitute edges. Each article can contain hyperlinks to several other Wikipedia articles. Authors are free to create a hyperlink connection between a term that occurs in the article and the corresponding Wikipedia entry. Therefore, some authors use links more often than others and some article receive links more often than others.

Figure 3.3 shows that the majority of Wikipedia articles has no or only a few inbound links. On average, an article has 20.5 inbound links. As reported by Belomi and Bonato, Wikipedia article with a high number of inbound links are mainly about geo topical topics, famous people and abstract nouns or common words [42].

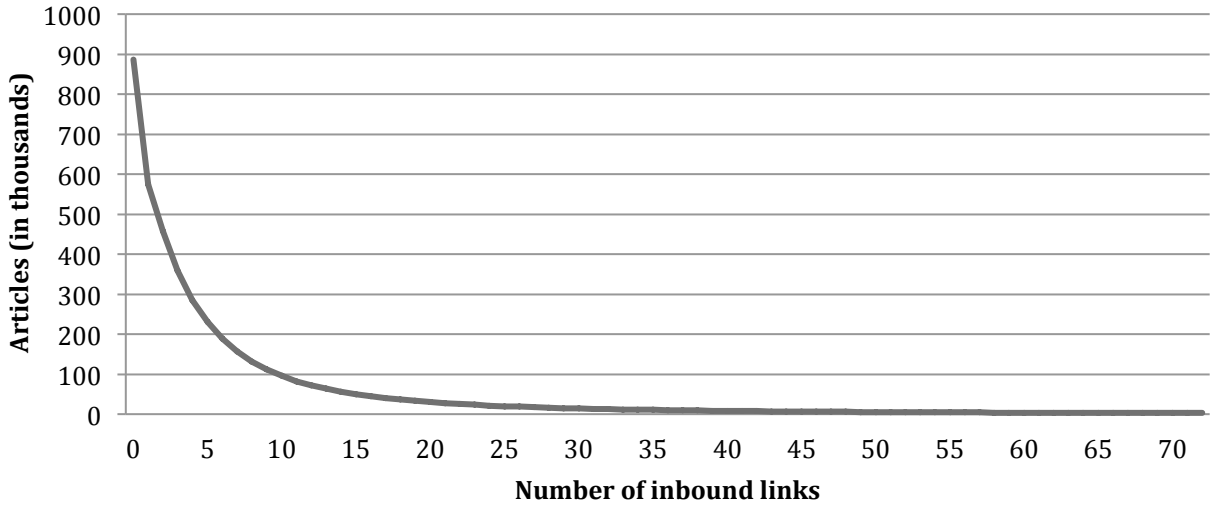



**Figure 3.3: Distribution of inbound links per article. Avg.: 20.5 links. Max.: 392 873 links.**

Figure 3.4 illustrates how often Wikipedia article contain outbound links. On average, an article has 35.9 outbound links. The number differs to the number of inbound links, because an article A can link multiple times to another article B. But we count the links from article A count as one inbound link for article B, but a multiple outbound links for article A.

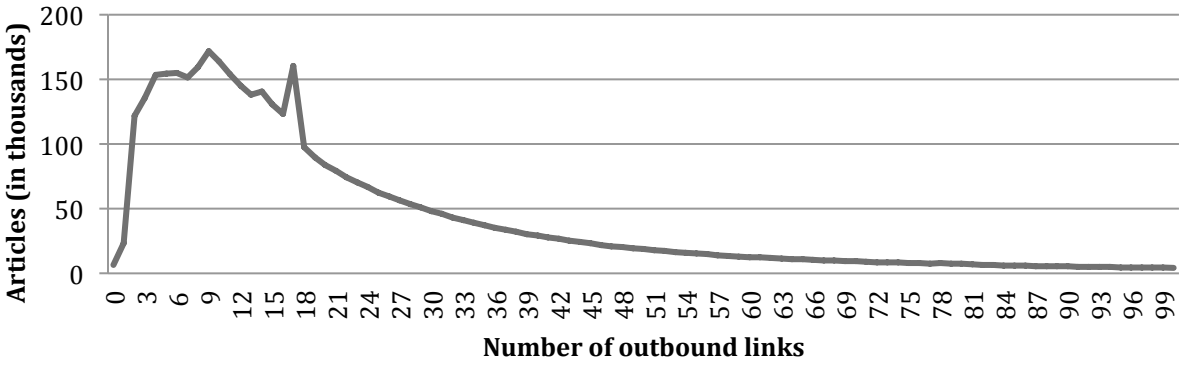



**Figure 3.4: Distribution of outbound links per article. Avg.: 35.9 links. Max.: 9 329 links.**

Figure 3.5 displays the ratio of words per outbound links. It shows that the usage of links varies, but also that the majority of articles has a similar number of around 8-26 words per outbound link. Therefore we see a correlation of number of words and number of outbound links in Figure 3.1 and Figure 3.4.

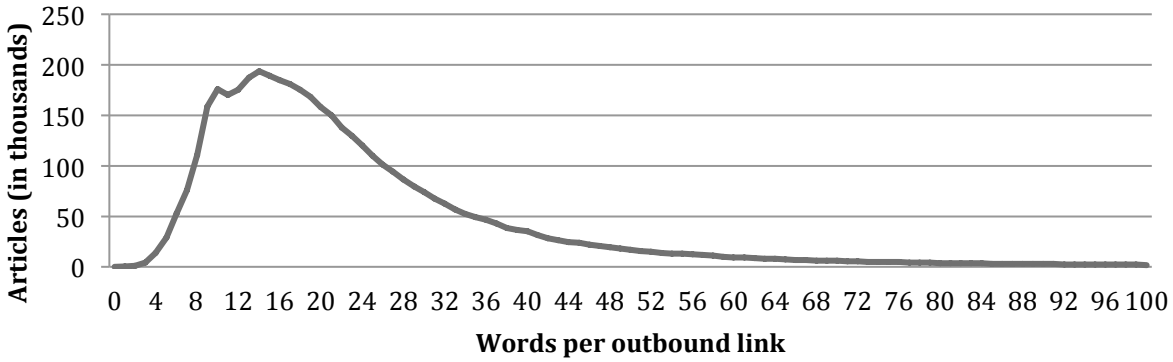



**Figure 3.5: Words per outbound link. Avg.: 28.01 words/links. Max.: 26,420 words/links.**

Wikipedia contains more than 11 million articles in 288 languages. The English version of Wikipedia, which we analyse in this section, contains about 4.6 million entries and accounts for the majority of articles. A multilingual evaluation would require an additional effort, since articles in different languages are not connected by in-text hyperlinks and therefore would have to be fetched from Wikidata[2].

[2] http://en.wikipedia.org/wiki/Help:Interlanguage_links

Wikipedia offers free dumps, i.e. copies, of all available content in different formats. We use a XML dump, because XML facilitates machine processing. Each article is formatted in Wiki mark-up, a lightweight mark-up language that is a simplified intermediate to HTML [45].

The collection used for this analysis consists of 36 million pages in Wiki mark-up. A page is any kind of Wikipedia content (namespaces), e.g. images, categories, user profiles or articles. The dump, which we downloaded[3], has a size of approximately 99 GB and was created in September 2014.

In summary, the English Wikipedia is test collection with diverse vocabulary and article structure, organised as a large directed link graph. These properties are crucial for our evaluation. They increase the amount of data the document similarity measures can take into account and allow automated processing. Therefore, we choose Wikipedia article as subject of the following evaluation. Hence, we use terms "documents" or "retrieved documents" as expressions for Wikipedia articles that are retrieved by a document similarity measure.

## 3.3 Gold Standard

The "See also" section is an element of Wikipedia articles that consists of a list of internal links, which point to topically related articles. For the reasons explained hereafter, we expect that "See also" links serve well as a quasi-gold standard for our evaluation. In advance we clarify the terminology of gold standard and quasi-gold standard.

The common approach of evaluating IR systems is to compare the retrieved documents to a reference model to classify a document as either relevant or irrelevant. A gold standard or ground truth is the perfect reference model that provides the best possible responses to any tested query. True positives are all retrieved documents that are part of the gold standard and therefore relevant. All other retrieved documents are false positives, i.e. irrelevant. For many applications a gold standard remains a theoretical concept, which is impossible to achieve in real world. Even a traditional user study, in which domain experts are asked to identify relevant documents, cannot completely eliminate misjudgements especially false negative errors as experts may miss relevant documents.

Therefore, we introduce the term quasi-gold standard as approximation of a perfect reference model. A quasi-gold standard provides relevance judgements of comparable quality as the relevance judgments of domain experts. Retrieved documents that are part of the quasi-gold standard are true positive. However, the quasi-gold standard cannot distinguish whether all other retrieved documents are false positive or false negative. In the context of Wikipedia, a quasi-gold standard is capable of determining if a retrieved document is a relevant

[3] https://dumps.wikimedia.org/enwiki/latest/

recommendation for a topically related Wikipedia article, but not, if a recommendation is irrelevant.

Aside from their main content, Wikipedia articles contain pointers to additional information in form of references or external links, but also a so-called “See also” section. The Wikipedia guideline states that this section should include a list of internal links to topically related Wikipedia articles. The purpose of "See also" links is to enable readers to explore tangentially related topics [46]. The links can assist readers in finding related articles. For our evaluation, we assume that ”See also” links correlate with the expected results of a literature recommender system. Referring to Manning [25], the “See also” links are a user-generated judgement of relevance, i.e. they are a quasi-gold standard.

When using “See also” as a quasi-gold standard, we can classify document relevance as follows:

The documents that the investigated similarity measures retrieved and that exist as “See also” links are judged as relevant. Retrieved documents for which no “See also” link exists are classified as irrelevant. At this point, we see a problem:

We expect the “See also” links to be an incomplete gold standard, since Wikipedia’s volunteers, whose main objectives might be creating textual content rather than providing literature recommendations, create this content. Even if a retrieved document cannot be found within the “See also” links, it still can be topically related, i.e. relevant. Therefore, we can decide if a result is relevant, but not if it is irrelevant. A true binary classification is not possible. Hence, we expect a precise true positive classification for documents that exist as “See also” links, while many results could be classified as false negative without document similarity measure failures, when the retrieved document are simply missed by “See also” links. Consequently, the performance measure should consider these properties of a quasi-gold standard by not excessively penalising a similarity measure for documents that cannot be classified as relevant.

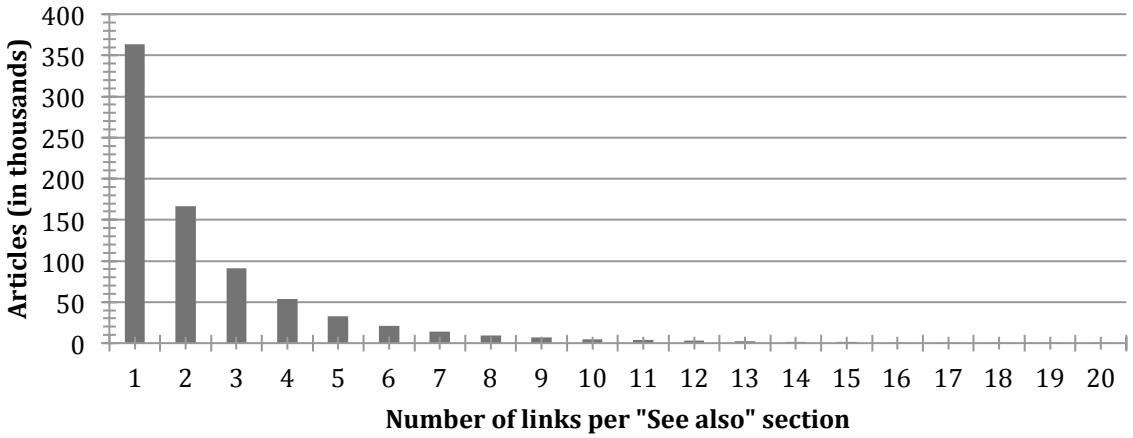



**Figure 3.6: Number of links per "See also" section. Avg.: 2.6 links. Total: 2,022,601 links.**

We are able to extract the “See also” section and its links using an automated process, since Wikipedia articles are structured in Wiki mark-up language. Wikipedia contains 777.047 articles with a “See also” section. More than two million internal links to Wikipedia can be found within the “See also” sections. On average a “See also” section consists of 2.6 links (Figure 3.6).

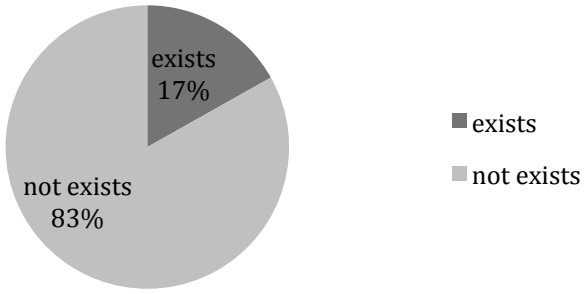



**Figure 3.7: Percentage of article with "See also" section. Exists: 777,047; Not exists: 3,835,682**

We expected that only a small fraction of all Wikipedia articles contains a “See also” section, since this section is optional for Wikipedia authors [46]. Yet, an evaluation based on this quasi-gold standard allows us to take the results of 777,047 queries into account, many more than a user study would usually do. User studies of this research field rarely test hundreds or more results [47].

## 3.4 Performance Measures

Answering the research questions requires the ability to measure the performance of the evaluated document similarity measures. We measure the performance by calculating the Mean Average Precision (MAP) for each document similarity measure. The following section introduces MAP and explains the suitability of the performance measure for our research purposes.

### 3.4.1 Precision & Recall

Evaluating results has always been a key task in IR. Is an IR system effective? Is another technique superior? Over the past decades, two properties have been established to answer those questions:

> **Precision**: The number of retrieved documents that are relevant in relation to the total number of retrieved documents.
>
> **Recall**: The number of retrieved documents that are relevant in relation to the total number of relevant documents.

A high recall can be achieved if an IR system retrieves all documents regardless of their relevance. However, this strategy will lower precision, since irrelevant documents are retrieved as well. Likewise, an IR system that retrieves only one relevant document when multiple relevant documents exist achieves a high precision, but a low recall. Commonly, precision and recall behave contradictory. Whether a high precision or high recall is preferable depends on the domain and the use case. A typical user of an Internet search engine might be interested in browsing exclusively through the first ten results [48], thus, prefers a high precision over recall. On the contrary, a researcher doing a literature review may be willing to screen significantly more than ten literature recommendations to find a relevant paper. Hence a researcher may favour high recall over precision.

Both precision and recall rely on the ability to judge a document's relevance. As we introduce in Chapter 1 relevance is the ability to satisfy a user's information need, which can differ from user to user and query to query. In most real world use cases a strict division in relevant or irrelevant documents is difficult. Some documents might be highly relevant and others marginally. However, for simplicity and comparability, we use a binary classification of relevance.

The standard approach to judge relevance is a user study, in which each participant decides whether a retrieved document is relevant or not. We, however, did not perform a user study, but used "See also" links as quasi-gold standard. This decision increases the risk to miss out relevant documents. As we state in Section 3.3, the performance measure therefore should not penalise an IR system for retrieved documents that cannot be classified as relevant.

Furthermore, more than one document can be relevant and the number of relevant documents, i.e. "See also" links, varies from query to query. Therefore, the performance should be able to handle multiple relevant results.

Being intuitive and easily discriminable are additional requirements for the evaluation measure. Ideally, a performance measure should be a normalised scalar score. A perfect IR system, which returns all relevant and no irrelevant documents, should receive a score of 1. In the worst of retrieving only irrelevant documents, a score of 0 should be assigned. Moreover, the score should be comparable across similarity measures to determine the best performing measure.

### 3.4.2 Top-K Results

Precision and recall are set-based evaluation measures. They are calculated using unranked sets of documents. However, the document sets retrieved by similarity measures in this evaluation are ranked. Therefore, we decided to consider rank information as a performance criterion.

The rank-based performance measures evaluate a limited subset of retrieved document, which are called the top $k$ retrieved documents, where $k$ represents the number of documents in the subset. The value of $k$ differs from use case to use case similar to the ratio of precision and recall. For Wikipedia the value $k$ also depends on the user's information need. The Wikipedia manual does not point out an exact value, moreover, it says that the number of recommended articles "should be limited to a reasonable number" [46].

For the following evaluations we use $k = 10$. While scientific literature recommender systems may use a higher $k$ for the previously explained reasons, many common Internet services, e.g. Internet search engines, retrieve the top ten documents on the first result page. Also, the average number of 2.6 links per "See also" section (Figure 3.6) is too low to justify a higher $k$. A lower $k$ seems as well not suitable as several articles are probably topically related with each other.

### 3.4.3 Mean Average Precision

We used Mean Average Precision (MAP) as rank-based performance measure, because it is widely used among the IR community [25] and meets our requirements.

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{|R_q|}\sum_{j=1}^{|R_q|}\mathrm{Precision}(R_{qj}) \qquad (3.1)$$

MAP is computed as shown in Equation 3.1. As the name indicates, it is the mean of the average precision scores for each query of a set of queries, where $Q$ is a set of queries with cardinality $|Q|$ and with at least one relevant result. $R_q$ are the relevant results for query $q$ in

ranked order with cardinality $|R_q|$. $\text{Precision}(R_{qj})$ is the precision value of relevant results $R_{qj}$ for the first $j$ relevant results for query $q$ as defined in the beginning of this section.

Figure 3.8 illustrates the computation of MAP score for an example query set $Q \in \{q_1, q_2\}$, where relevant results are marked with "X" in the corresponding column.

| $k$ | **1** | **2** | **3** | **4** | **Avg.** |
|---|---|---|---|---|---|
| $q_1$ | | X | X | | |
| $j$ | | 1 | 2 | | |
| **Precision**($R_{q_1 j}$) | | 0.50 | 0.67 | | 0.58 |
| $q_2$ | X | | X | X | |
| $j$ | 1 | | 2 | 3 | |
| **Precision**($R_{q_2 j}$) | 1.0 | | 0.67 | 0.25 | 0.64 |
| $\text{MAP}(Q) = \frac{0.58 + 0.64}{2} = 0.61$ | | | | | |

**Figure 3.8: MAP example for two queries $q_1$ and $q_2$.**

In our context $Q$ is equivalent to the set of Wikipedia articles with a "See also" section and $R_q$ are the relevant documents retrieved by CoCit, CPA or MLT. By calculating MAP score for each similarity measure we easily can determine whether one approach is superior or not.

The resulting MAP score is normalised scalar. Thereby, MAP ensures comparability across different IR systems. The rank of a relevant result has an influence on the average precision score, but also false negative errors are not penalised heavily compared to other measures, e.g. Geometric Mean Average Precision [49], [50]. For this reasons, we take MAP for an appropriate performance measures for evaluating document similarity measures.

In preparation of this research we also made test runs with the performance measure Mean Reciprocal Rank. In comparison to MAP the different performance measure did not reveal any different evaluation results.

# 4 Implementation

In this section, we describe how we implemented the document similarity measures and their evaluation. We start with introducing the technologies our implementation is based on, why we used them and describe the setup we run our experiments on. To make it reproducible, we continue by explaining the similarity measures from the technical perspective.

## 4.1 Technology

We choose a large test collection for the reasons we explain in Section 3.2. This decision caused the challenges regarding the methods of data processing. Therefore, we make use of the so-called "Big data" technologies. In the following we briefly outline characteristics of these technologies:

Academia, industry and media use the term "Big Data" is inconsistently. There is no single unified definition [51]. We go along with a definition of Gartner that is known as the "three Vs":

*"Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making."* [52]

Technologies for storing and processing "Big Data" therefore have requirements that differ from traditional Relational Database Management Systems (RDBMS):

**High volume**: Systems need to be capable to store and process several terabytes or even petabytes of data. Thus, the systems are forced to use distributed storage and parallel processing in computer clusters as data exceeds the limits on a single machine. This requirement is crucial for our experiment as the intermediate results of the CPA and CoCit computation are several terabytes in size.

**High variety**: Systems need to handle various data types, i.e. structured, unstructured data as well as everything in between. "Big Data" system do not rely on a strict consistently column layout, whereas RDBMS typically work with fixed database schema. For instance, in this experiment we process a Wikipedia XML dump and evaluate a citation-graph.

**High velocity:** Systems need to process data at high speed to allow analysis of data streams at near real-time. High velocity is often achieved to the detriment of consistency. In contrast to traditional ACID guarantees of RDBMS, NoSQL data models for "Big Data" provide so-called "eventual consistency" to enable massively concurrent insert and read operations [53]. Elasticsearch, a technology used in this experiment, uses a NoSQL data model.

In the following, we introduce the programming model of MapReduce and the technologies Apache Hadoop, Apache Flink and Elasticsearch (Lucene) had met these requirements.

### 4.1.1 MapReduce Programming Model

A well-established example for "Big Data" technologies is the MapReduce programming model. First introduced by Dean and Ghemawat at Google [54], MapReduce is a pattern for processing and generating large datasets using many computers. Programs that follow this paradigm consist of two second order functions, a map function (Mapper) that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function (Reducer) that merges all intermediate values associated with the same intermediate key. Both citation-based similarity measures we test in this thesis are expressible using this functional style model.

MapReduce is capable to meet the outlined "Big Data" requirements by providing a framework for automated parallelisation and execution of its programs on computer cluster. Therefore, a MapReduce program, a so-called job, splits input data into chunks, which can be independently processed by the parallelised map function. In the next step, the data is redistributed among the nodes depended on the keys produced by the Mapper (shuffling). This way, all data belonging to one pair is located on the same machine. Then each computer (node) performs the reduce task on each key and stores the result in a file system.

Aside from distributed processing on many computers, this parallel approach also provides redundancy and fault tolerance. If one map or reduce task fails on one node, the work can be re-executed by another node, assuming the input data is still available. This allows easy recovery in case computers fail.

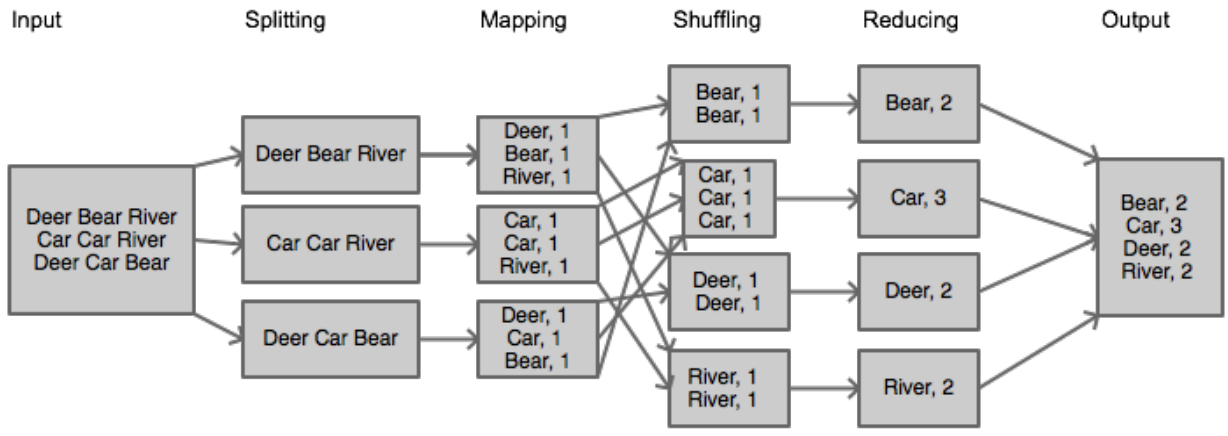



**Figure 4.1: MapReduce word count example.**

Figure 4.1 illustrates a simple word count program implemented in the MapReduce programming model. It counts the words of the input text. The input is a multi-line text file, which is split into single lines in the first step. Then, the Map operation extracts all words from the line and transforms each word to a two-tuple consisting of the word and an integer value set to one, which represents the number of word occurrences. The tuples are then sorted and grouped by the word during the Shuffling phase. Afterwards, the Reduce operation sums up the

word count for each tuple and produces the final output. The example shows that the Map and Reduce operations can be performed independently on different computers.

### 4.1.2 Apache Hadoop

The MapReduce programming model exists in several implementations. A widely used framework implementing this model is Hadoop [55], [56]. Written in Java, this framework published by the Apache foundation is available through an Apache open source license and consists of several components for distributed storage and distributed processing of Big Data. Core components are the Hadoop Distributed File System (HDFS) and Hadoop MapReduce.

**HDFS:** The storage system of Hadoop has a master/slave architecture. A single master server (NameNode) manages the file system namespace and the access to files by clients. The files themselves are stored on a number of slave servers (DataNodes). Those files are split into chunks, usually 64 or 128 MB blocks of data, and distributed on different machines. Typically each block is replicated three times within the cluster to provide fault tolerance. The NameNode determines which replication is stored on which DataNode. It also periodically checks the file system on errors. In case of failure the NameNode restores those blocks automatically by using their replications. Also, this block layout facilitates sequential I/O and therefore increases performance as sequential I/O decreases the time spent waiting for disk seeks and rotational latency.

File system namespace operations that affect the mapping of blocks are performed by the NameNode, whereas read or write requests by clients are handled by the DataNodes. The system is designed that the master server is not involved in any I/O operations.

This setup allows HDFS to provide the ability to access large amounts of data with high I/O throughput. Therefore, we use HDFS as storage engine for the computation and evaluation of the document similarity measures.

### 4.1.3 Apache Flink

Major parts of our experiment, like evaluation and citation-based similarity measures, are implemented with the Apache Flink framework. In the following, we give an overview about Apache Flink as technology and explain why we used it.

Formerly known as Stratosphere and started as research project of TU Berlin, Apache Flink is a part of the Apache Software Foundation [57], [58]. Apache Flink is, like Hadoop, an open source Java framework for processing "Big Data". It focuses on batch and streaming data processing. It does not contain its own storage engine, but it can be built upon a distributed file systems like HDFS. Figure 4.2 illustrates the layer architecture of Apache Flink.

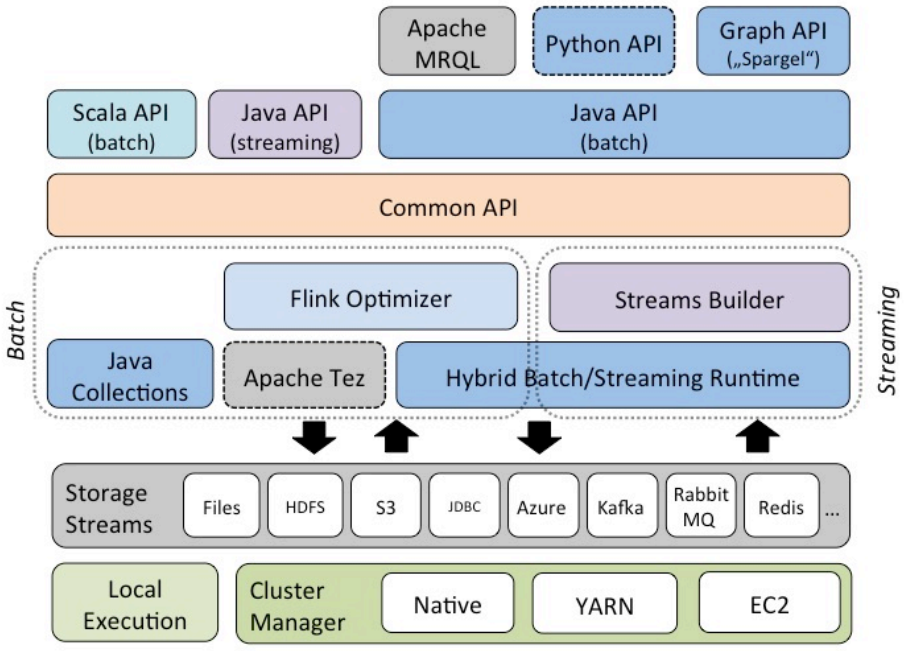



**Figure 4.2: Apache Flink layer overview. Source** [59]

Hadoop's MapReduce programming model enables processing of large datasets and is suitable for many real world use cases. Nevertheless, writing efficient application in MapReduce requires strong programming skills and in-depth knowledge of its architecture. Apache Flink has been developed, in order to allow non-experts to use such system, save development time and make application code easier to understand and maintain.

The goal of this research is the evaluation of document similarity measure. Even if the implementation is essential, we do not focus on developing an optimised implementation of each document similarity measure. Therefore, Apache Flink offers us the opportunity to benefit from its "Big Data" technologies without requiring huge effort in development. Also, our evaluation implementation uses functionalities, like Group and Join operators, which are not available in

simple MapReduce (Section 4.2.6). We used the Flink Java API to implement CoCit, CPA and the evaluation process. The programs were transformed through the Commons API and internally optimised by the Flink Optimizer.

### 4.1.4 Lucene & Elasticsearch

Aside from CoCit and CPA, we also evaluate the text-based MLT. As we explain in Section 2.2.1, the text-based similarity measure applies VSM and TF-IDF to determine document similarity. In our experiment, we use Elasticsearch, which is built on top of Apache Lucene, as an implementation of a text-based concept. In the following, we start with introducing Apache Lucene and its MoreLikeThis functionality and continue with Elasticsearch.

#### 4.1.4.1 Lucene

Apache Lucene is a free, open source project developed by the Apache Software Foundation and implemented in Java [60], [61]. Lucene is a document indexing and search technology that can be used for a variety of information retrieval task. Search functions of a various number of websites, like Wikipedia and Twitter, rely on Lucene. The framework is divided in two main components, namely Indexer and Searcher, which are used for indexing and searching documents stored in a file system.

The indexing process converts input data into searchable Lucene documents and creates an index. A Lucene document consists of fields, where each field has a name and an unstructured textual content. In our experiment we extract title and text from each Wikipedia article and generate a Lucene document. Then the unstructured textual content is tokenized into single words, which are analysed by performing operations like lowercase transformation, stop words removal and stemming. These operations decrease the number of term rows in the term-document matrix (Figure 2.1), but they are mainly language dependent, since stemming and stop words differ from language to language. Lucene provides this functionality for several languages, e.g. English and German.

After the text analysis Lucene stores the documents in an inverted index. This type of data structure is widely used by many search engines. Instead of storing full documents and searching through the whole content, with an inverted index only the extracted tokens are used as lookup keys and mapped to document they belong to. Resulting in decrease of disk space usage and increase of search speed. Looking up the indexed keys is sufficient, when searching for documents, which contain a specified search term.

Lucene's Searcher component is capable of performing all search queries to the Lucene index. First, the search query is parsed. Next, the query is processed depending on the type of query. Finally, the results are returned as ranked results to the user.

#### 4.1.4.2 TermQuery

The results of queries based on terms (TermQuery) are first collected with a Boolean retrieval model that retrieves a set of documents containing the query terms. This step can be performed quickly as it does not involve extensive computation. Next, the subset is sorted with a scoring formula found on VSM and TF-IDF (Section 2.2.1).

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t=1}^{n} (q_t \cdot \text{tf}(t,d) \cdot \text{idf}(t)^2 \cdot \text{termBoost}_t) \qquad (4.1)$$

Equation 4.1 is Lucene's scoring function [62] for a query consisting of one or more terms and is represented by the query vector $\boldsymbol{q}$ of length n, where $\boldsymbol{q_i} = \begin{cases} 1 & \text{term } i \text{ occurs in query} \\ 0 & \text{otherwise} \end{cases}$, and a document vector $\boldsymbol{d}$ of the same length that represents the row in the term-document matrix (Figure 2.1) here:

- $\text{coord}(q,d)$ is number of occurrences of query terms in document $d$, implemented as $\text{coord}(q,d) = q \cdot d$
- $\text{queryNorm}(q)$ is a normalising factor used to make scores between queries comparable, implemented as $\text{queryNorm}(q) = \frac{1}{\sqrt{\text{queryBoost}_q^2 \cdot \sum_{t=1}^{n} (q_i \cdot \text{idf}(t) \cdot \text{termBoost}_t{}^2)}}$
- $\text{tf}(t,d)$ is the Term Frequency of $t$ in the document $d$, implemented as $\text{tf}(t,d) = \sqrt{d_t}$
- $\text{idf}(t)$ stands for the Inverse Document Frequency and is calculated by the following logarithmic formula: $\text{idf}(t) = 1 + \log\left(\frac{\text{m}}{\sum_{k=1}^{m} w_{t,k}}\right)$
- $\text{termBoost}_\text{t}$ is a score factor used to give term $t$, set to 1 by default, where $\text{termBoost}$ is vector of length n. This factor is also used in the MoreLikeThis query (Section 4.1.4.3).
- $\text{queryBoost}_\text{q}$ is score factor used to give query $q$ preference, when combing several queries, set to 1 by default.

We leave out a field normalisation factor, as neither the experiment nor the following example uses Lucene's field queries. Also, the notation of function $\text{score}(q,d)$ in Equation 4.1 is simplification, since the computation depends on the elements $w_{i,j}$ from the term-document matrix to calculate the Inverse Document Frequency.

When applying the scoring function on a result set, not all factors are calculated freshly. All factors depending on the term-document matrix are computed in advance, i.e. when indexing a new document. This decrease the time needed by Lucene to answer a query. However, it requires as well more indexing effort and therefore increases the time spent for adding new documents to the index.

For a better understanding, we illustrate in Figure 4.3 an example of Lucene's scoring function that is based on the data from the term-document matrix in Figure 2.2.

**Document Vector:**

$d_2 = (5 \quad 4 \quad 0)$

**Query Vector:**

$q = (1 \quad 1 \quad 0)$

**Term Frequency:**

$$\text{tf}(1, \text{d}_2) = \sqrt{d_{2_1}} = \sqrt{5} \cong 2.24$$
$$\text{tf}(2, d_2) = \sqrt{d_{2_2}} = \sqrt{4} = 2$$

**Inverse Document Frequency:**

$$\text{idf}(1) = 1 + \log\left(\frac{\text{m}}{\sum_{k=1}^{m} w_{1,k}}\right) = 1 + \log\left(\frac{4}{9}\right) \cong 0.19$$
$$\text{idf}(2) = 1 + \log\left(\frac{\text{m}}{\sum_{k=1}^{m} w_{2,k}}\right) = 1 + \log\left(\frac{4}{15}\right) \cong -0.32$$

**Query Terms in Document:**

$$\text{coord}(q, d_2) = q \cdot d_2 = 9$$

**Query Normalisation:**

$$\begin{aligned}\text{queryNorm}(q) &= \frac{1}{\sqrt{\text{queryBoost}_q^2((\text{idf}(1) \cdot \text{termBoost}_1)^2 + (\text{idf}(2) \cdot \text{termBoost}_2)^2)}} \\ &= \frac{1}{\sqrt{\left(1 + \log\left(\frac{4}{9}\right)\right)^2 + \left(1 + \log\left(\frac{4}{15}\right)\right)^2}} \cong 2.68\end{aligned}$$

**Lucene Score:**

$$\begin{aligned}\text{score}(q, d_2) &= \text{coord}(q, d_2) \cdot \text{queryNorm}(q) \cdot \sum_{t=1}^{3} (q_i \cdot \text{tf}(t, d_2) \cdot \text{idf}(t)^2 \cdot \text{termBoost}_t) \\ &= 9 \cdot \frac{1}{\sqrt{\left(1 + \log\left(\frac{4}{9}\right)\right)^2 + \left(1 + \log\left(\frac{4}{15}\right)\right)^2}} \cdot \left(\sqrt{5} \cdot \left(1 + \log\left(\frac{4}{9}\right)\right)^2 + \sqrt{4} \cdot \left(1 + \log\left(\frac{4}{15}\right)\right)^2\right) \\ &\cong 6.92\end{aligned}$$

**Figure 4.3: Example of Lucene's scoring function for a query $q$ and document $d_2$.**

Query vector $\mathbf{q}$ represents a search for two terms "car" and "truck", while document represented by document vector $\mathbf{d_2}$ contains five times the term "car" and four times the term "truck". The example below does not contain any preferences for terms or queries and therefore all boost factors are set to 1. The resulting score is approximately 6.92.

Moreover, the example of Lucene's scoring functions shows that, even though Lucene relies on VSM and TF-IDF, the actual implementation differs from their original concepts (Section 2.2.1). Lucene's Term Frequency is denoted as square root, while Inverse Document Frequency is implemented as logarithmic formula. In this way, Lucene uses a more complex but highly optimised approach.

#### 4.1.4.3 MoreLikeThis

MLT has the characteristics of a document similarity measure as its purpose is to retrieve documents similar to input document. Thereby, Lucene does not operate exactly like the VSM concept, which we outline in Section 2.2.1. A computation of Cosine similarity of all document pairs would not scale. Instead, a MLT query is efficiently executed as a set of TermQueries to retrieve similar documents.

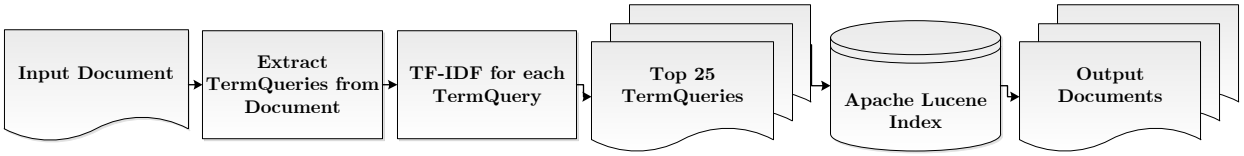



**Figure 4.4: Lucene's MoreLikeThis - from input document to set of TermQueries.**

First, the input document of a MLT query is processed like in the indexing process. Next, for each term created a TF-IDF score is calculated. All terms are then sorted descending by TF-IDF and limited a configurable number of terms. In this experiment, we select the top 25 terms. Afterwards, a query consisting of the selected terms is created. The weighting of each TermQuery corresponds with its TF-IDF score, i.e. preference of a term with the TermBoost factor within the scoring function (Equation 4.1). Boolean operators link all TermQueries so at least one or more queries need to match. Finally, documents including their score are returned in a ranked order.

#### 4.1.4.4 Elasticsearch

Elasticsearch, developed by Elastic a US-based company, is a full-text search engine build on top of Apache Lucene [63], [64]. It is written in Java and available via Apache Open Source License. Elasticsearch comes with all of Lucene's search functionalities including MLT. It is designed to be distributive and scalable on server cluster.

We choose Elasticsearch as platform for the text-based similarity measure, because it is easy to install, does not require sophisticated configurations and is capable of indexing the Wikipedia text corpus.

## 4.2 Experimental Setup

In the following, we introduce our experiment applications of CoCit, CPA, quasi-gold standard computation and evaluation as well as the Elasticsearch-based MLT implementation.

### 4.2.1 Hardware & Software

The experiment was performed on a cluster of 10 IBM Power 730 (8231-E2B) servers. Each machine had 2x3.7 GHz POWER7 processors with 6 cores (12 cores in total), 2 x 73.4 GB 15K RPM SAS SFF Disk Drive, 4 x 600 GB 10K RPM SAS SFF Disk Drive and 64 GB of RAM.

We used Apache Flink v0.8 and Hadoop v2.0. The text-based similarity measure was evaluated with Elasticsearch v1.4.2. All versions were the latest stable releases at the time of writing. We used the software's default settings.

### 4.2.2 Components

In advance of the evaluation we need to generate the results of each similarity measure and extract the "See also" links. Three separated applications compute the intermediate results (Figure 4.5).

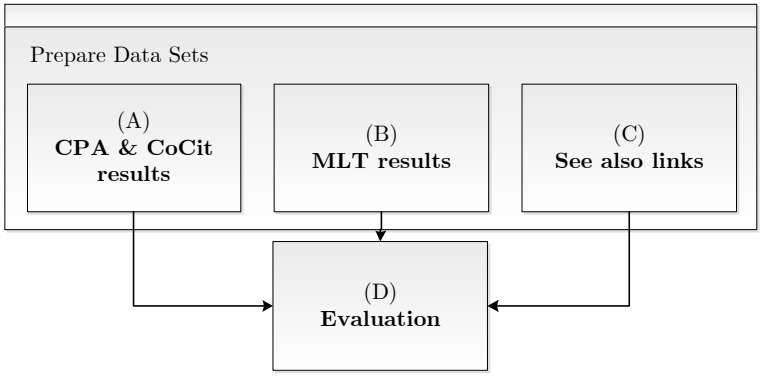



**Figure 4.5: Application components.**

CPA and CoCit are implemented within one application (A), whereas MLT is performed by the second application (B) and the third generates the gold standard dataset (C). Intermediate datasets are stored in HDFS and later processed by the evaluation application (D). In the following we explain the individual implementations as Apache Flink jobs, their challenges and how we solved them. The source code is available on GitHub[4,5].

[4] CPA, CoCit, SeeAlso & Evaluation: https://github.com/TU-Berlin/cpa-demo
[5] MoreLikeThis: https://github.com/mschwarzer/Wikipedia2Lucene

### 4.2.3 CPA & CoCit Implementation

Both CoCit and CPA use a citation graph to compute document similarity. The fact that both measures rely on the same data, allows us to combine them in one application, in which it is necessary to extract all citations. However, our test collection Wikipedia does not contain citations, moreover, we determine similarity based on hyperlinks (Section 3.2). Extracting links from all documents is thereby the first of four processing step:

1. Reading Wikipedia documents
2. Extracting links from each document
3. Building LinkTuple from links
4. Summing up LinkTuple values

The separated stages of the program for generating CPA and CoCit results are illustrated in Figure 4.6. The LinkTuple data structure that stores co-citation information is shown in Figure 4.7.

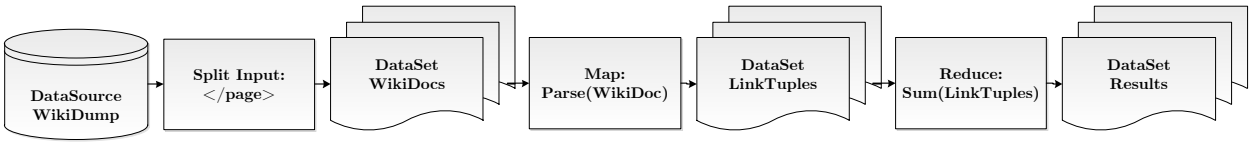



**Figure 4.6: CPA and CoCit program plan.**

#### 4.2.3.1 Parsing Wikipedia Articles

At first the Wikipedia XML dump, which is located in HDFS, is defined as data source of this program. We read from this input by using a *DelimitedInputFormat*[6] that extracts an XML element for each Wikipedia page. Since the input is stored in HDFS the reading process can be performed in a parallel manner.

Then, the *Map* function parses the XML content to extract information as title, text and namespace of each page. If the namespace determines that the page is not an article, rather a user profile or category page, the page is discarded. Redirect pages, which also belong to the article namespace, but do not contain any content except one hyperlink to the redirected article, are discarded, too.

Elsewise, the document text is processed. The "See also" section is removed by searching for sections that headlines contain the string "See also". The search operation is implemented as a case-insensitive regular expression. Even if the Wikipedia guideline proposes the headline "See

[6] http://ci.apache.org/projects/flink/flink-docs-release-0.7/api/java/org/apache/flink/api/java/record/io/DelimitedInputFormat.html

also", some authors might name the section in another way. In case of a different naming, we are not able to detect such section as "See also" section. We expect the percentage of missing detection to be very small, since we never noticed a different naming of the "See also" section during the manual evaluation.

Next, hyperlinks to other Wikipedia articles are extracted out of the leftover text. Afterwards, all links are then combined pairwise with each other. Thereby, we retrieve all co-citations. The resulting tuples of link pairs are stored in a *LinkTuple* object as illustrated in Figure 4.7. The object consists of the first link target (Page A) and the target of the second link (Page B). To eliminate duplicate link pairs, like A-B and B-A, the order is not defined by the occurrence in the text. Instead, link pairs are ordered alphabetically. The *LinkTuple* also includes the number of occurrence of this link pair, e.g. the Co-Citation strength.

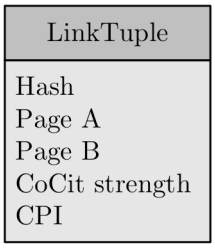



**Figure 4.7: LinkTuple**

#### 4.2.3.2 Document-based CPI Computation

Co-Citation Proximity Index (CPI) was already introduced as metric of CPA to quantify co-citation proximity. Based on their empirical evaluation on scientific publications, Beel and Gipp suggested static CPI values depending on their textual occurrence (Table 2.1). But, as they also say, different document types probably require different weighting of co-citation proximity [8].

As we discuss in Section 3.2, our test collection differs in several ways from scientific publications. Wikipedia articles are not grouped in journals or books, so we cannot determine proximity by this level of occurrence. Moreover, the length and structure of Wikipedia documents varies as Figure 3.1 and Figure 3.2 show. Evaluation by paragraph or chapter therefore is not suitable for our use case.

Thus, we decided to introduce a new dynamic model of CPI that can be adjusted depending on the requirements of document type. Thereby, we consider the proposal of Tran et al. to generalise the citation proximity level [39]. Analogue to the Term-Document Matrix, we define a $m \times m$-matrix with element $v_{i,j}$ that stores the link position for all $m$ documents. Specifically the column for document $j$, $v_{*,j}$ holds the positions for links to other documents in words counted from the beginning oft the document[7].

For example if document $j$ links to document $i$ at position $k$, $v_{i,j} = k$. We define the $j$-link-distance $\Delta_j(a,b) = \begin{cases} |v_{a,j} - v_{b,j}| & v_{a,j} > 0 \ \wedge v_{b,j} > 0 \\ 0 & \text{otherwise} \end{cases}$.

[7] Without loss of generality, we assume for the following description that each document links at most once to any other document.

The CPI for a single co-citation in document $j$ is defined as the link distance damped by an exponential parameter $a$, where $\alpha$ defines how the link distance is weighted.

$$\mathrm{CPI}_j(a,b) = \Delta_j(a,b)^{-\alpha} \quad (4.2)$$

The exact value of $a$ needs to be computed dependent on the document type, e.g. the model needs to be optimised for performance, whereby $a$ is not allowed to be negative, because a negative value of $a$ less would result in a weighting, which prefers co-citations with a greater distance. When $a$ is zero, CPI fixed to 1 and therefore independent from the link distance. In this case, CPA equals CoCit, since only the number of co-citations is counted, e.g. proximity has no effect. The optimised CPI model for Wikipedia articles will be defined in the experiment (Section 5.1).

#### 4.2.3.3 Intermediate Results

Finally, a hash value of Page A and B is stored in the LinkTuple object. This hash value is used as key in all MapReduce operations. A single hashed key led to significant performance improvements, since the Shuffling phase requires many comparison operations, when sorting and grouping all intermediate results by their keys. And, comparison of a single hash value can be done faster than a comparison of the two page names.

The processed LinkTuple objects are the output of the Map task. At this stage the amount of intermediate results is at its peak, therefore we made at this stage the effort of the hash key optimisation. The Mapper returns more than 37 billion records, each uncompressed record is approx. 932 bytes in size, as result all intermediate results are approx. 32 TB in size. The intermediate results are shrunk by a then performed Combine operation. It lowers network traffic and enhances performance. The Combiner functionality is not explained here, as it runs the same operations as the Reducer does.

#### 4.2.3.4 Final CPI Computation

The Reduce operation is performed on a dataset of LinkTuple objects that are grouped by the hash value as key. Compared to the Mapper the functionality of this operation is quite simple.

$$\mathrm{CPI}(a,b) = \sum_{j=1}^{m} \Delta_j(a,b)^{-\alpha} = \sum_{j=1}^{m} \begin{cases} \left|v_{a,j} - v_{b,j}\right|^{-\alpha} & v_{a,j} > 0 \wedge v_{b,j} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

As shown in Equation 4.3, the Reducer calculates the total CPI for a document pair $a, b$ by adding up CPI values of $a, b$ for all $m$ documents.

For better understanding, we illustrate the document-based and final CPI computation in the following example, where the corpus consists of the documents $d_{1\text{-}4}$ with $m$=4 and their respective links. CPI parameter $\alpha$ is set to 1. The matrix below represents the link positions within all documents, while each column represents a document $j$ and each row represents a link to its respective document $i$, as shown in Section 4.2.3.2.

| | | Documents | | | |
|---|---|---|---|---|---|
| | | $d_1$ | $d_2$ | $d_3$ | $d_4$ |
| Links | $d_1$ | 0 | 50 | 5 | 95 |
| | $d_2$ | 30 | 0 | 90 | 30 |
| | $d_3$ | 40 | 10 | 0 | 35 |
| | $d_4$ | 50 | 10 | 95 | 0 |

**Figure 4.8: Link-Position Matrix. Columns represent documents, while rows represent links.**

Document $d_4$ is the citing document in the illustration of CPA's concept (Figure 2.8), while documents A-C correspond to $d_{1\text{-}3}$, where the citation markers of $d_2$ and $d_3$ are in a more close proximity compared to $d_1$ and $d_2$ and therefore $d_2$ and $d_3$ are stronger related than $d_1$ and $d_2$. Equation 4.4 and 4.5 reflect this by computing CPI values of $d_2$ and $d_3$ as well as of $d_1$ and $d_2$ based on document $d_4$.

$$\mathrm{CPI}_4(d_2, d_3) = \Delta_4(d_2, d_3)^{-\alpha} = \left|v_{2,4} - v_{3,4}\right|^{-\alpha} = |30 - 35|^{-1} = \frac{1}{5} \quad (4.4)$$

$$\mathrm{CPI}_4(d_1, d_2) = \Delta_4(d_1, d_2)^{-\alpha} = \left|v_{1,4} - v_{2,4}\right|^{-\alpha} = |95 - 30|^{-1} = \frac{1}{65} \quad (4.5)$$

As $\mathrm{CPI}_4(d_2, d_3) = \frac{1}{5}$ is greater than $\mathrm{CPI}_4(d_1, d_2) = \frac{1}{65}$, B and C are determined as more similar than A and B based on document D. For a corpus-wide similarity measure of $B$ and $C$, the final CPI value is calculated as shown in Equation 4.6.

$$\mathrm{CPI}(d_2, d_3) = \sum_{j=1}^{4} \Delta_j(d_2, d_3)^{-\alpha} = \mathrm{CPI}_1(d_2, d_3) + \mathrm{CPI}_4(d_2, d_3) = \frac{1}{20} + \frac{1}{5} = 0.25 \quad (4.6)$$

The resulting final CPI value for the documents $d_2$ and $d_3$ is 0.25.

This operation is also performed with the Co-Citation strength in appropriate manner. The resulting dataset contains all existing co-citations, i.e. document pairs with a similarity assessment, which occur in the Wikipedia test collection. Finally, the result set is stored as CSV file in HDFS.

### 4.2.4 Computing Quasi-Gold Standard

The extraction of the "See also" links is implemented in a similar way to CPA and Co-Citation.

The Wikipedia XML dump is used as input, too. Articles are also extracted in the same way, but instead of removing the "See also" section, all other content is discarded. Next, all links are extracted out of the "See also" section. The data is then mapped to a 3-tuple consisting of the article name of the link target, the article name of the article that includes the "See also" link and the number of links within the section (Figure 4.9). Also, HDFS is used to store the result dataset as CSV file.

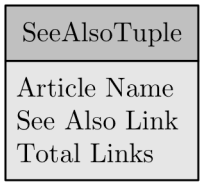



**Figure 4.9: SeeAlsoTuple**

### 4.2.5 MoreLikeThis Implementation

The implementation of the text-based similarity measure does not rely, like CPA and CoCit, on the Hadoop ecosystem or Apache Flink. Instead, the out-of-the-box solution Elasticsearch performs the task of text-based document retrieval and therefore we did not implement the actual similarity algorithm by ourselves. Moreover, we implemented a Java application that uses the Elasticsearch API client to write all Wikipedia articles to Elasticsearch and to retrieve topically related documents for "See also" article by performing MLT queries (Section 4.1.4.3).

Figure 4.10 illustrates the conceptual steps that are run to create the MLT results.

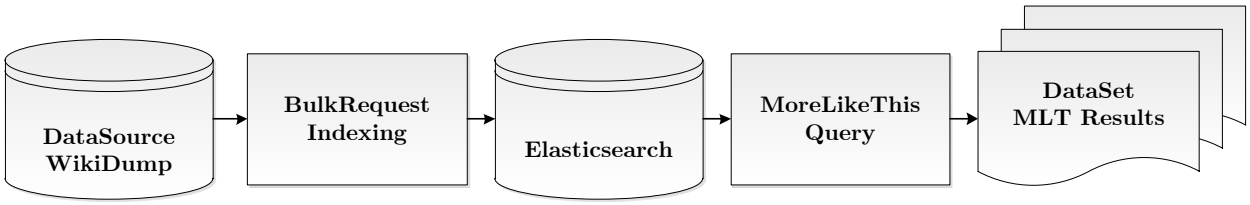



**Figure 4.10: MoreLikeThis implementation. MLT-queries to Elasticsearch for "See also" articles.**

At first the application extracts Wikipedia articles for the XML dump, following the same pattern as used in the previous programs (Section 4.2.3.1). For indexing we use the *BulkRequest*[8] functionality of Elasticsearch API client to add multiple articles at the same time to Lucene's index. Otherwise the term-document matrix related-factors in the scoring function would be

[8] https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html

updated after each insertion (Equation 4.1). Concurrent requests delay these updates and therefore increase the indexing speed.

The second step deals with retrieving similar Wikipedia articles. As we only evaluate articles, which contain a "See also" section, we query only those articles to avoid unneeded results. Performing a MLT query to Elasticsearch can retrieve those documents, thereby Elasticsearch response with a JSON object that contains the documents, which are similar to a chosen input document (Section 4.1.4.3). Aside from the actual document name, each result also includes a numeric score that determines its rank.

Lastly, document name, score as well as the input document are stored in a CSV file on the local file system.

### 4.2.6 Evaluation Implementation

In the final evaluation process, we merge all the previously generated datasets, find all relevant documents, which were retrieved by each document similarity measure, based on the "See also" dataset and calculate a MAP score for each query and for CoCit, CPA and MLT. Figure 4.11 illustrates the evaluation program that is implemented as Flink job.

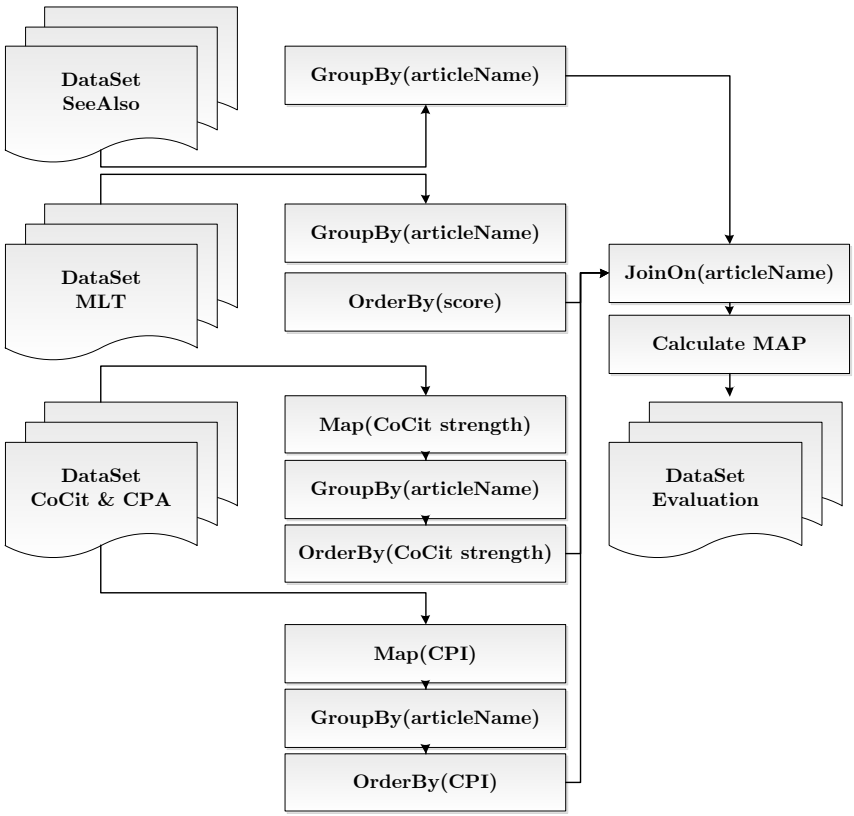



**Figure 4.11: Evaluation program. Intermediate datasets are ordered and joined by article name.**

The evaluation program has three input datasets, which are all stored in HDFS:

1. "See also" link dataset, i.e. gold standard (Figure 4.9)
2. MLT dataset, i.e. documents retrieved for "See also" articles by MLT (Section 4.2.5)
3. CoCit and CPA dataset, i.e. *LinkTuples* with CoCit and CPA rankings (Figure 4.7)

The Flink job processes all input datasets in parallel manner. The "See also" link dataset is grouped by the name of the article containing the "See also" links, resulting in a 2-tuple consisting of the article name and a list of "See also" links. The MLT dataset is as well grouped by the article name, but also ordered by the MLT score to get the retrieved documents in ranked order.

As the results of CoCit and CPA are stored in the same dataset, their processing also involves an additional *Map* operation. Thus, each LinkTuple is mapped to a CoCit result, which has the CoCit strength as score value, and to a CPA result, which has the CPI as score value. When initially generating the CoCit and CPA dataset, the article names were forced to be in alphabetical order to avoid duplicates. In the evaluation, we need to recreate these duplicates, because a *LinkTuple* of Page A and Page B is not only a recommendation of Page A for Page B, but also a recommendation of Page B for Page A. Therefore, all *LinkTuples* are also mapped to their representative with reverse alphabetical order. The intermediate results of CoCit and CPA are as well grouped by the article name and ordered by their score values. We needed to implement the OrderBy operation by ourselves as Flink's sorting was malfunctioned.

Next, the program joins the datasets with matching article names. As a result, we have data records of Wikipedia articles with the matching "See also" links and the retrieved documents of each similarity measure in ranked order. Based on this data, the relevance of the retrieved documents is judged. If a retrieved document, exists in the list of "See also" links, it is determined as relevant. Depending on relevance and ranking, the MAP score is calculated.

The qualitative and quantitative evaluation is performed based on the resulting dataset.

### 4.2.7 Runtimes

In the following, we report and compare the runtimes for computing all results of CoCit, CPA and MLT. We executed all applications with the same experimental setup from Section 4.2.1, while neither Apache Flink nor Elasticsearch had been optimised for runtime performance. For the experiment each application was run several times for debugging and testing purpose. The following runtimes are average values:

**Table 4.1: Runtimes of each program.**

| **Program** | **Runtime** |
|---|---|
| CoCit & CPA | 3h 10min |
| MLT total | 12h 30min |
| *a) Indexing* | *7h 30min* |
| *b) Retrieval* | *5h* |
| "See also" extraction | 1h 15min |
| Evaluation | 2h 30min |

Table 4.1 shows the runtimes of each program. When comparing the runtimes of the text-based to the citation-based similarity measure, we see that the computation of CoCit and CPA is 9h 20min faster than MLT in total. The citation-based computation also includes all Wikipedia articles (4.6 million), while the retrieval process of MLT is only performed for articles with "See also" section (0.7 million).

Based on this information, we conclude that MLT, i.e. Lucene, involves a more extensive computation than CoCit or CPA, since MLT has a much longer runtime.

# 5 Results

The goal of this section is to compare the text-based MLT and the citation-based document similarity measures CoCit and CPA regarding their recommendation quality. The relevance of the retrieved documents is measured by our "See also" quasi-gold standard. We start by determining a value of $\alpha$ for calculating CPA results, then, continue with a quantitative evaluation. We group the result set by different metrics to show performance correlations. Then, we evaluate four samples qualitatively. Lastly, we additionally evaluate the similarity measures based on a Wikipedia clickstream dataset.

## 5.1 CPA Optimisation

As we use a dynamic CPI model instead of static CPI values as suggested by Gipp and Beel (Section 4.2.3), we need to optimise CPA for Wikipedia articles, before comparing it with other similarity measures. Therefore, we need to find an optimised value for the constant $\alpha$.

The optimised α value should result in the maximum of our performance measure, i.e. Mean Average Precision. Therefore, we performed CPA with $\alpha$ values in a range of 0 to 100. Next, we evaluated the retrieved results of each batch by calculating the MAP scores. This optimisation process benefits from the automated evaluation, as the effort to perform this process up to one hundred times is low compared to a user-based evaluation.

Figure 5.1 and Figure 5.2 plot a series of MAP scores dependent on $\alpha$ values (range 0 to 100). For each batch all 777,047 Wikipedia article with a „See also" section were used.

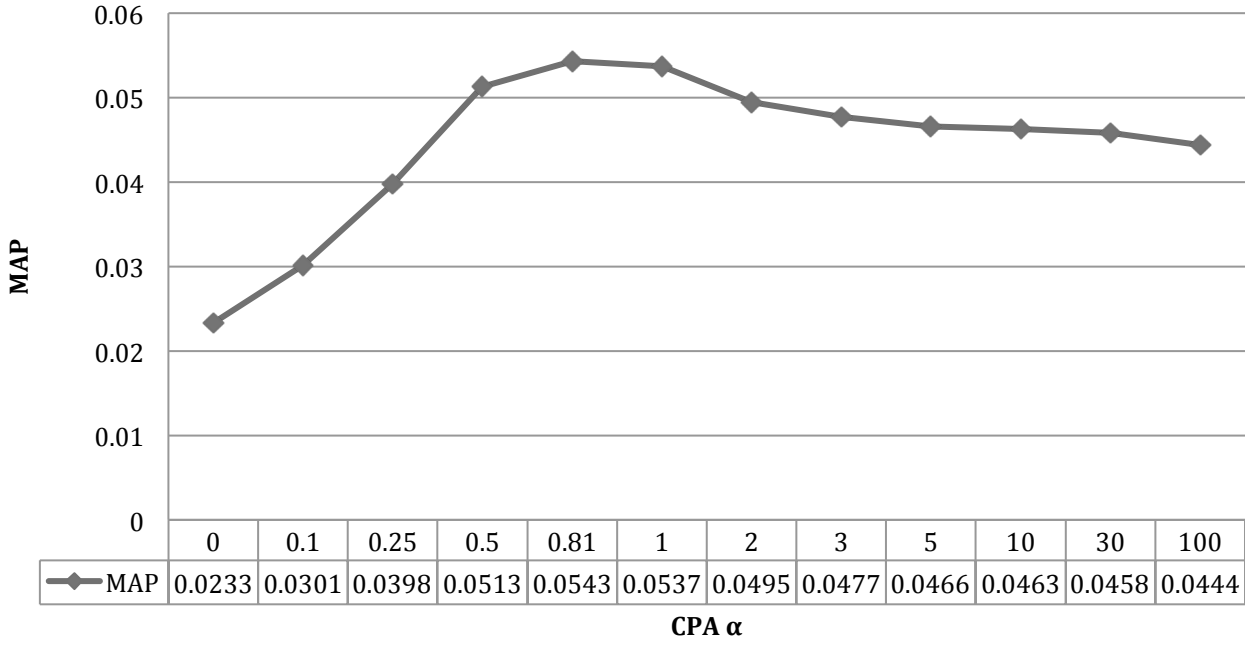



**Figure 5.1: MAP score of CPA linked to α parameter, range 0-100, max. MAP at α = 0.81**

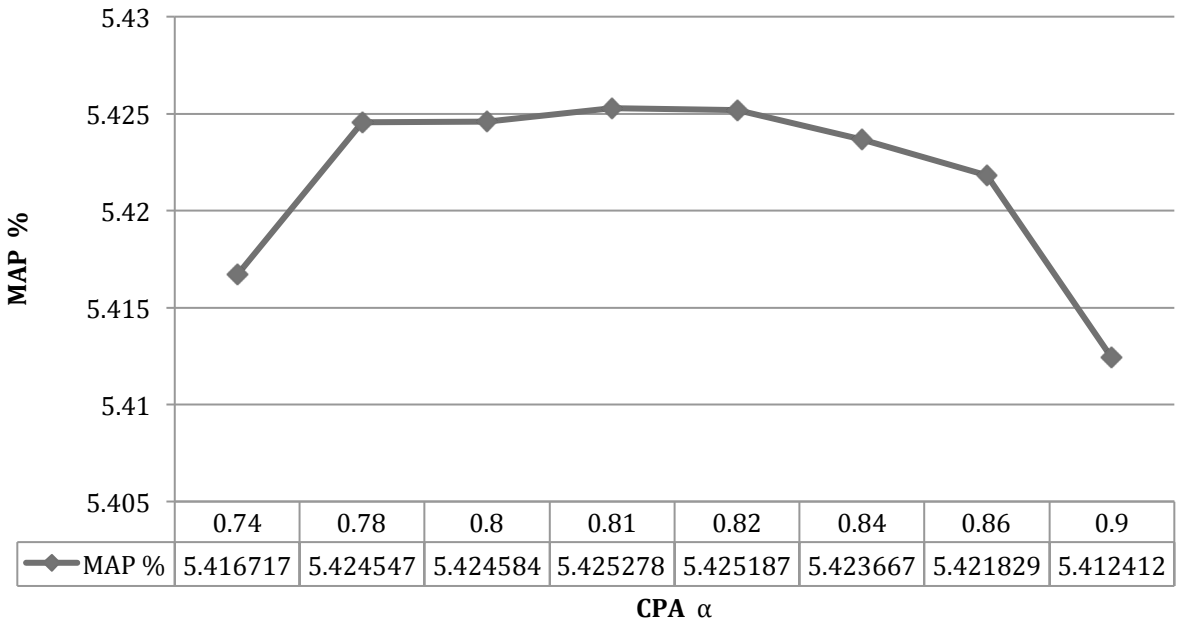



**Figure 5.2: MAP scores in % of CPA linked to α parameter, range 0.74-0.9, max. MAP at α = 0.81**

The plot shows that MAP reaches its maximum of 0.05425278 MAP, when $\alpha$ is set to 0.81. We choose to optimise $\alpha$ as accurate to two decimal places, since the effect of change is neglect able at this scale. Changing $\alpha$ from 0.8 to 0.81 lowers MAP only approximately 0.01%.

The minimum MAP is scored, when $\alpha$ is set to zero. At this point CPA does not involve the co-citation proximity, i.e. CPA is equal to CoCit. In other words, CoCit is a special case of CPA, which is not the optimum of CPA. In $\alpha$ range from 0 the MAP score is increasing until its maximum. Values of $\alpha$ above 0.81 resulted in a decreasing MAP score.

As a result, we state that CPA performs best in terms of MAP when $\alpha$ is set to 0.81. This $\alpha$ value is used in the CPI formula in the following evaluation.

## 5.2 Quantitative Evaluation

In the following two sections, we show the results of our quantitative evaluation.

First, we evaluate the whole test collection and find out that we need to limit the queries to have same-sized result sets for each document similarity measure.

Second, we evaluate subsets dependent on article properties. In both evaluations, MLT delivers the best MAP score, second best is CPA and CoCit scores the lowest MAP.

### 5.2.1 Overall Evaluation

Figure 5.3 shows the Mean Average Precision (MAP) of each document similarity measure for all queries, where Wikipedia articles with a "See also" section represent queries and the "See also" links determine, whether a retrieved document is relevant or not. The test collection contains 777.047 articles with "See also" sections and 1.949.153 "See also" links, i.e. links to sources considered as relevant.

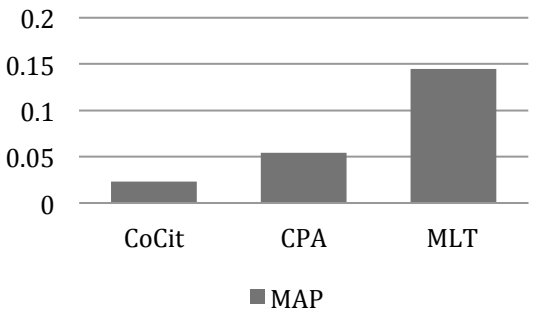



**Figure 5.3: MAP score of CoCit, CPA and MLT.**

**Table 5.1: Number of retrieved documents, absolute and relative relevant documents with $k$=10 and MAP score by CoCit, CPA and MLT.**

| | CoCit | CPA | MLT |
|---|---|---|---|
| **Retrieved documents** | 6,935,605 | 6,935,605 | 7,767,215 |
| **Relevant documents** | 118,447 | 232,357 | 457,444 |
| % | 1.71% | 3.35% | 5.89% |
| **MAP** | 0.023301 | 0.054252 | 0.14452159 |

The number of retrieved documents per query is limited to $k = 10$ (Section 0). On all queries CoCit and CPA retrieve in total 6,935,605 documents, of which 1.71% of CPA and 3.35% of CoCit are judged as relevant. The number of retrieved documents of CoCit and CPA is the same, since both work on the same data. MLT retrieves more documents than the citation-based approaches (7,767,215 retrieved documents), because there are some articles that do not have any inbound links (Figure 3.3). CoCit and CPA rely on these links, therefore, both similarity measures cannot retrieve any documents, if no inbound links exists. On the other hand, there are as well queries, to which MLT retrieves a few or no documents. Nonetheless, MLT retrieves more results than CoCit and CPA, because the test collection consists of fewer articles with low linkage than articles with few words. Still, MLT retrieves the greatest percentage of relevant documents (5.89%).

But the fact, that the document similarity measures retrieve a different amount of documents, makes the comparison of their results less convincing. For this reasons, we need to limit the evaluation to those queries, to which the number of retrieved documents of CoCit, CPA and MLT is the same.

Unless otherwise specified, we use as our query set articles that meet the following two criteria:

a) The article includes a "See also" section.

b) All tested similarity measures can compute a similarity score for the article in question.

The resulting query set includes 679,309 articles.

Figure 5.4 and Table 5.2 show the impact on the evaluation. For the limited query set, the performance of citation-based measures improves slightly. Yet, overall ranking remains unaltered: MLT performs best in quantitative analysis for identifying related Wikipedia articles.

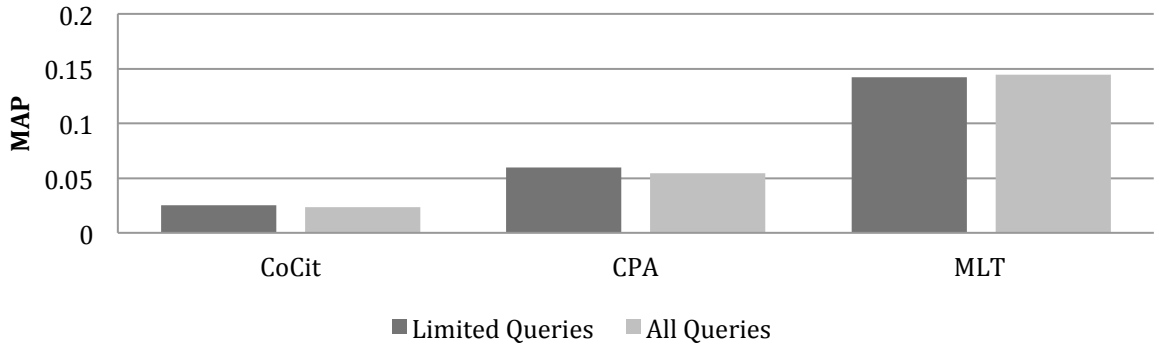



**Figure 5.4: Impact of query limitation on MAP.**

**Table 5.2: Impact of query limitation on retrieved documents. Impact is the respective difference of before and after applying query criteria.**

| | **CoCit** | **CPA** | **MLT** |
|---|---|---|---|
| **Retrieved documents** | 6,792,882 | 6,792,882 | 6,792,882 |
| *Impact* | -142,723 | -142,723 | -974,333 |
| **Relevant documents** | 113,843 | 227,264 | 405,609 |
| *Impact* | -4,604 | -5,093 | -51,835 |
| % | 1.68% | 3.35% | 5.97% |
| *Impact* | -0.03% | 0% | +0.08% |
| MAP | 0,025261 | 0,059985 | 0,141965 |
| Impact | +0,001959 | +0,005732 | -0,002556 |

### 5.2.2 Subset Evaluation

In Section 3.2, we introduce Wikipedia as a diverse test collection. The following analysis evaluates Wikipedia from different perspectives by dividing the whole evaluation corpus in 10-quantiles (deciles) depended on four article properties, which are article length, outbound links, inbound links and number of "See also" links.

This evaluation investigates the effect of article properties on the performance of the document similarity measures. We chose the four properties, because they were easy to extract and we except them to affect the similarity measures.

None of the article properties led to the creation of a query subset, which resulted in a different performance ranking. In any case MLT is ranked first, CPA second and CoCit third.

Figure 5.5 – 6.8 illustrate the MAP scores of each document similarity measure depending on the respective article property, while Table 5.3 – 6.5 name the interval borders of each decile.

#### 5.2.2.1 Words

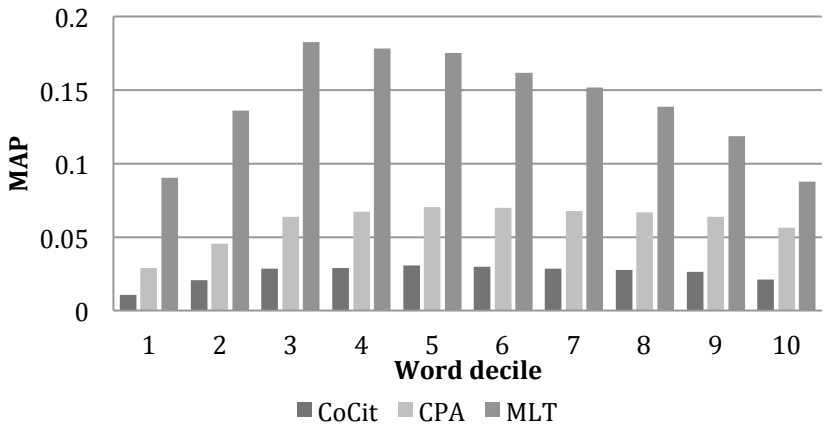



**Figure 5.5: MAP per word decile. Avg.: 1202 words**

**Table 5.3: Word deciles.**

| Deciles | Interval borders | |
|---|---|---|
| 1 | 9 | 126 |
| 2 | 126 | 207 |
| 3 | 207 | 310 |
| 4 | 310 | 429 |
| 5 | 429 | 585 |
| 6 | 585 | 790 |
| 7 | 790 | 1089 |
| 8 | 1089 | 1602 |
| 9 | 1602 | 2788 |
| 10 | 2788 | 75178 |

MLT, CoCit and CPA perform best, when the article length is in the range of the third to fifths decile. MLT's MAP score decreases in the lower and upper deciles, whether CoCit and CPA show only in the first and second decile a significant decrease of their MAP score.

The length of an article is measured in number of its words. On average an article of the evaluation corpus consists of 1202 words, but only a third of all articles has more than 1000 words. In other words, the majority are relatively short articles.

We expected the article length to strongly affect MLT, because this method relies on the article text. On the contrary, the citation-based similarity measures are independent from the actual article content or length, since they measure the citation in other articles. For this reason, we assume that the number of words has no or little impact on the performance of CoCit and CPA.

Figure 5.5 shows that articles with 207-585 words result in the best MAP score by MLT, whereas the $10^{th}$ decile ($>$ 2788 words) yields in the lowest MAP score of MLT, even lower than the second-worst performing $1^{st}$ decile (9-126 words). This pattern implies that MLT needs a minimum article length, at least 126 words, to work properly, but also faces a maximum of approximately 2750 words, that decreases the performance of MLT. Given the concept behind MLT, this outcome does not surprise. When an article consists of only a few words, it is more difficult to find other articles matching these words and rank the articles by their degree of similarity. Short articles are hardly distinguishable by topic. Similarly, very long article with ten thousands of words that cover several subtopics are also hard to distinguish by their vocabulary, because the vocabulary of each subtopic might vary that much that it is difficult to determine a set of words, which represents the whole article. What number of words is too less or too much, depends on the respective implementation of VSM or TF-IDF. A different weighting in the scoring formula of MLT (Section 0) might lead to different article length dependencies.

On contrary, the decile analysis supports our assumption of the citation-based similarity measures. The article length has little impact on their performance. CPA score stays stable in the range of 0.063 to 0.070 MAP in the $3^{rd}$ to $9^{th}$ decile. CoCit shows similar results at a lower level between 0.026 and 0.03 MAP. The maximum MAP score of both citation-based systems is in the $5^{th}$ decile. Low MAP scores are in the outer deciles of the $1^{st}$, $2^{nd}$ and $10^{th}$ decile. A possible explanation for the low performance of short article is that a low number of words might indicate poor article quality, and therefore these article receive less citations. But we cannot prove this assumption, as there is no clear indicator for article quality.

#### 5.2.2.2 Outbound Links

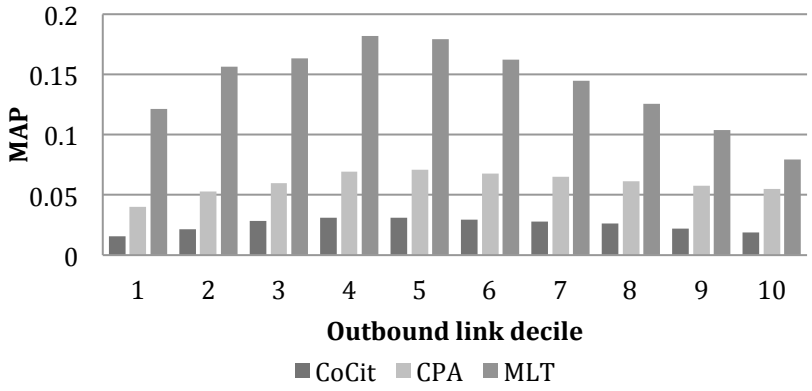



**Figure 5.6: MAP per outbound link decile. Avg.: 54.3 links**

**Table 5.4: Outbound link deciles.**

| Deciles | Interval borders | |
|---|---|---|
| 1 | 0 | 5 |
| 2 | 5 | 9 |
| 3 | 9 | 13 |
| 4 | 13 | 18 |
| 5 | 18 | 24 |
| 6 | 24 | 32 |
| 7 | 32 | 45 |
| 8 | 45 | 67 |
| 9 | 67 | 120 |
| 10 | 120 | 8624 |

MAP scores depended on outbound links show a similar pattern as article length in Figure 5.5. A dependency between the two article properties words and outbound links causes the similarity of both charts. The sample correlation coefficient of word decile and outbound link decile is approximately 0.89. Also, Figure 3.5 illustrates that the majority of articles contain links at a similar frequency, e.g. on average 28 words are written per outbound link.

MLT scores high MAP in the centric deciles and low MAP in the outer deciles. CoCit and CPA have low MAP scores in the first two deciles, whether deciles above two have a higher and stable MAP score.

#### 5.2.2.3 Inbound Links

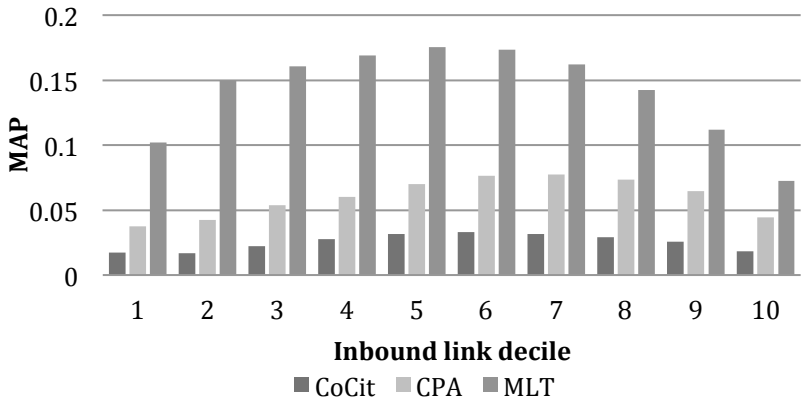



**Figure 5.7: MAP per inbound link decile. Avg.: 56.2 links**

**Table 5.5: Inbound link deciles.**

| Deciles | Interval borders | |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | 2 |
| 3 | 2 | 3 |
| 4 | 3 | 4 |
| 5 | 4 | 7 |
| 6 | 7 | 10 |
| 7 | 10 | 16 |
| 8 | 16 | 29 |
| 9 | 29 | 71 |
| 10 | 71 | 39873 |

MLT achieves the maximum MAP score in $5^{th}$ decile with a symmetric decrease to outer deciles, whereas the scores of CoCit and CPA are shifted to higher deciles. CoCit has its maximum MAP score in $6^{th}$ decile and CPA its in $7^{th}$ decile.

There is an uneven distribution of inbound links among the evaluation corpus. Only a few article receive many inbound links, as the $10^{th}$ decile includes all articles with more than 71 links. On contrary there are many articles with a little amount of inbound links.

Inbound links as data source for document similarity measures are essential for citation-based systems but do not affect text-based systems. Accordingly, we assume CoCit and CPA to perform better, when the number of inbound links increases. Speaking of deciles, the expected maximum is in the $10^{th}$ decile. Secondly, the performance of MLT is supposed to be stable throughout all deciles.

Nonetheless, Figure 5.7 disproves our assumptions. To be more precise, MLT's performance is far away from being stable. The MAP score of MLT differs from the lowest decile ($10^{th}$ with 0.07 MAP) to the highest decile ($5^{th}$ with 0.17 MAP) approximately 40%. MLT also shows a decreasing MAP towards the $1^{st}$ decile. We cannot provide a direct explanation for this outcome, but we see some possible coherences: Wikipedia articles with a high number of inbound links are mainly in the category of common or abstract nouns, famous people or geopolitical entities [42]. In other words, these articles tend to cover more general topics. Thus, the "See also" sections of these articles might also contain links of more general articles. Our qualitative shows, however, that MLT primarily retrieves more specific articles (Section 5.3). This might be the reason for MLT scoring a low MAP at a high number of inbound links.

Furthermore, a large number of inbound seems to have an negative effect on the CoCit and CPA performance, because the $10^{th}$ decile is third lowest performing decile after the $1^{st}$ and $2^{nd}$. CoCit performs best in the $6^{th}$ decile with 0.03 MAP, whereas CPA has its maximum performance in the $7^{th}$ decile with 0.07 MAP. These results contradict our assumption as $6^{th}$ and $7^{th}$ decile consist of articles, which have only 7-16 inbound links. It is evident that CPA performs worse when only a low number of inbound links, i.e. a few co-citations, is available. More surprisingly is that the MAP also drops in the $10^{th}$ decile, i.e. with many co-citations. We explain this results with the calculation of the CPI: Co-citations that have a higher proximity are valued more than co-citations with low proximity. So also co-citations with low proximity count. Therefore, the value of proximity decreases as the total number of co-citations increases. Many co-citations with low proximity can result in a higher CPI than a few co-citations with high proximity. As result, at a high number of inbound links CPA converges to CoCit. Thus, CPA also drops in performance.

#### 5.2.2.4 "See also" Links

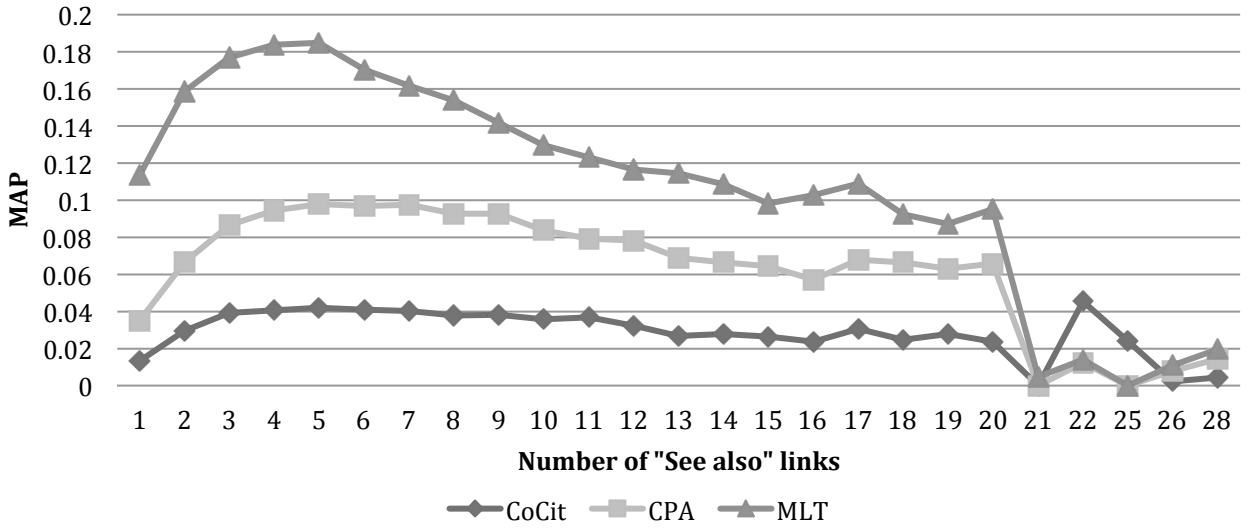



**Figure 5.8: MAP per number of "See also" links.**

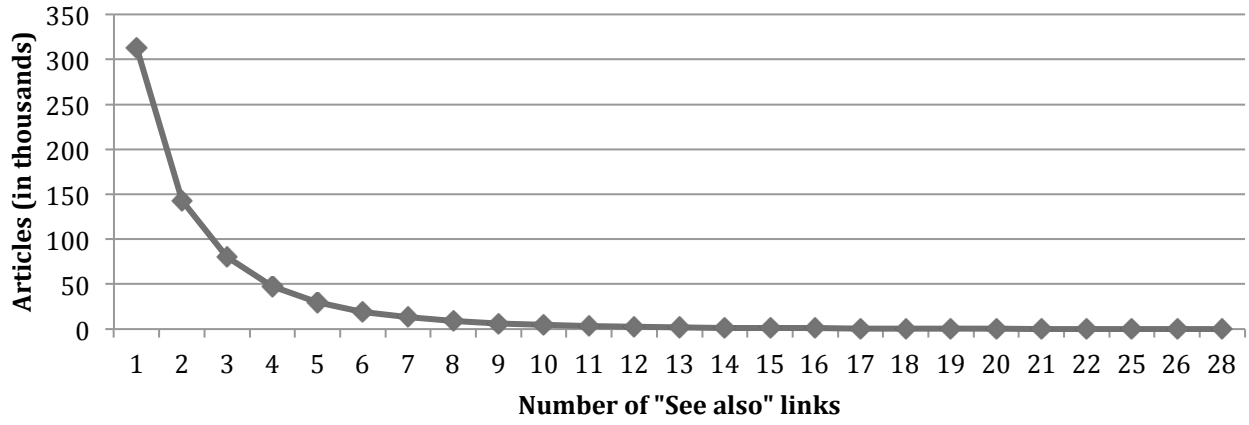



**Figure 5.9: "See also" links per article in evaluation query set.**

Figure 5.8 shows the relation of MAP to the number of "See also" per article, i.e. number of relevance judgments by the quasi-gold standard. MAP of MLT, CoCit and CPA starts low at one "See also" link per article, then MAP increases until its reaches the maximum at five links with MLT's score 0.184 MAP, CPA's score 9.097 MAP and CoCit's score 0.042 MAP. As the number of "See also" links exceeds five, the MAP decreases.

MAP scores for articles with more than 20 "See also" links are inconstant, because there are only a few articles (1-3) having this amount of links.

#### 5.2.2.5 Relevant Documents

Besides calculating the MAP scores of each document similarity measure, we counted additionally the number of retrieved documents, which are determined as relevant by the quasi-gold standard.

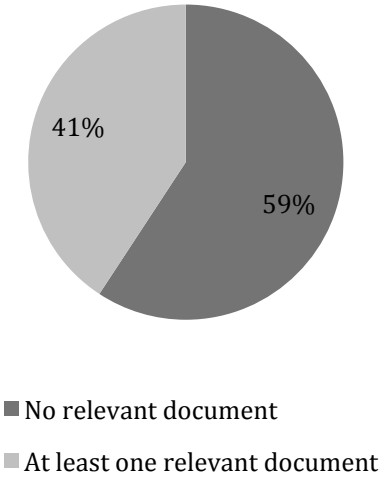



**Figure 5.10: Relevant documents retrieved.**

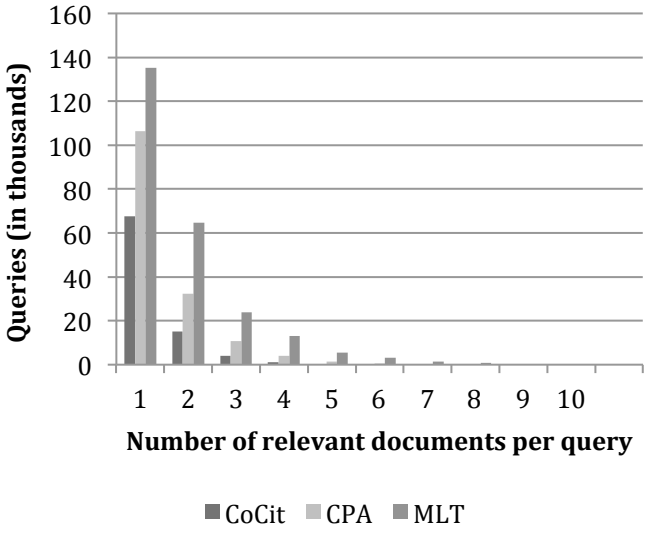



**Figure 5.11: Relevant documents per query.**

Figure 5.10 illustrates the percentage of queries, to which none of the document similarity measures retrieved any relevant documents. No relevant documents had been retrieved for total 460.325 queries (59%). In Figure 5.11 we see the number of relevant documents per query matched with their frequency. For MLT, only 1.5% queries led to five or more relevant documents.

These numbers rise to the following questions: How well do the tested similarity measures perform in the 59% queries, where none relevant document had been retrieved? Are the documents retrieved in these 59% truly irrelevant or only missed out by the "See also" links, i.e. false negative errors? A purely quantitative evaluation cannot answer these questions. Also, we do not except the similarity measures to retrieve mainly irrelevant. Thus, we also investigate in Section 5.3 samples of the test collection, whereby we compare "See also" links to our relevance judgements.

## 5.3 Qualitative Evaluation

The qualitative evaluation provides a deeper insight into the documents retrieved by our document similarity measures and the relevance judgements by our quasi-gold standard. Besides seeing that MLT tended to retrieve documents lexically close to the query, whereas Co-Citation and Co-Citation Proximity Analysis favoured documents from a broad topical spectrum, we acknowledged that the "See also" links often miss out relevant documents.

After analysing the results in a quantitative manner, we already can approve MLT as best performing document similarity measure in identifying related Wikipedia articles. Yet, the question about the reasons for this outcome needs to be answered, since the quantitative evaluation does not reveal any major differences in ranking depending on the tested article properties. Therefore, we choose a sample of representative queries to analyse the documents retrieved by each similarity measure. We tested four different articles, chosen for their diversity and comprehensibility:

1. **Newspaper**: General topic, above average length and high number of inbound links.
2. **Technical University of Berlin**: Educational institution, location related, medium length and medium inbound links.
3. **Finance Act**: Political topic, recurring event, medium length and low number of inbound links.
4. **Site map**: Niche technical topic, short article and low number of inbound links.

Ollivier and Senellart chose articles for the same reasons in their study [41]. But Wikipedia changed from 2006 to 2014. So did the properties of their articles. Therefore, we evaluate different articles.

In the following sections, we analyse the retrieved documents for each article one by one. The sample data itself can be found in Section 5.4. To summarise, the qualitative evaluation does not support the performance ranking of the quantitative evaluation, which is mainly due to false negative relevance classification by "See also" links. Based on our relevance judgements, we find CPA and MLT at a similar recommendation quality level, as CPA retrieves on average 8 relevant documents, whereas we classify 7.75 documents of MLT's results as relevant. In particular, MLT retrieves in 3 of 4 samples only relevant documents. Only in the sample of "Site map", MLT retrieves only one relevant result. On contrary, CPA provides only for the sample of "Finance Act" complete relevant results, while each of the other four samples retrieves 2-4 irrelevant documents. However, CoCit is also in the qualitative evaluation the worst performing similarity measure.

### 5.3.1 Newspaper

The Wikipedia article "Newspaper" contains general information about newspaper as periodical publication, its historical development, categories, formats and other newspaper related topics. The article consists of 6313 words and is linked by 7611 other articles. The "See also" section includes two links to newspaper related lists: "List of newspaper comic strips" and "Lists of newspapers".

Neither MLT, nor CPA, nor CoCit retrieve any of the "See also" links. However, in our point of view all of the ten MLT's results are relevant, CPA retrieves only two of ten documents, which we determine as irrelevant and CoCit also retrieves five of ten relevant documents. Thus, our relevance judgment contradicts the quasi-gold standard for this sample.

All documents retrieved by MLT are newspaper related: "Online newspaper" and "Weekly newspaper" as newspaper types and journalism related articles like "Community journalism" or "Journalism and freedom", but most articles are about actual newspaper publications, e.g. "Lebanon Daily News". CPA tends to retrieve a broader spectrum of related topics, e.g. newspaper format ("Tabloid", "Broadsheet" etc.) or other medias ("Television", "Book" etc.). Yet, two of CPA's results ("United States" and "English language") are not topically relevant. CoCit retrieves many irrelevant articles from the geopolitical category ("United States", "New York City" etc.), as well as newspaper formats ("Tabloid", "Broadsheet" etc.).

The results of this sample are typically for all similarity measures. MLT is more specific than the citation-based approaches and CoCit has a higher error rate than CPA. Also, we see the irrelevant geopolitical articles, which are frequently co-cited with "Newspaper" by articles about newspaper publications, e.g. "New York Times", but are not topically related. Co-citations with these article can be considered as irrelevant for identify related articles, as they occur in a large fraction of the test collection.

### 5.3.2 Finance Act

The Wikipedia article "Finance Act" refers to a series of budgetary legislation issued by the UK Parliament. It provides an overview about the topic and gives details about each year. It has 1246 words and a low number of 32 inbound links. The "See also" links consist of a link to a "List of short titles" (legislations) and 15 links to the Finance Acts from 1999 to 2014, whereas the articles of Finance Act 2010-2014 do not exist in Wikipedia.

MLT retrieves all articles of the Finance Acts from 1999 to 2009. They show a clear focus on the term "Finance Act". All retrieved article contain of a low number of words and have at the same time a very similar vocabulary. Thus, the complete results set is relevant by the "See also" links. In contrast, CoCit's and CPA's results cover other budgetary or political topics, e.g. "Parliament of United Kingdom" or "Income Tax". None of these are part of the gold standard

and, therefore, not counted as relevant, even if we judge all CPA's results and nine of the ten CoCit results as relevant. "New York Times" is the article retrieved by CoCit that is irrelevant in our point of view.

This example shows a problem of the "See also"-based evaluation: Given only the relevance judgments by our quasi-gold standard, MLT shows an overwhelming advantage over the citation-based measure. MLT retrieved only relevant documents, whereas CoCit and CPA retrieved none. Yet, in our point of view the results of CoCit and CPA reveal the same quality as the MLT result. So "See also" links do not serve as a valid gold standard in this sample.

### 5.3.3 Site Map

A site map is a list of pages of a web site accessible to web crawlers or users, often used for search engine optimisation. The Wikipedia article describes different types of site maps and gives technical background information. The article has a medium length and only 12 inbound links. It also includes six "See also" links to article related to web sites and search engines.

Compared to the other samples, MLT retrieves for this article many irrelevant results. Only one document ("Search engine submission") is determined as relevant found on "See also" links and our judgement. The other documents are lexically similar, because they are all types of maps, but in the semantically context those documents are irrelevant. On contrary, the citation-based similarity measures retrieve a higher number of relevant documents. All of CPA's and CoCit's results are in the field of Internet technologies, whereas CPA retrieves more documents relevant to "Site map". Again, CoCit tends to retrieve a broader spectrum of related articles.

This sample shows the limitation of text-based similarity measures, when a lexical similarity does not result in a semantic similarity. Co-Citation and CPA do not face this problem and therefore retrieve more relevant documents.

### 5.3.4 Technical University of Berlin

The article of the Technical University of Berlin includes information about its history, campus, organisation and a list of notable alumni and professors. It consists of 2542 words and receives a medium number of 311 inbound links. The "See also" refers to six articles about other universities located in Berlin.

None of the documents retrieved by MLT, CPA or CoCit are relevant according to the "See also" links. Yet, we assess 10 MLT results, 8 CPA results and 5 CoCit results as relevant. MLT retrieves articles of TU Berlin employees, related projects and other universities. CPA's and CoCit's results range from location-, university- and science-related to historical articles. Some locations like "Berlin" or "Charlottenburg" are topically relevant, whereas "Germany" or "Hamburg" do not seem to be particularly useful for an average reader. Historical articles

("World War II" or "German Empire") also have in our point of view little semantic relationship to the topic.

All document similarity measures retrieve the relevant article "Humboldt University of Berlin". Still, it is not recognised by the "See also" links, even if a slightly different named link "Humboldt Universität zu Berlin" exists in the "See also" section. However, both links refer to the same Wikipedia article, but they are not evaluated as identical links, because "Humboldt Universität zu Berlin" points to a page, that redirects the reader to "Humboldt University of Berlin". The "See also" links of "Freie Universität Berlin", „Universität der Künste" and „Beuth Hochschule für Technik Berlin" are also referring to a redirect.

## 5.4 Sample Data

Tables 5.7 – 5.10 on the following pages list the results retrieved by CoCit, CPA and MLT for each sample query. Each tables shows for CoCit, CPA and MLT the rank, the retrieved document, its score and is additionally marked when it is determined as relevant, where S stands for relevance judgment by "See also" links and A is relevance judged by the author of this thesis. The last row of each table sums up the total number of relevant documents.

**Table 5.6: Information on article properties and sum of relevant documents of each sample query.**

| **Article** | **Inbound links** | **Outbound Links** | **Words** | **Headlines** |
|---|---|---|---|---|
| Finance Act | 32 | 40 | 1246 | 14 |
| Site map | 12 | 26 | 841 | 6 |
| Newspaper | 7611 | 262 | 6313 | 33 |
| Technical University of Berlin | 311 | 188 | 2542 | 10 |

**Table 5.8: Sample results for "Newspaper".**

**Article** Newspaper — *S = Relevance by "See also" links*

**"See also" links** List of newspaper comic strips, Lists of newspapers — *A = Relevance by thesis author*

| | CoCit results | Score | S | A | CPA Results | Score | S | A | MLT Results | Score | S | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | United States | 2667 | | | Broadsheet | 386 | | X | Online newspaper | 1,32 | | X |
| 2 | Broadsheet | 1890 | | X | Tabloid (newspaper format) | 258 | | X | Community journalism | 1,25 | | X |
| 3 | Tabloid (newspaper format) | 1453 | | X | Magazine | 252 | | X | History of British newspapers | 1,21 | | X |
| 4 | English language | 955 | | | United States | 148 | | | Weekly newspaper | 1,19 | | X |
| 5 | News | 749 | | X | English language | 77 | | | Lebanon Daily News | 1,19 | | X |
| 6 | New York City | 739 | | | Publisher | 72 | | X | Decile of newspapers | 1,18 | | X |
| 7 | The New York Times | 637 | | X | Comic strip | 67 | | X | The Huntsville Times | 1,16 | | X |
| 8 | United Kingdom | 619 | | | Book | 58 | | X | Journalism and freedom | 1,15 | | X |
| 9 | Magazine | 610 | | X | Television | 57 | | X | Midland Daily News | 1,11 | | X |
| 10 | Australia | 551 | | | Radio | 54 | | X | The Laconia Daily Sun | 1,11 | | X |
| | | total | 0 | 5 | | total | 0 | 8 | | total | 0 | 10 |



**Table 5.7: Samples results for "Finance Act".**

**Article** Finance Act

**"See also" links** Finance Act 1999-2014, List of short titles

| | CoCit Results | Score | S | A | CPA Results | Score | S | A | MLT Results | Score | S | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Parliament of United Kingdom | 19 | | X | Parliament of United Kingdom | 0,83 | | X | Finance Act 2007 | 2,38 | X | X |
| 2 | Chancellor of Exchequer | 15 | | X | Chancellor of Exchequer | 0,61 | | X | Finance Act 2002 | 2,19 | X | X |
| 3 | HMRC | 13 | | X | Firearms Act | 0,57 | | X | Finance Act 2001 | 2,05 | X | X |
| 4 | New York Times | 12 | | | Gordon Brown | 0,53 | | X | Finance Act 2006 | 1,9 | X | X |
| 5 | Act of Parliament | 11 | | X | Taxation of Chargeable Gains Act 1992 | 0,48 | | X | Finance Act 2009 | 1,88 | X | X |
| 6 | Gordon Brown | 10 | | X | Fatal Accidents Act | 0,47 | | X | Finance Act 2000 | 1,87 | X | X |
| 7 | Hansard | 10 | | X | Capital Gains Tax | 0,42 | | X | Finance Act 2004 | 1,65 | X | X |
| 8 | Income Tax | 10 | | X | Income Tax | 0,41 | | X | Finance Act 2005 | 1,58 | X | X |
| 9 | Capital Gains Tax | 9 | | X | HMRC | 0,4 | | X | Finance Act 1999 | 1,57 | X | X |
| 10 | Corporation Tax | 9 | | X | Value Added Tax | 0,39 | | X | Finance Act 2003 | 1,54 | X | X |
| | | total | 0 | 9 | | total | 0 | 10 | | total | 10 | 10 |

**Table 5.9: Samples results for "Site map".**

**Article** Site map

**"See also" links** Biositemap, Contact page, Home page, Index (search engine), Link page, Search engine optimization, Sitemaps, Web indexing, XML

*S = Relevance by "See also" links*
*A = Relevance by thesis author*

| | CoCit Result | Score | S | A | CPA Results | Score | S | A | MLT Results | Score | S | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | HTML | 6 | | X | RSS | 0,67 | | X | Search engine submission | 2,51 | X | X |
| 2 | RSS | 6 | | X | Web crawler | 0,44 | | X | Caverio Map | 2,22 | | |
| 3 | Content management system | 5 | | X | Semantic web | 0,42 | | X | Image Map | 2,22 | | |
| 4 | Information retrieval | 5 | | X | Metadata | 0,42 | | X | Map Algebra | 2,22 | | |
| 5 | PHP | 5 | | | Text mining | 0,41 | | | Map Projection | 2,22 | | |
| 6 | BSD Licence | 4 | | | Lexical markup framework | 0,37 | | | Normal Map | 2,22 | | |
| 7 | Facebook | 4 | | | Web indexing | 0,34 | X | X | Road Atlas | 2,22 | | |
| 8 | Gerard Salton | 4 | | | Blog | 0,32 | | | Vinland Map | 2,22 | | |
| 9 | Internet | 4 | | | Full text search | 0,31 | | X | World Map | 2,22 | | |
| 10 | Lexical analysis | 4 | | | World wide web | 0,26 | | | Dymaxion Map | 2,22 | | |
| | | total | 0 | 4 | | total | 1 | 6 | | total | 1 | 1 |



**Table 5.10: Sample results for "Technical University of Berlin".**

**Article** Technical University of Berlin

**"See also" links** Berlin School of Economics and Law, Freie Universität Berlin, Hertie School of Governance, Humboldt Universität zu Berlin, Universität der Künste, Beuth Hochschule für Technik Berlin

| | CoCit Results | Score | S | A | CPA Results | Score | S | A | MLT Results | Score | S | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Germany | 375 | | | Germany | 13,82 | | | Anja Feldmann | 1,3 | | X |
| 2 | Berlin | 264 | | X | Berlin | 9,9 | | X | Braunschweig University of Technology | 1,1 | | X |
| 3 | World War II | 108 | | | Technical University of Munich | 5,3 | | X | Humboldt University of Berlin | 1,08 | (X) | X |
| 4 | Humboldt University of Berlin | 94 | (X) | X | Humboldt University of Berlin | 4,6 | (X) | X | Sebastian Möller | 1,05 | | X |
| 5 | German Empire | 71 | | | Charlottenburg | 3,5 | | X | List of architecture schools in Germany | 0,95 | | X |
| 6 | Technical University of Munich | 71 | | X | RWTH Aachen | 2,9 | | X | Berlin Mathematical School | 0,92 | | X |
| 7 | United States | 69 | | | Physics | 2,5 | | X | Graz University of Technology | 0,89 | | X |
| 8 | Hamburg | 54 | | | Habilitation | 2,4 | | X | BIG-SNE | 0,88 | | X |
| 9 | Physics | 52 | | X | World War II | 2,3 | | | Clausthal University of Technology | 0,84 | | X |
| 10 | University of Göttingen | 52 | | X | Free University of Berlin | 2,3 | (X) | X | Telekom Innovation Laboratories | 0,81 | | X |
| | | total | 0 | 5 | | total | 0 | 8 | | total | 0 | 10 |

## 5.5 Clickstream Evaluation

In the following, we shortly introduce an additional Wikipedia data source for a large-scale quasi-gold standard. The evaluation of a clickstream dataset partially contradicts the qualitative and quantitative evaluation. Based on the clickstream dataset, CPA is the best performing document similarity measure. Due to limitations in time, we are not able to investigate Wikipedia clickstreams in detail, as they were not part of the planned scope of the thesis.

The quantitative evaluation showed the significant advantage of MLT over CPA and CoCit in terms of MAP and the number of relevant documents retrieved, whereas the qualitative evaluation relativizes the performance differences. We see that the citation-based similarity measures have a small but stable error rate, because of highly co-cited articles (Section 6.1.2). MLT also retrieves for one of four queries mainly irrelevant documents (Section 5.3.3). Thus, we looked for evidence, which can explain this mismatch and proves that "See also" links do not overlap with our relevance judgment on a large scale.

Just after finishing the "See also"-based evaluation, we stumbled upon a Wikipedia clickstream dataset that was first released in February 2015 by WikiResearch [65]. The clickstream dataset was not available, when we had outlined the scope of this thesis. Therefore, we did not consider this dataset in the first place.

The clickstream dataset consists of aggregated referrer information for Wikipedia articles during the month of February 2015 (January 2015 is also available). Based on this data, we can determine the number of clicks on outbound links for available articles. For outbound links, which occur multiple times in an article, only the total number of clicks is provided. The clickstream dataset consists of data for 1,383,301 Wikipedia articles.

The number of clicks on a link can be considered as a judgement of relevance, because we assume that the more relevant a linked document is the more frequent its link gets clicked. Therefore, clickstream data can also be used as quasi-gold standard for evaluating document similarity measures. Instead of a binary relevance classification, which "See also" links enable, clickstreams allow a classification on a cardinal scale, i.e. the number of clicks per link. Thus, clickstreams provide a more distinct relevance judgement than "See also" links.

We implemented the clickstream-based evaluation in a similar manner as the "See also"-based evaluation (Section 4.2.6). Instead of testing if a retrieved document exists as a "See also" link, we assigned the number of clicks to all retrieved documents. By doing this, we generated a dataset, which consists of records containing the retrieved document, its rank and its click count, for each tested similarity measure.

Table 5.11 reports the total and average number of clicks on the documents retrieved by CoCit, CPA and MLT. Each row represents the values for the top-$k$-results of each similarity measure,

where $k \in \{10, 5, 1\}$. The results are based on 63,013 queries that were chosen according to the following criteria:

a) The article includes a "See also" section.

b) Clickstream data exists for the article in question.

c) All tested similarity measures can compute a similarity score for the article in question.

**Table 5.11: Evaluation of clickstream data from Februar 2015. Based on 63,013 queries.**

| | Avg. Clicks | | | Total Clicks | | |
|---|---|---|---|---|---|---|
| $k$ | CoCit | CPA | MLT | CoCit | CPA | MLT |
| 10 | 94.03 | 146.92 | 95.26 | 5,924,887 | 9,258,024 | 6,002,344 |
| 5 | 63.34 | 110.99 | 65.92 | 3,991,517 | 6,993,818 | 4,153,886 |
| 1 | 19.37 | 45.28 | 18.88 | 1,220,564 | 2,853,247 | 1,189,440 |

We clearly see a different outcome in the performance of the tested similarity measures. CPA has by far the greatest number of clicks, whereas CoCit and MLT have a similar click count. On average the first document retrieved by CPA is clicked 45.20 times. The first results of CoCit (19.37 clicks) and MLT (18.88 clicks) are clicked less than half as often. This clickstream-based ranking supports the results of the qualitative evaluation. Likewise, CPA performs not as badly as the quantitative evaluation implies.

Aside from the different performance ranking, this analysis also reveals that clicks are primarily made on the top result ($k = 1$). The top ten documents receive only 2.5 times more click than the first ranked document.

Yet, we cannot conclude that clickstream data provides a better relevance judgement than "See also" for the following reasons:

1. A simple comparison of total click numbers, which we did due to time constraints, does not consider all aspects of this dataset. In our analysis, we weighted all articles and clicks equally. We do not consider that popular articles with much traffic can generate much more clicks than niche articles. Thereby, popular articles have a higher impact on click numbers. In the same way, such a weighting is not necessarily bad, as it focuses on articles that are important to the users. Also, the number of outbound links is not reflected in this analysis. A link is probably less frequently clicked the more outbound links an article contains. Likewise, multiple outbound links to the same article increase the probability of clicks to the article.
2. The number of queries in this evaluation is much small than the queries used in the "See also"-based evaluation.
3. Clicks can only occur on links that exist in the article content. These in-content links are also included for navigational purposes, while "See also" are actual literature recommendations. The Wikipedia guideline explicitly suggests to only include links in

"See also" sections that do not exist in other parts of the article [46]. Therefore, we see conceptual differences of both quasi-gold standards for relevance judgment.

To summarise, the Wikipedia clickstream dataset allowed us to quantify the recommendation quality of the tested similarity measures in an additional promising evaluation. It proved our assumption of CPA's performance, which we derived from the qualitative evaluation. CPA's results are more frequently clicked than MLT's results and therefore CPA might provide the more relevant results. But a final decision requires a more detailed analysis. In other words, further research is needed to tap the full potential of the clickstream quasi-gold standard and to increase the significance of a clickstream-based evaluation.

# 6 Conclusions & Future Work

The goal of this thesis was to evaluate CoCit, CPA and MLT on the English version of Wikipedia. First, we investigated how suitable the citation-based similarity measures are to identify related Wikipedia articles. Second, we compared the performance in identifying related Wikipedia articles of CoCit and CPA to the baseline of the text-based MLT. Finally, we analysed the use of "See also" links as quasi-gold standard in a large-scale evaluation of document similarity measures.

## 6.1 Conclusions

The first and second research questions are answered by a quantitative and qualitative evaluation concerning the retrieval performance of the tested similarity measures. A satisfying recommendation quality of the citation-based document similarities measures would be equivalent to a successful extension of their application domain. Table 6.1 shows a side-by-side comparison of strengths and weaknesses of the text-based approach (MLT) to the citation-based approaches (CoCit, CPA).

**Table 6.1: Summary of the most important findings of the "See also"-based evaluation.**

| **Performance Evaluation** | |
|---|---|
| **MoreLikeThis** | **CPA / CoCit** |
| + Overall higher MAP score | + Shorter runtime, simple algorithm |
| + Works for most queries | + CPA performs better than CoCit |
| ± Retrieves close related topics | + Independent from synonyms and languages |
| ± Works best with medium length articles | ± CPA equals CoCit at $\alpha = 0$ |
| - Fails to recognise synonyms | ± Retrieve broader topical spectrum |
| - Long runtime, complex algorithm | - Retrieves irrelevant but highly co-cited articles |
| - Language dependent | - Requires inbound links |

From our quantitative and qualitative evaluation, we draw the following conclusions:

- Based on the quantitative "See also" evaluation, MLT is the best performing similarity measure for identifying related Wikipedia articles, followed by second-ranked CPA and third-ranked CoCit.
- The quantitative evaluation did not reveal a single subset of the evaluation corpus, where MLT was not the best performing similarity measure.
- The qualitative evaluation did not reflect the quantitative performance ranking. Recommendation quality of citation-based measures appeared not as low as in the quantitative evaluation. In particular, CPA performed similar to MLT.
- The clickstream evaluation supports the qualitative results: CPA's results received the most clicks.

### 6.1.1 MoreLikeThis

MLT provided the best performance in terms of the quantitative "See also"-based evaluations. Only one of four sample of the qualitative evaluation resulted in a low performance. MLT typically retrieved documents that were topical neighbours, e.g. entities, when a general topic was queried, or episode for a series. For instance, queries for "Newspaper" or "Finance Act" resulted in actual newspaper publications, e.g. "Lebanon Daily News", or Finance Acts of a certain year. These retrieved documents seemed to be convenient for users, who have some prior knowledge of the topic and are looking for examples or more detailed information. MLT also showed an advantage over CPA and CoCit regarding the amount of retrieved documents (Table 5.1). MLT retrieved documents for queries, which could not be served by the citation-based similarity measures. Those queries belonged to article without any inbound links. In addition, MLT performed surprisingly well on short articles. Articles with a length of 207-310 words made MLT perform best, whereas MLT's performance dropped when the number of words increased. We pointed out that the reasons for this behaviour is due to subtopics in long articles, which complicate determination of semantic similarity for text-based approaches (Section 5.2.2.1).

However, concluding that MLT is the best document similarity measure in any case would be wrong. The example "Site map" showed a well-known problem of text-based similarity measures: Lexical similarity does not necessarily implicate semantic similarity (Section 2.1.1). Therefore, MLT retrieved documents related to the term "Map", which were not in the same context as "Site map". In contrast, CPA and CoCit were able to retrieve relevant documents at this sample.

### 6.1.2 CPA & CoCit

Although, the quantitative evaluation showed significant disadvantage of the citation-based similarity measures in comparison with MLT, we cannot state that CPA and CoCit generally perform worse than MLT, since the qualitative and clickstream evaluation provided a different outcome, especially regarding CPA.

In none of the samples in the qualitative evaluation, the citation-based similarity measures retrieved only irrelevant document. Mostly, their results were relevant but not captured by the gold standard. Therefore, we assumed that "See also" links do not truly correspond to our relevance judgment. The clickstream evaluation quantified this assumption. Documents retrieved by CPA were more frequently clicked than MLT's results. Based on this findings, we conclude that CoCit and especially CPA are suitable for identify related articles in Wikipedia.

The experiments showed that considering co-citation proximity improves Co-Citation scores. In other words, CPA performs better than CoCit. Both methods tended to retrieve a broader spectrum of related topics, not just topical neighbours. CoCit showed a disadvantage compared to CPA by retrieving documents that were too distantly related to be relevant. For example,

"Germany" is technically speaking related to "Technical University of Berlin", because it is the country, where the university is located. Nonetheless, the article "Germany" is probably irrelevant to the user. On the other hand, retrieved documents like "Radio" or "Book" for the query "Newspaper" are probably relevant, even if they are not directly related. Thus, we recommend using CPA and CoCit as document similarity measures if the information need of the user is to get a first broad overview of a topic.

Also, we saw a tendency to retrieve topically more distantly related documents, when the number of inbound links increased. Especially geopolitical-related articles were retrieved for articles with high co-citation count. For instance, many company-centric articles include links to general articles of the company's field as well as to company's city or country. Thus, "United States" is the top co-cited article of "Newspaper", because there are many Wikipedia articles about newspapers from the United States linking to both articles. Bellomi and Bonato already showed this dominating role of geopolitical topics and common words in the Wikipedia link network [42]. These often co-cited articles are similar to the words in the concept of VSM and TF-IDF. They often occur in a text, but have little topical meaning, because they also occur very frequently in the whole corpus.

Furthermore, other topical irrelevant links can often be found in the information box, which is located next to the text within Wikipedia articles. Information boxes include commonly links to topical irrelevant articles like "Nicknames" in "Technical University of Berlin". Also, the section "External links" contains many irrelevant links. For example, Wikipedia articles of musicians and actors often have external links to the artists' profiles on websites like *Internet Movie Database*[9] or *Discogs*[10]. Besides the actual link to each profile, many articles also include a link to the Wikipedia article of those websites. As a result, the articles of *Internet Movie Database* or *Discogs* are often retrieved, when querying music- or movie-related topics. The equivalent of these links in MLT are stop words, i.e. words that are irrelevant and are getting removed from the document text, before MLT indexes documents.

### 6.1.3 Summary

Summing up, MLT performed better than CPA and CoCit in our "See also"-based evaluation. While MLT had a significant advantage from the quantitative evaluation, our sample analysis showed that MLT and CPA are at least equivalent, when we judged document relevance. The clickstream-based evaluation revealed also a shift in rankings towards the advantage of CPA. However, the clickstream analysis was too superficial to give a well-grounded argument for a different in performance ranking of the similarity measures.

[9] http://www.imdb.com/
[10] http://www.discogs.com/

Despite overall performance ranking, the qualitative evaluation proved that the appropriateness of a similarity measure eventually depends on the individual information need. For an early-stage literature research or other use cases that require retrieving a broader spectrum of related topics, we recommend using CPA as document similarity measure, based on the findings of the qualitative evaluation. Otherwise, MLT should be the document similarity measure of choice. Furthermore, we showed that CoCit is a special case of CPA, whereby our optimised CPA approach outperforms CoCit (Section 5.1).

Regardless of the performance disadvantage of CPA towards MLT, we see CPA as a concept with high future potential. MLT is the result of substantial research efforts. The concepts of VSM and TF-IDF evolved over decades. Lucene's implementation of the MLT algorithm is complex and highly optimised. Thus, MLT has a much longer runtime than CPA (Table 4.1). In contrast, CPA has a relatively short history. CPA's algorithm is quite simple and offers potential for optimisation. Moreover, CPA works language independent, whereas MLT requires language dependent operations like stop word removal and stemming.

### 6.1.4 "See also" Links

The question remains whether "See also" links are an appropriate quasi-gold standard for topically related articles in a large-scale evaluation of document similarity measures. By using "See also" links as quasi-gold standard, we were able to evaluate three document similarity measures with more than 600.000 queries and to optimise our CPI model. Thus, we could show tendencies found on different article properties and determine the best performing similarity measure for identifying topically related Wikipedia articles.

However, we still needed the qualitative evaluation in form of a small-scale manual user study to reveal certain characteristics of tested similarity measures. The results of the qualitative evaluation contradicted those of quantitative evaluation. Thus, we ended up validating the "See also"-based results with the clickstream dataset, i.e. another quasi-gold standard.

We pointed out in Section 3.2 that only 17% of all Wikipedia articles included a "See also" section. Those sections contained only a low number of links (2.6) on average. Approximately 2 million "See also" links have been extracted. Nonetheless, the majority of queries did not result in any document marked as relevant in our gold standard, even though we judged manually most of the sample results as relevant, i.e. "See also" links had a high false negative error rate. The "See also" quasi-gold standard did not contain many relevant documents for different reasons. Lack of uniform link names had some effect (see "Technical University of Berlin"), but mainly the low number of "See also" links, compared to the high number of relevant Wikipedia articles, resulted in the missing relevance judgments. For instance, the article "Newspaper" consists of only two "See also" links. Yet, there are probably more than hundred relevant articles in Wikipedia. Therefore, the probability of retrieving those two articles is low, especially, when

relying on a binary classification of relevance. The "See also" link to the article "List of newspaper comic strips" might be relevant, but not as relevant as "List of newspapers". In this context, relevance is better expressed as a scalar value that ranges between highly relevant and irrelevant. Therefore, "See also" links have disadvantage for evaluating document similarity measures.

A judgment about "See also" links also requires a comparison to other quasi-gold standards. For instance, a user study, which asks experts to name a limited number of relevant documents, will always miss out on relevant articles. Probably, such a user study would also lead to a binary relevance classification, when the experts name only highly relevant documents. Consequently, this expert quasi-gold standard would have – more or less – the same problems as "See also" links without having the advantage of a large-scale evaluation. Only a retrospective user study, which asks experts to rate result sets, would allow a complete and scalar relevance judgment. Thus, we also performed a qualitative evaluation to enhance the insights from the "See also" link evaluation. However, such a user study is not likely to evaluate 600.000 queries.

Consequently, a user study has not necessarily a quality advantage over "See also" links as relevance judgement. But "See also" links are beneficial in terms of scale, as at the scale of "See also" links the law of large numbers apply. Accordingly, the amount of relevance misjudgements is negligible, when the number of queries is at this high level. Therefore, Wikipedia's "See also" sections can indeed serve as quasi-gold standard, which allows performing automated large-scale evaluations of document similarity measures.

### 6.1.5 Clickstream Dataset

Nonetheless, we used the clickstream dataset as additional quasi-gold standard. It is interesting data source and differs in many aspects to "See also" links. The primary advantages of clickstream over "See also" links are:

a) Clickstreams provide a scalar relevance judgment: The more clicks are counted, the more relevant a retrieved document is. "See also" links can only give a binary relevance classification.
b) Clickstreams are theoretical available for all Wikipedia articles, whereas "See also" links are limited to articles that contain a "See also" section. Even if clickstream data is currently only available for January and February 2015, i.e. 1.3 million articles are covered by clickstream data. In the future WikiResearch will probably release more datasets.
c) Clickstreams reflect the judgments of many Wikipedia users, whereas only the selected group of Wikipedia authors creates "See also" links. A bigger group of users provides a better-balanced relevance judgement.

On the hand, clickstreams bring up several disadvantages:

a) Clickstreams can only determine relevance of a retrieved document, when a link to that document exists in the article. The article in question might not link to documents from broader topical spectrums, which are also important as literature recommendation. Clickstreams cannot cover these documents.
b) The weighting of clicks is not clear yet. More research is needed to find an appropriated click weighting to reflect different popularity and outbound links of articles.
c) Clickstreams can also include bot-generated clicks. Even if the authors of the dataset attempted to remove bot clicks, bot-generated clicks may still affect the results [66].

Due to these aspects and the lack of research on Wikipedia's clickstreams, we cannot state that one of the quasi-gold standards is more suitable for this experiment. Both "See also" links and clickstreams seem suitable to overcome the limitation of traditional user studies and allow automated large-scale evaluations of document similarity measures.

## 6.2 Future Work

In the following we briefly discuss potential areas for future work. We begin with work regarding Apache Flink, continue with implementation-related issues and improvements of the CPA concept. Lastly, we close the thesis with gold standard specific future work.

### 6.2.1 Apache Flink

As a first future work, we propose to add a functional sorting operator to Apache Flink, as we could not use the currently built in sorting operator, because it was not working correctly (Section 4.2.6). Furthermore, it would be useful for Flink to natively provide a hash-based GroupBy operator for multiple fields, for performance reasons we needed to implement this functionality by ourselves (Section 4.2.3.3).

### 6.2.2 Implementation

Second, we propose to eliminate error sources of the evaluation process by analysing redirects and devaluing outer-text links. The experiment revealed that Wikipedia links are used inconsistently. Two articles A and B may refer to the same article C, but the links from A and B were not recognised as having the same link target, when, for instance, A pointed through a redirection to C. Besides CoCit and CPA, the "See also" links were also affected by this issue (Section 5.3.4). Hence, we propose to analyse all internal Wikipedia links to check, if they contain a redirection. Then, such redirections should be mapped to their link target, to prevent these redirection issues.

### 6.2.3 CPA

Moreover, we showed that outer-text links, which do not belong to the actual article text, are a source of irrelevant CoCit and CPA results, because these links, e.g. information boxes or article footer, are usually unrelated to the article topic. Instead, outer-text links often refer to general topics or location-related articles. Devaluing or ignoring these links should improve the performance of the citation-based similarity measures. This procedure corresponds to the stop word removal or TF-IDF weighting in MLT.

Another area of future work, which we propose, is modifying the CPA concept to improve its performance in identify related articles. We defined proximity as the number of words between two links and calculated the CPI in a negative exponential formula (Section 4.2.3.2). Other weighting approaches might better meet the conditions of our test collection. For instance, proximity should also correlate with total article length or the number of inbound links an article receives. Both factors should be reflected in the CPI formula. This change can also be found in the concept of TF-IDF, it can be used to devalue citation of general articles, which are frequently co-cited but have no topically meaning, e.g. geopolitical entities.

Changing the definition of proximity from a structural to a semantic measure could also enhance CPA's performance. Methods of Natural Language Processing need to determine the semantic relationship of co-citations within an article as the semantic relationship of a co-citation seem to be more important than its actual proximity within the text.

The following examples illustrate the concept:

a) Topic X is covered by paper A. In contrast, paper B covers topic Y.

b) Paper A covers topic X. Also, paper B covers topic X.

For instance, the semantic meaning of co-citation relationship can be contradictory, when the sentences or paragraphs express opposing meanings. Example a) shows that the two papers A and B can cover different topics, even if their co-citation are in close proximity. On the other hand, example b) proves the opposite.

Consequently, the semantic connection of sentences (e.g. "In contrast" / "Also") as well affects semantic proximity of co-citations. Based on this semantic relationship, a new definition of co-citation proximity can be established to increase effectiveness of CPA.

Furthermore, a hybrid-CPA is possible. As Gipp already proposed [27], CPA can be combined with other similarity measures, e.g. MLT. Depending of the properties of an article, one similarity measure performs better than the other. For example, a hybrid-CPA can use MLT for article with a few inbound links to be independent from the number of links. Thereby, the hybrid system benefits from the advantages of each similarity measure.

### 6.2.4 Gold Standards

A major field of future work regards the tested quasi-gold standards. We suggest two options regarding the "See also" quasi-gold standard. First, the experiment showed problems like a shortage of "See also" sections as well as a low number of links per section. Creating "See also" links is not in our hands. Yet, we can motivate authors in the Wikipedia community to increase the usage of "See also" links. If the Wikipedia guideline promotes the "See also" section, probably more links will be added and therefore the false negative error rate could drop. As a result, the value of "See also" links as quasi-gold standard would increase.

Second, we propose to use the openness of Wikipedia for new evaluation approaches. Instead of comparing the results of the document similarity measures with the "See also" links, we can create a Wikipedia bot [67], which writes the results automatically to the "See also" sections. Then, the Wikipedia authors revise the links for relevance similar to a crowdsourcing campaign. Afterwards, an automatic evaluation can test if the links had been removed or not. Links, which had not been removed, can be judged as relevant. The usage of a Wikipedia bot goes along with the Wikipedia guideline. However, such a crowdsourcing evaluation requires a long time span, since it might take weeks or months until all results are revised.

As we outline in Section 5.5, the clickstream quasi-gold standard needs further research. The dataset just recently became available. Thus, no research had been done yet. Also, we evaluated the clickstreams only superficially.

At first, the quantitative clickstream evaluation needs to be compared to a manual qualitative evaluation to prove its validity as relevance judgment for the task of identify related Wikipedia articles. The number of clicks may better indicate the position of a link within the article than the links relevance. A link in the first paragraph is probably more frequently clicked than a link in the article end, since many users do not read the whole article. Other factors like total number of outbound links or article popularity also need to be investigated as they may have a strong impact on the distribution of clicks. Moreover, it is questionable if many Wikipedia users can give better literature recommendation than the authors. Furthermore, more work needs to be done for filtering bot-generated clicks as authors of the dataset suggest [66].

Moreover, we propose to investigate generally the clickstream quasi-gold standard in similar manner to our "See also"-based study. A detailed evaluation of both, "See also" links and clickstreams, would allow a better understanding of the advantages and disadvantages of each quasi-gold standard.

# A Appendix

## A.1 Bibliography

[1] M. Ware and M. Mabe, "The stm report," *Int. Assoc. Sci. Tech. Med. Publ.*, no. November, 2009.

[2] R. Lachica, D. Karabeg, and S. Rudan, "Quality, Relevance and Importance in Information Retrieval with Fuzzy Semantic Networks," *Proc. TMRA*, 2008.

[3] J. Lin and W. J. Wilbur, "PubMed related articles: a probabilistic topic-based model for content similarity.," *BMC Bioinformatics*, vol. 8, p. 423, Jan. 2007.

[4] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," *Proc. 21st Natl. Conf. Artif. Intell.*, vol. 1, pp. 775–780, 2006.

[5] H. Small, "A New Measure of the Relationship Two Documents," vol. 24, no. 4, pp. 28–31, 1973.

[6] I. Marshakova, "System of document connections based on references," *Sci. Tech. Inf. Ser. VINITI*, vol. 6, no. 2, pp. 3–8, 1973.

[7] M. Cristo, E. S. De Moura, and N. Ziviani, "Link Information as a Similarity Measure in Web Classification," *Lect. Notes Comput. Sci.*, vol. 2857, no. String Processing and Information Retrieval, pp. 43–55, 2003.

[8] B. Gipp and J. Beel, "Citation Proximity Analysis (CPA)-A new approach for identifying related work based on Co-Citation Analysis," *Birger Larsen Jacqueline Leta, Ed. Proc. 12th Int. Conf. Sci. Inf.*, vol. 2, no. July, pp. 571–575, 2009.

[9] S. Liu and C. Chen, "The effects of co-citation proximity on co-citation analysis," *Proc. ISSI*, 2011.

[10] Pew Internet & American Life Project, "How students use Wikipedia," *Director*, 2007. [Online]. Available: http://www.pewinternet.org/2007/04/24/wikipedia-users/. [Accessed: 10-Apr-2015].

[11] NIST, "Text REtrieval Conference (TREC) Test Collections." [Online]. Available: http://trec.nist.gov/data/test_coll.html. [Accessed: 05-May-2015].

[12] M. Busch, "Twitter's New Search Architecture," 2010. [Online]. Available: https://blog.twitter.com/2010/twitters-new-search-architecture. [Accessed: 23-May-2015].

[13] D. Cohen, E. Amitay, and D. Carmel, "Lucene and Juru at Trec 2007 : 1-Million Queries Track," in *TREC 2007*, 2007, pp. 0–6.

[14] M. Konchady, *Building Search Applications: Lucene, Lingpipe, and Gate*. 2008.

[15] N. Rubens, "The Application of Fuzzy Logic to the Construction of the Ranking Function of Information Retrieval Systems," vol. 10, no. 1, pp. 20–27, 2006.

[16] D. Lin, "An Information-Theoretic Definition of Similarity," *Proc. ICML*, pp. 296–304, 1998.

[17] J. B. Lovins, "Development of a stemming algorithm," *Mech. Transl. Comput. Linguist.*, vol. 11, no. June, pp. 22–31, 1968.

[18] C. Fox, "A stop list for general text," *ACM SIGIR Forum*, vol. 24, no. 1–2. pp. 19–21, 1989.

[19] C. Shin and D. Doermann, "Document image retrieval based on layout structural similarity," *Int. Conf. Image Process. Comput. Vision, Pattern Recognit.*, pp. 606–612, 2006.

[20] D. Buttler, "A short survey of document structure similarity algorithms," *Int. Conf. Internet Comput.*, pp. 3–9, 2004.

[21] C. Fellbaum, "WordNet," in *Theory and Applications of Ontology: Computer Applications*, 2010, pp. 231–243.

[22] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Review*, vol. 37, no. 4. pp. 573–595, 1995.

[23] G. Salton, A. Wong, and C. Yang, "A Vector Space Model for Automatic Indexing," *Communications*, vol. 18, no. 11, 1975.

[24] K. S. Jones, "Index term weighting," *Information Storage and Retrieval*, vol. 9, no. 11. pp. 619–633, 1973.

[25] C. D. Manning and P. Raghavan, "An Introduction to Information Retrieval," 2009.

[26] Johnson, "How MoreLikeThis Works in Lucene," *Blog*, 2008. [Online]. Available: http://cephas.net/blog/2008/03/30/how-morelikethis-works-in-lucene/.

[27] B. Gipp, "Citation-based Plagiarism Detection: Applying Citation Pattern Analysis to Identify Currently Non- Machine-Detectable Disguised Plagiarism in Scientific Publications," *Springer Vieweg Research*, 2014. .

[28] L. C. Smith, "Citation Analysis," *Libr. Trends*, vol. 30, pp. 83–106, 1981.

[29] B. Larsen, *References and Citations in Automatic Indexing and Retrieval Systems: Experiments with the Boomerang Effect.* 2004.

[30] Egghe and Rousseau, "Introduction to Informetrics - quantitative methods in library, documentation and information science." p. 204, 1990.

[31] M. Eto, "Evaluations of context-based co-citation searching," *Scientometrics*, vol. 94, no. 2, pp. 651–673, 2013.

[32] E. Hetzner, "A simple method for citation metadata extraction using hidden markov models," *Jcdl*, pp. 280–284, 2008.

[33] M. Thelwall and D. Wilkinson, "Finding similar academic Web sites with links, bibliometric couplings and colinks," *Inf. Process. Manag.*, vol. 40, no. 3, pp. 515–526, 2004.

[34] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking," *World Wide Web Internet Web Inf. Syst.*, vol. 54, no. 1999–66, pp. 1–17, 1998.

[35] Wikipedia, "Wikipedia:Manual of Style/Linking," *Wikipedia*, 2014. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#General_points_on_linking_style. [Accessed: 20-Apr-2015].

[36] M. Kessler, "Bibliographic coupling between scientific papers," *Am. Doc.*, vol. 97, no. January, 1963.

[37] J. Martyn, "Bibliographic Coupling," *Journal of Documentation*, vol. 20, no. 4. pp. 236–236, 1964.

[38] E. Garfield, "From Bibliographic Coupling to Co-Citation Analysis via Algorithmic Histro-Bibliography," *Current*, 2001.

[39] N. Tran, P. Alves, S. Ma, and M. Krauthammer, "Enriching PubMed Related Article Search with Sentence Level," pp. 650–654, 2009.

[40] US National Library of Medicine, "PubMed." [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed. [Accessed: 30-Apr-2015].

[41] Y. Ollivier and P. Senellart, "Finding Related Pages Using Green Measures : An Illustration with Wikipedia," *Proc. AAAI*, pp. 1427–1433, 2007.

[42] F. Bellomi and R. Bonato, "Network Analysis for Wikipedia," *Proc. Wikimania*, p. 81, 2005.

[43] Alexa, "Wikipedia Alexa details." [Online]. Available: http://www.alexa.com/siteinfo/wikipedia.org. [Accessed: 31-Mar-2015].

[44] Wikimedia, "Wikipedia Page View Stats." [Online]. Available: http://stats.wikimedia.org/EN/TablesPageViewsMonthlyCombined.htm. [Accessed: 31-Mar-2015].

[45] Wikipedia, "Wiki markup," *Wikipedia*, 2015. [Online]. Available: http://en.wikipedia.org/wiki/Wiki_markup. [Accessed: 10-Apr-2015].

[46] Wikipedia, "Wikipedia:Manual of Style/Layout," *Wikipedia*, 2014. [Online]. Available: https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Layout#See_also_section. [Accessed: 15-Apr-2015].

[47] D. Milne, "Computing Semantic Relatedness using Wikipedia Link Structure," in *Work*, 2007, vol. 7, p. 8.

[48] P. Petrescu, M. Dr. Ghita, and D. Loiz, "Google Organic CTR Study 2014," 2014.

[49] G. V. Cormack and T. R. Lynam, "Validity and power of t-test for comparing MAP and GMAP," *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '07*, p. 753, 2007.

[50] S. Robertson, "On GMAP," *Proc. 15th ACM Int. Conf. Inf. Knowl. Manag. - CIKM '06*, p. 78, 2006.

[51] J. S. Ward and A. Barker, "Undefined By Data: A Survey of Big Data Definitions," *arXiv.org*, 2013.

[52] M. A. Beyer and D. Laney, "The Importance of 'Big Data': A Definition," *Gartner Publications*, pp. 1–9, 2012.

[53] V. N. Gudivada, D. Rao, and V. V. Raghavan, "NoSQL Systems for Big Data Management," *2014 IEEE World Congr. Serv.*, pp. 190–197, 2014.

[54] J. Dean and S. Ghemawat, "MapReduce : Simplified Data Processing on Large Clusters," *Commun. ACM*, vol. 51, no. 1, pp. 1–13, 2008.

[55] Apache, "Apache Hadoop Website." [Online]. Available: http://hadoop.apache.org. [Accessed: 30-Apr-2015].

[56] T. White, *Hadoop: The definitive guide*, vol. 54. 2012.

[57] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke, "The Stratosphere platform for big data analytics," *VLDB J.*, pp. 939–964, 2014.

[58] Apache, "Apache Flink Website." [Online]. Available: http://flink.incubator.apache.org/.

[59] S. Ewen, "Apache Flink Overview at Stockholm Hadoop User Group," 2014. [Online]. Available: http://www.slideshare.net/stephanewen1/apache-flink-overview-at-stockholm-hadoop-user-group. [Accessed: 30-Apr-2015].

[60] Apache, "Apache Lucene Website." [Online]. Available: http://lucene.apache.org/. [Accessed: 30-Apr-2015].

[61] E. Hatcher and O. Gospodnetic, *Lucene in Action*, vol. 54. 2005.

[62] Apache, "Apace Lucene - Similarity Class." [Online]. Available: https://lucene.apache.org/core/2_9_4/api/core/org/apache/lucene/search/Similarity.html. [Accessed: 30-Apr-2015].

[63] Elastic, "Elasticsearch Website." [Online]. Available: https://www.elastic.co/products/elasticsearch. [Accessed: 02-May-2015].

[64] O. Kononenko, O. Baysal, R. Holmes, and M. W. Godfrey, "Mining modern repositories with elasticsearch," *Proc. 11th Work. Conf. Min. Softw. Repos. - MSR 2014*, pp. 328–331, 2014.

[65] Wikimedia, "Research: Wikipedia clickstream," *Wikimedia*, 2015. [Online]. Available: http://meta.wikimedia.org/wiki/Research:Wikipedia_clickstream. [Accessed: 27-May-2015].

[66] E. Wulczyn, "Wikipedia Clickstream: Getting Started." [Online]. Available: http://ewulczyn.github.io/Wikipedia_Clickstream_Getting_Started/. [Accessed: 25-May-2015].

[67] Wikipedia, "Wikipedia:Bots," *Wikipedia*. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Bots. [Accessed: 05-Jun-2015].

## A.2 List of Abbreviations

| | |
|---|---|
| **ACID** | Atomicity, Consistency, Isolation, Durability |
| **CoCit** | Co-Citation |
| **CPA** | Co-Citation Proximity Analysis |
| **CPI** | Co-Citation Proximity Index |
| **CSV** | Comma-separated Values |
| **e.g.** | exempli gratia |
| **HDFS** | Hadoop Distributed File System |
| **i.e.** | id est |
| **I/O** | Input / Output |
| **IR** | Information Retrieval |
| **MAP** | Mean Average Precision |
| **MLT** | MoreLikeThis |
| **NoSQL** | Not Only SQL (Structured Query Language) |
| **RDBMS** | Rational Database Management System |
| **TF-IDF** | Term Frequency – Inverse Document Frequency |
| **VSM** | Vector Space Model |
| **XML** | Extensible Markup Language |

## A.3 List of Figures

## A.4 List of Tables