Master thesis - University of Groningen
Faculty of Mathematics and Natural Sciences
February 2013

Semantic relatedness using web data

The comparison of concepts using search results

FERNAND GEERTSEMA

SEMANTIC RELATEDNESS USING WEB DATA

THE COMPARISON OF CONCEPTS USING SEARCH RESULTS

FERNAND GEERTSEMA

Master thesis
Computing Science – Faculty of Mathematics and Natural Sciences
University of Groningen

February 2013 – version 1.2

Fernand Geertsema: *Semantic relatedness using web data,* The comparison of concepts using search results, February 2013

SUPERVISORS:
Mathijs Homminga
Alexander Lazovik
Michael Wilkinson

ABSTRACT

Investigating the existence of relations between people is the starting point of this research. Previous scientific research focussed on relations between general concepts in lexical databases. Web data was only part of the periphery of scientific research. Due to the important role of web data in determining relations between people further research into relatedness between general concepts in web data is needed.

For handling the different contexts of general concepts in web data for calculating semantic relatedness three different algorithms are used. The Normalized Compression Distance searches for overlapping pieces of text in web pages to calculate semantic relatedness. The Jaccard index on keywords uses text annotation to find keywords in texts and uses these keywords to calculate an overlap between them. The Normalized Web Distance uses the co-occurrence of concepts to calculate their semantic relatedness.

These approaches are tested with the use of the WordSimilarity-353 test collection. This dataset consists of 353 different concepts pairs with a human assigned relatedness score. The concepts in this collection are the input for gathering web pages from Google, Wikipedia and IMDb. Variables that influence the results of the algorithms are the number of web pages, the type of content and algorithm specific variables like the used compressor and weight factors.

The results are analysed on accuracy, robustness and performance. The results show that the context of concepts can be used in different ways to calculate semantic relatedness. The Normalized Compression Distance achieves higher scores than the Jaccard index on the general web data from Google and Wikipedia. Even though this score is influenced by writing styles on web pages. The better performance of the Normalized Compression Distance and the higher scores on general web data make it a good candidate for applications with automated semantic relatedness calculations. To achieve better scores further research into better compressors and cleaning of input data will improve the accuracy of this algorithm and decrease the sensitivity to writing style. For applications that provide exploratory insights in semantic relatedness, the Jaccard index on keywords is advised.

CONTENTS

1

INTRODUCTION

Do Barack Obama and Albert Einstein have anything in common? A quick search on the Internet shows that Einstein was born in 1879 and died in 1955. Six years before Obama was born. Einstein was born in Ulm, Germany and Obama in Hawaii, United States. They studied different subjects. Einstein followed mathematics and physics at Eidgenössische Technische Hochschule Zürich in Switzerland and Obama law at Harvard Law School in the USA. At first glance they have nothing in common. Further investigation shows their relatedness in the fact that they both received a Nobel prize. Obama received his Nobel prize for Peace in 2009. Einstein received the Nobel prize for Physics in 1921.

Investigating the existence of relations between people is the starting point of this research. It is easy to gain information about people in the public eye. Their names appear in Wikipedia, gossip columns and news articles all over the Internet. Since the introduction of social media, web data contains more and more information about everyone. For instance their online profile, curriculum vitae and status messages. How can this information be used to identify relations between persons?

Previous scientific research focussed on relations between general concepts in lexical databases. Web data was only part of the periphery of scientific research. Due to the important role of web data in determining relations between people further research into relatedness between general concepts in web data is needed. This thesis, semantic relatedness using web data, aims to bridge the gap between research into relatedness of general concepts and relatedness of persons by researching the relatedness of general concepts in web data.

Concepts, semantic relatedness and web data are the three main elements of this research and will therefore be explained in the following three paragraphs.

1.1 CONCEPTS

> *Concept (noun \'kän-.sept\ - source Merriam-Webster)*
>
> 1. *something conceived in the mind: thought, notion*
> 2. *an abstract or generic idea generalized from particular instances*

Human beings apply the second part of the definition to objects they see to simplify communication. They assign attributes to con-

cepts (e.g. animals, trades, food) to enable easier recognition and comparison. Examples of these comparisons are: What do the concepts tiger and jaguar have in common? They live in the wild, are carnivores, have sharp claws, vicious teeth and a patterned fur. What do the concepts Bentley and Jaguar have in common? They both have wheels, mirrors, a chassis and an engine. These two examples show that the outcome of the comparison depends on the context of the concepts.

1.2 SEMANTIC RELATEDNESS

Semantics is a branch of linguistics and logic concerned with meaning [1]. Semantics is used to examine the degree of relatedness between concepts in different contexts. The degree of relatedness is divided in three specific measurements by Budanitsky & Hirst [2]. They propose the following semantic measurements: semantic relatedness, semantic similarity and semantic distance.

Table 1: Semantic measurements and their relations.

Type of relation	Example	Related	Similar	Distance
Meronymy ("has part")	hand - finger	√		√
Holonymy ("is part of")	room - house	√		√
Antonymy ("opposite of")	hot - cold	√		
Functional ("using")	car - gasoline	√		√
Synonym ("same meaning")	car - automobile	√	√	√
Hyponyms ("more generic")	car - motor vehicle	√	√	√
Similar classification	car - bicycle	√	√	√

Semantic relatedness is a general measurement that includes all kinds of relations. It includes meronymy (“has part”), holonymy (“is part of”), antonymy (“opposite of”) and functional (“using”) relations. With these generic relations the number of potential relations is infinitive.

Semantic similarity contains synonyms (“same meaning”), hyponyms (“more generic”) and concepts with a similar classification. Budanitsky [3] views semantic similarity as a small selection of semantic relatedness . Resnik [4] demonstrates the difference between semantic relatedness and semantic similarity with an example of a car and

gasoline. *Semantic similarity represents a special case of semantic relatedness: for example, cars and gasoline would seem to be more closely related than, say, cars and bicycles, but the latter pair are certainly more similar.*

Semantic distance is related to semantic relatedness and semantic similarity. When concepts are semantically similar or related there distance is close and when unsimilar or unrelated their distance is large. For antonyms the opposite is valid. E.g. the concepts hot and cold are semantically related but their distance is far apart.

These semantic measurements are shown with some examples in table 1.

1.3 WEB DATA

Previous research on estimating the degree of semantic relatedness between concepts was focussed on lexical databases. These lexical databases [5][6] provided the basis for researchers to calculate the degree of semantic relatedness between concepts with machines [2]. A disadvantage of these lexical databases is the manual labour needed to create and revise them.

More recent approaches [7][8] use Wikipedia as database. The articles on Wikipedia are used to represent the concepts. Advantages of this database are the high number of articles and the multiple languages. The relatedness of concepts is calculated with the use of the article categories or the incoming and outgoing links of articles.

The size of the used dataset limits the scope in these approaches. If no Wikipedia article exists for a certain concept, its relatedness to other concepts is unknown. To broaden the scope web data can be used. The web data contains business information, personal information, encyclopaedic information etc. The difficulty of using web data to represent concepts lies in heterogeneity of the web. Web pages use different text formats, structures and different writing styles. The structure of web pages focusses on displaying, which weakens their semantic structure. The use of heterogeneous web data for examining relatedness of concepts will increase the size of the dataset and could therefore increase the number of comparisons.

1.4 STRUCTURE OF THE THESIS

This thesis has the following structure. First the research question is formulated (chapter 2), followed by an introduction of related work (chapter 3). The algorithms (chapter 4) used in the setup of the research (chapter 5) are discussed. These algorithms are tested on three different datasets (chapter 6). With the results (chapter 7) for this test data the research question is answered (chapter 8).

2

RESEARCH QUESTION

In this research web data is used to represent concepts. These web pages demonstrate the different contexts of these concepts. E.g. a search on Google for the concept “tiger” shows links to web pages about the animal, the Asian airline, the baseball team and the golf player. To represent these different contexts of concepts multiple web pages have to be used during the comparison, e.g. the fifty web pages about the tiger are compared to fifty web pages about the jaguar. This use of the context of the concepts leads to the following research question:

> "How can the context of concepts in web data be used to calculate semantic relatedness?"

An universal similarity measurement is the Normalized Compression Distance [9]. This measurement calculates the similarity between two sets of data, e.g. the similarity of books. Previous research focussed on the Normalized Web Distance (par. 4.4) to calculate relatedness of concepts in web data. This new approach uses Normalized Compression Distance and leads to the first sub question:

1. "What is the added value of Normalized Compression Distance on calculating semantic relatedness?"

The web pages representing concepts may have similar properties. These properties can be indicators for the relatedness of concepts. One of the approaches to extract these properties from web pages is keyword extraction. This method extracts a list of keywords from a given text. This list acts as a glossary for the content of a web page.

2. "What is the added value of keyword extraction on calculating semantic relatedness?"

3 RELATED WORK

In the field of semantic relatedness various benchmarks are available. These benchmarks made it possible for research to develop and new approaches to be evaluated. Most researches use lexical and structured databases. Some unstructured or web data approaches are available.

3.1 BENCHMARKS

One of the major boosts for the field of semantic relatedness is the availability of evaluation datasets (benchmarks). These benchmarks make it possible to compare the performance of different solutions. One of the first publicly available benchmarks is the dataset of Rubenstein and Goodenough [10]. This dataset consists of 65 word pairs, which are given a similarity score between zero and four. These similarity scores are based on the average of 51 human assigned scores. A higher value means a higher similarity between concepts. A research finding of Rubenstein and Goodenough is:

> *"There is a positive relationship between the degree of synonymy (semantic similarity) existing between a pair of words and the degree to which their contexts are similar."*

This finding shows that the context of concepts can be used to measure their relatedness. A second dataset is created by Miller and Charles [11]. This dataset consists of a subset of 30 word pairs from the Rubenstein and Goodenough dataset with the same ranking.

A current collection of semantic annotated words pairs is the WordSimilarity-353 test collection [12]. This dataset is consists of 353 words pairs with human assigned scores. This dataset also contains all the word pairs from the dataset of Miller and Charles, but with new similarity scores. This dataset is gaining popularity due to the higher number of word pairs compared to the Rubenstein and Goodenough dataset. It includes semantic similar and semantic related word pairs [13]. Four entries in this dataset are shown in Table 2.

Calculating semantic relatedness values for each concept pair enables researchers to compare their scores with these benchmarks. A common way to compare these semantic relatedness values is by calculating their correlation.

Correlation is a frequently used method to evaluate semantic relatedness algorithms. The available WordSimilarity-353 test collection consists of a restricted set of interval values (0-10). With the use of

Table 2: Four entries from the WordSimilarity-353 test collection.

First concept	Second concept	Human-assigned score (0-10)
Jaguar	Tiger	8.00
Jaguar	Cat	7.42
Jaguar	Car	7.27
Jaguar	Stock	0.92

correlation it is possible to calculate the linear dependency between two datasets. The usual calculation for this would be the Pearson product-moment correlation coefficient denoted as r. The Pearson's r is a widely used statistical measurement to find linear dependencies between datasets. The result of the Pearson's r calculation is a value between -1 and +1. A positive value means that when the first variable increases the second one increases too. A negative value means that when the first variable decreases the second variable decreases too.

Semantic relatedness algorithms can produce non-linear values e.g. values on a logarithmic scale. A statistical method that works on linear and non-linear values is the Spearman correlation.

The Spearman correlation is denoted as ρ and is named after Charles Spearman,[1] an English psychologist, who was active in the fields of statistics. The Spearman correlation uses the ranking of values instead of its absolute value to calculate the correlation coefficient.

The calculation of the Spearman correlation between set $X = \{x_i\}_1^n$ (measurement results) and set $Y = \{y_i\}_1^n$ (the benchmark) is given by

$$\rho(X, Y) = 1 - \frac{6 \sum_{i=1}^{n} (R(x_i) - R(y_i))^2}{n(n^2 - 1)}. \tag{1}$$

With $R(x_i)$ as the rank of x_i and $R(y_i)$ as the rank of y_i in this equation.

The Spearman correlation is a de facto standard to evaluate semantic relatedness research. This correlation measurement will therefore be used to evaluate the different approaches. An example of calculating the Spearman correlation for two datasets can be found in Appendix A.

1 Charles Spearman and Karl Pearson were both professors at the University College London and the statistical work in the field of correlation created a feud between them.

3.2 RELATED RESEARCH

Research in the field semantic relatedness is focussed on lexical [2] and structured databases [7][8]. The researches using unstructured data or web data are limited, but present.

The research of Finkelstein et al. [12] uses a vector-based approach, where each concept is represented as a vector in a multi-dimensional space. To obtain data for semantic comparison they sampled 10,000 documents in 27 different knowledge domains like computers, business and entertainment. Using a correlation-based metric they achieved a Spearman score of 0.44 with these multi-dimensional vectors on the WordSimilarity-353 test collection.

A similar vector-based approach is used by Reisinger and Mooney [14]. They collect the occurrences of words from a corpus (text collection) and cluster these vectors in different word-types. The semantic similarity between two word-types is computed as a function of their cluster centroids, instead of the centroid of all the word occurrences. This clustering of centroids results in a Spearman score of 0.77 on the WordSimilarity-353 test collection.

Cilibrasi and Vitanyi [15] describe the Normalized Web Distance to measure the semantic distance between concepts. This measurement uses the co-occurrence of concepts to estimate their semantic relatedness. This co-occurrence measurement is estimated by using the number of search results for each concept. Garcia et al. [16] use this measurement to calculate the semantic relatedness of the concepts in the WordSimilarity-353 test collection. They achieve Spearman scores ranging from 0.41 to 0.78 for different online search engines e.g. Yahoo, Google and Altavista. This Normalized Web Distance is used as a reference algorithm in this thesis.

4 METHODS FOR RELATEDNESS

To compare concepts by using web data, a selection of web pages related to these concepts is obtained. These web pages are the search results for each concept. These search results are used by the Normalized Compression Distance, the Jaccard index on keywords and the Normalized Web Distance to calculate the relatedness of concepts.

4.1 WEB DATA

The web data used to represent the concepts are search results for the concept. These web pages are obtained by querying a search server with the lexical form of the concepts i.e. the textual representation of the concept is used as query. These queries can be general like "car" and "chef" or specific like "Volkswagen Golf 1.6 TDI BlueMotion" and "Jamie Oliver". This lexical form of the concepts can result in web pages with different contexts. The query "tiger" will result in web pages about the animal, the Asian airline, the baseball team and the golf player. All of these pages give an insight in the different meanings of the concept "Tiger". These search results are the input for the algorithms as shown in figure 1.

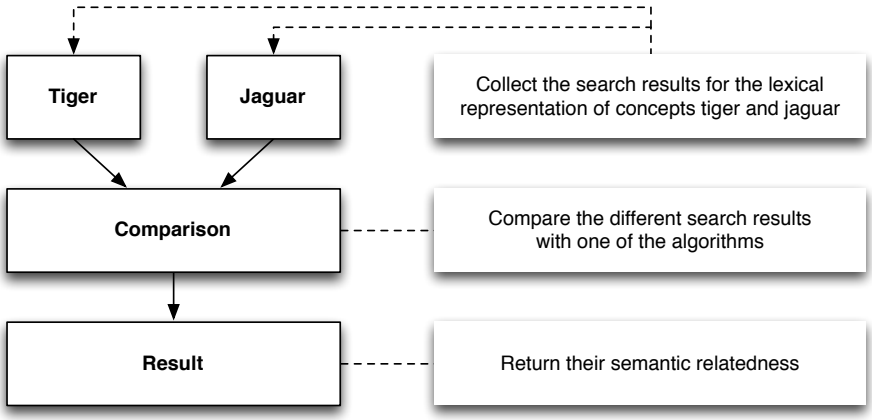

Figure 1: The comparison of "Tiger" and "Jaguar".

The algorithms in this thesis use web data to estimate semantic relatedness. The theory behind these estimations is the Distributional Hypothesis. This linguistical theory states that *"words that occur in the same contexts tend to have similar meanings"*[17]. This theory is supported by multiple researches as listed in "the Distributional Hypothesis" [18]. An example of a supporting statement is from Rubenstein and Goodenough [10] *"There is a positive relationship between the degree*

of synonymy (semantic similarity) existing between a pair of words and the degree to which their contexts are similar." The general idea behind the Distributional Hypothesis is described by Magnus Sahlgren [18] as *"there is a correlation between distributional similarity and meaning similarity, which allows us to utilize the former in order to estimate the latter"*.

This theory is applied by algorithms in different ways:

1. *The Normalized Compression Distance* uses the textual context of concepts and calculates the overlap between these contexts.
2. *The Jaccard index on keywords* uses text annotation to find keywords in the context of the concepts. The overlap between these keywords is calculated by the Jaccard index.
3. *The Normalized Web Distance* uses the explicit co-occurrence of concepts in their context to calculate semantic relatedness.

4.2 NORMALIZED COMPRESSION DISTANCE

The Normalized Compression Distance (NCD) is a universal similarity distance measure introduced by Rudy Cilibrasi and Paul Vitanyi in their article "Clustering by compression" [9]. The Normalized Compression Distance can detect patterns in two datasets. When there is a high overlap in these patterns, this results in a high similarity score. This distance measure has been applied successfully to the clustering of language families, the clustering of literature, the clustering of music files, whole-genome phylogeny of fungi and detecting viruses that are close to the SARS virus [9]. This research focusses on finding patterns in text. The detection of patterns depends on external compressors. These compressors can be block-sorting (Bzip2), Lempel-Ziv (Zlib) and statistical (PPMZ) [19].

The following sentence spoken by John F. Kennedy for his inaugural address in 1961[1] is used to explain the detection of patterns and the compression of text.

> "Ask not what your country can do for you - ask what you can do for your country."

This sentence can be compressed by searching for patterns. These patterns in text are usually words or parts of words that occur multiple times. The patterns found in this example are shown in figure 2. The spaces are replaced by horizontal lines in this figure.

To create a compression of the sentence a numbered index is used. Each number replaces a word in the sentence as shown in figure 3.

1 see http://computer.howstuffworks.com/file-compression.htm for the full example and explanation

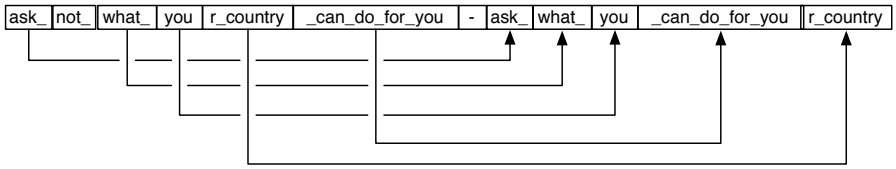

Figure 2: Pattern recognition.

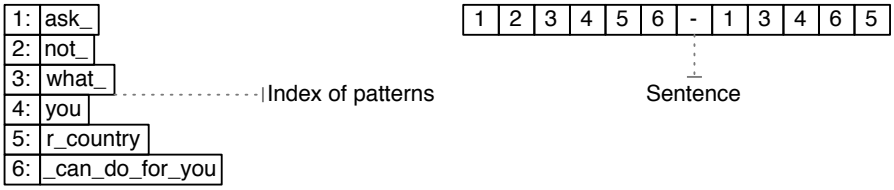

Figure 3: Compression index.

The compression of this sentence shows the basics of a compressor. Compressors use different methods to find these patterns. The compression can be optimized in multiple ways by compressors, but their basic functionality will still be the same. With the knowledge that a compressor finds patterns and compresses them the Normalized Compression Distance is introduced.

The Normalized Compression Distance uses two input datasets. A third dataset is constructed by chaining the data of these datasets. All three datasets are the input for the compressor, which compresses the datasets. The sizes of these compressed datasets are entered into the NCD as

$$\mathrm{NCD}(x,y) = 1 - \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))}. \tag{2}$$

In this equation[2] the datasets are represented by x, y and xy. C is the compressor used. $C(x)$ is the size of the input x after compression. $C(xy)$ is the size of the chained input of x and y after compression.

The NCD calculates the overlap between datasets by using compressed data sizes. By compressing data the compressor can discover patterns in this data. A high number of patterns in data results in a small compression size. By compressing the chained datasets not only the number of patterns in the datasets are calculated, but also the patterns that exist between them. The compression sizes of all three datasets are used in the equation to calculate a relatedness measurement. This measurement will approach a value of one, when the datasets are very different and only a limited number of patterns between the datasets can be found. The NCD returns a low value if their relatedness is high.

2 The "1−" is added to the original NCD for it's easier comparison with the other algorithms and the human assigned scores.

To explain this equation (2) some fictional data is used. Two datasets x and y are provided to the NCD algorithm. These datasets differ but they have some text patterns that overlap. To find these overlapping text patterns a compressor is used. In figure 4 the data of these datasets is shown. The dataset in the center represents the datasets x and y chained together. The solid black lines show the overlapping text patterns between the datasets.

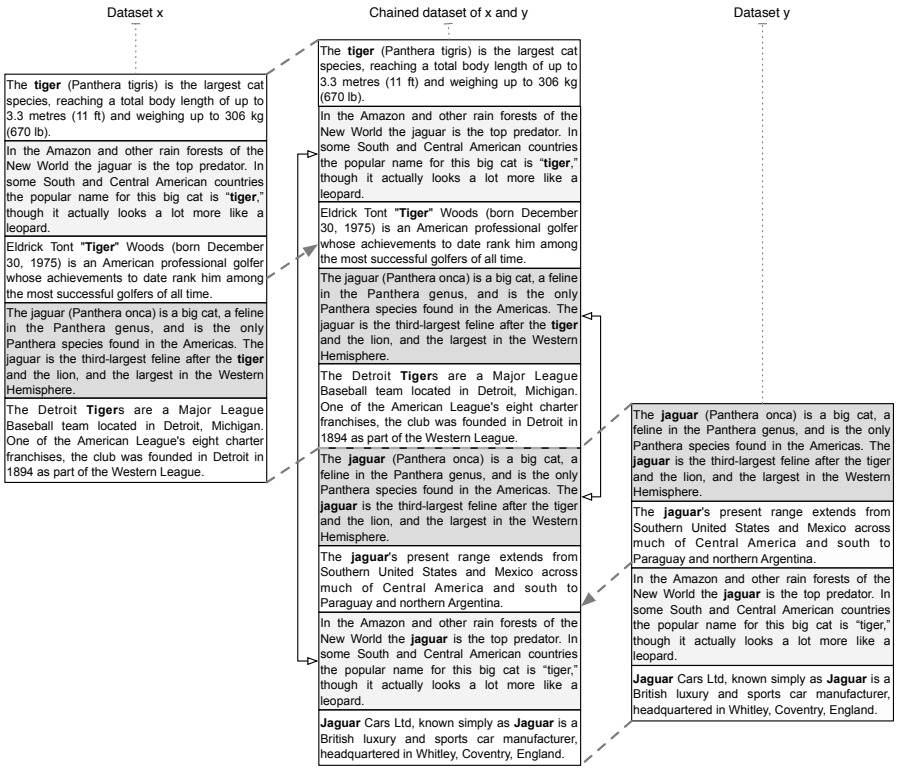

Figure 4: Two datasets with their chained combination in the center. The left dataset contains texts about the concept tiger. The right dataset contains texts about the concept jaguar.

Compressing these datasets results in binary files. These binary files (figure 5) contain the data to reproduce the original data. The file sizes of these binary files are used to calculate the similarity between the input datasets x and y. The values in figure 5 result in

$$\text{NCD} = \frac{262 - min(202, 143)}{max(202, 143)} = \frac{262 - 143}{202} = 0,5891089109. \quad (3)$$

The NCD algorithm is a practical implementation of the theoretic Normalized Information Distance. The proof of this algorithm is given in "The Similarity Metric"[20]. The Normalized Information Distance is a theoretic algorithm, because it's using the non-computable Kolmogorov complexity as compressor.

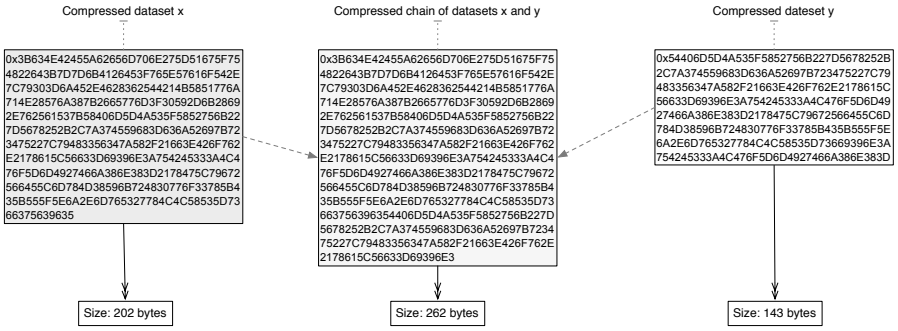

Figure 5: Three compressed datasets and their sizes. The center datasets is a chained version of the left and right dataset.

4.2.1 *Compressors*

Multiple compressors can be used by NCD to calculate a relatedness value. The used compressors are Bzip2, Zlib and Snappy. The first compressor, Bzip2, is a block-sorting compressor, the latter ones, Zlib and Snappy, are Lempel-Ziv compressors.

The block-sorting compressor, Bzip2[3], is based on the Burrows-Wheeler transform algorithm[21]. The algorithm creates an altered [4] representation of input data. It groups similar characters together. Sorting data can create such an alteration too. The advantage of the Burrows-Wheeler transformation is that the original dataset can be recreated using the altered representation. The grouping of similar characters makes it possible to find patterns and use these to compress the data. The compression of this permutated data is done in multiple steps. The type of steps and the number of steps used during compression of data depend on the required level of compression. A higher compression level will use more steps and therefore be slower and use more memory.

The Lempel-Ziv compressors, Zlib and Snappy, build a dictionary of common patterns to compress data. An example of this compression is given in 4.2. The first Lempel-Ziv compressor, Zlib, is used in file compression formates like gzip and zip. Zlib is also the standard for transferring compressed web pages between the web browser and web server. The second Lempel-Ziv compressor, Snappy, is a fast compressor developed by Google with a focus on high throughput. This compression is used in their BigTable storage and in their MapReduce framework. This focus on high throughput leads to a lower compression ratio. For more information about these compression techniques the reader is referred to "Common pitfalls using normalized compression distance"[19].

3 Site: http://bzip.org/
4 E.g. the word BANANA is altered to BNNAAA

4.2.2 *Compression settings*

Bzip2 and Zlib compressors have two settings, compression level and block/window size. A higher compression level means more compression, but also a longer execution time. The block/window size defines the size of the scope used to analyse the data. A larger block/window size will result in a higher compression, but uses more memory too.

The Normalized Compression Distance is based on the Normalized Information Distance that uses the Kolmogorov complexity as theoretic compressor. This theoretic compressor provides the most optimal compression of an object. In practice this most optimal compression is approached by using the settings that result in the highest compression ratio.

4.2.2.1 *Bzip2*

Block size is the most important setting for Bzip2. It defines the size of the scope used to analyse data. This setting is similar to the "Window size" in Zlib. The "Work factor" setting has a limited impact on the compression. This setting is similar to the "Compression level" in Zlib.

BLOCK SIZE An integer from 1 to 9. A higher value gives a higher compression, but uses more memory. The used value is 9, the highest.

WORK FACTOR An integer from 1 to 250. This value controls the compression phase, when presented with highly repetitive input data. A higher value will lead to a better compression but it impacts the execution time of the compression negatively. The default[5] value for this setting is 30, which is used in this thesis too.

4.2.2.2 *Zlib*

Zlib has two important settings. The "Compression level" defines the level of compression. The "Window size" defines the scope of the data analysed.

COMPRESSION LEVEL An integer from 1 to 9. A higher value gives a higher compression, but it takes longer. The used value is 9, the highest.

WINDOW SIZE An integer from 1 to 15. This defines the size of the window used to analyse and compress the data. The used value is 15, the highest.

5 Documentation http://bzip.org/1.0.5/bzip2-manual-1.0.5.html

4.3 JACCARD INDEX ON KEYWORDS

The Jaccard index on keywords algorithm uses keywords, which are extracted from web pages to calculate the size of overlap. This overlap shows the relatedness between keyword sets. To extract these keywords from web pages an external application is used. This application is created by Kalooga to find keywords in the content of news publishers. With these keywords a short keyword list can be built that describes the data.

The process of keyword extraction is based on multiple steps. The first step is annotating the text using a Part-Of-Speech-tagger[6]. This annotation adds grammatical tags to individual words in the text. This tagging of words can classify nouns, verbs, adjectives, etc. This first step in analysing the text makes it possible for a computer to select the words of interest like the nouns in a text. Examples of these nouns are "doctor", "dog" and "dogs". These nouns will be matched to entries in a database. Because this database cannot contain all the different derivatives of a word, a stemmer is used. A stemmer [7] converts a word to its stem. E.g. "dogs", "doglike" and "doggy" are converted to "dog". These stemmed nouns are matched to entries in a structured database. This structured database contains multiple entries (definitions/meanings) for every match. E.g. "tiger" is represented as animal and golf player. Between all these different entries a distance is calculated. The distances between entries make it possible to disambiguate words and to score the importance of words. Words with the highest importance are returned as keywords. The steps for extracting keywords from text are summarized in figure 6.

An Open Source alternative for keyword extraction is the project "Wikipedia Miner", as discussed in the paper "An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links" [8].

Some basic measurements for calculating a distance between two keywords sets are the Cosine similarity, the Jaccard index and Dice coefficient [22]. In this research the Jaccard index is chosen for its extendibility. It is modified by adding extra weight factors. Similar adjustments to the Jaccard index are done by existing similarity measurements like the Tanimoto's similarity, the Dice coefficient and the Tversky index.

4.3.1 *Jaccard index*

The Jaccard[8] index, or Jaccard similarity coefficient, is a statistic used to calculate the similarity of two datasets. The Jaccard index will be

6 See http://en.wikipedia.org/wiki/Part-of-speech_tagging for a basic introduction
7 See http://en.wikipedia.org/wiki/Stemming for more information
8 The Jaccard index is named after the botanist Paul Jaccard

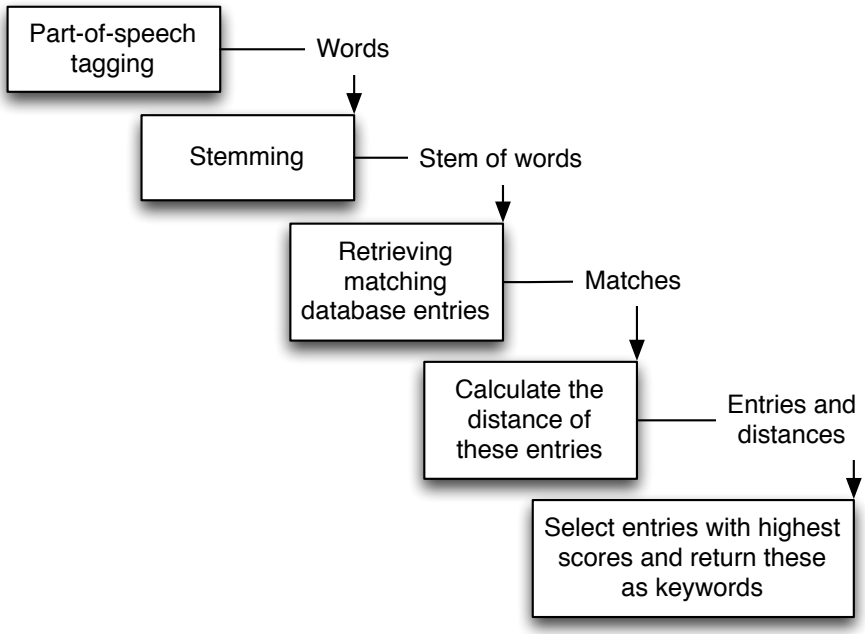

Figure 6: Keywords extraction overview.

used to calculate the similarity between two sets of keywords. The equation of the Jaccard index is

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{4}$$

In this equation A and B are two datasets of keywords. The numerator is the size of the intersection between dataset A and dataset B. The denominator is the size of their union. The result of this calculation is a number from 0 to 1. To function properly this calculation has the requirement $|A \cup B| > 0$.

One of the downsides of the Jaccard index is that it doesn't take the number of occurrences into account. To solve this a weighted version of this Jaccard index is used.

4.3.2 *Weighted Jaccard index*

To give more importance to keywords that occur often the Jaccard index is adapted to include weights, inspired on the Sørensen similarity index[9]. See figure 7 for an example of weights. Applying weights impacts the results. To make the results from the weighted Jaccard index comparable these need to be normalized.

The weighted Jaccard equation is given by

$$WJ(A,B) = \frac{\sum(k \in A \vee k \in B : NW(k,A) + NW(k,B))}{2}. \tag{5}$$

9 See http://en.wikipedia.org/wiki/Sorensen_similarity_index

The numerator of this equation is the total of the normalized weights of shared keywords between input sets. The normalized weights are calculated in

$$\mathrm{NW}(k, X) = \frac{W(k, X)}{\sum x \in X : W(x, X)}. \tag{6}$$

This equation normalizes the weight by dividing the weight of each keyword by the sum of all weighted keywords. The weight function used is

$$\mathrm{W}(k, X) = |x \in X : k \in x|. \tag{7}$$

This weight function counts the number of occurrences of each keyword in the input set. The weight function is sensitive to noise. To cope with this noise two variants are developed in 4.3.3 and 4.3.4.

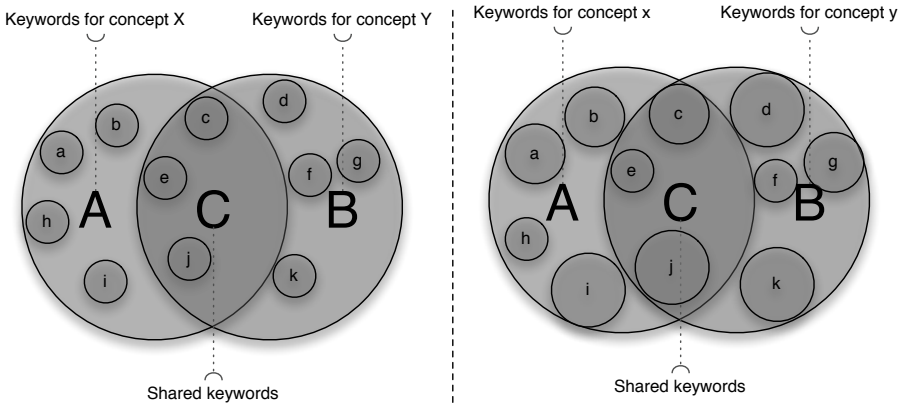

Figure 7: Normal (left) and weighted (right) Jaccard index of keywords.

4.3.3 *Weighted Jaccard index with Collection Frequency*

The Weighted Jaccard index with Collection Frequency and the Weighted Jaccard index with TF-IDF (Term Frequency-Inverse Document Frequency) are created to cope with noise. These two modifications are based on common techniques used in the field of Information Retrieval. These techniques are used to suppress generic words and stop words[10]. The first modification uses Collection Frequency of keywords to suppress generic words and stop words. This Collection Frequency is defined by the number of occurrences of the keyword in the total dataset. The weight function that uses this Collection Frequency is given by

$$\mathrm{W}(k, X, D) = \frac{d \in X : k \in d}{|\{d \in D : k \in d\}|}. \tag{8}$$

10 e.g. the, is, at, which, on, etc

In this equation the number of occurrences of a keyword is divided by the number of occurrences of this keyword in the total dataset. The total dataset is given by D.

4.3.4 *Weighted Jaccard index with TF-IDF*

The previous modification of the Jaccard index uses the Collection Frequency to calculate the relevance of a keyword by suppressing generic and stop words. This modification uses the Inverse Document Frequency instead of the Collection Frequency to achieve a similar goal. The Inverse Document Frequency of a keyword is given by

$$\mathrm{W}(k, X) = |x \in X : k \in x| * idf(k, D). \tag{9}$$

This equation is used in

$$\mathrm{idf}(k, D) = \log \frac{|D|}{|\{d \in D : k \in d\}|} \tag{10}$$

to calculate the new weight for each keyword.

4.4 NORMALIZED WEB DISTANCE

The Normalized Web Distance is a distance measure based on the co-occurrence of concepts. This distance measure is introduced in the paper "The Google Similarity Distance" [23] where it is called the Normalized Google Distance. In a later paper [15] this method is renamed to the Normalized Web Distance (NWD).

For the Normalized Web Distance to function a search engine is needed. Such a search engine can be "Google", "Yahoo" or any other search engine. This search engine provides three sets of data to the NWD. The first two datasets are the web pages linked to the different concepts. The third dataset contains the web pages that are linked to both concepts. This is shown in figure 8. The left and right datasets are collections of the web pages that are respectively linked to the concepts x and y. The dataset in the center contains the web pages that are linked to both concepts x and y.

The size of this intersection is used in the NWD equation to calculate the relatedness between concepts. The equation[11] of NWD is

$$\mathrm{NWD}(x, y) = 1 - \frac{max(log(f(x)), log(f(y))) - log(f(x, y))}{log(N) - min(log(f(x)), log(f(y)))}. \tag{11}$$

11 The "1−" is added to the original NWD for it's easier comparison with the other algorithms and the human assigned scores.

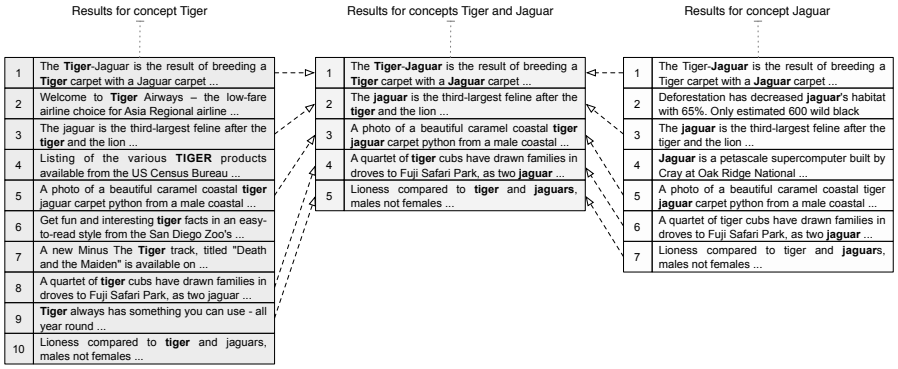

Figure 8: An example of data that NWD uses to calculate relatedness.

In this equation the concepts are represented by x and y. The f is the search engine used to find linked web pages. The $f(x)$ stands for the number of web pages linked to concept x. The $f(x,y)$ is the number of web pages linked to both concept x and concept y and N is the index size of the search engine. This equation shows some resemblance to the equation of NCD. This resemblance originates from their shared theoretic equation "Normalized Information Distance". This relation to the Normalized Information Distance is explained in more detail in the paper "The Google Similarity Distance"[23].

The algorithm returns a value between zero and infinity. Due to the logarithms on the input numbers and the size of N, the returned values usually stay within the zero to one range. The returned value of this algorithm is the opposite of semantic relatedness. The lower the value, the higher its relatedness.

E.g. the concept "tiger" is compared to the concept "cat". When using a search engine like Google the number of web pages linked to "tiger" is 567,000,000 and the number of web pages linked to "cat" is 2,510,000,000. The combination of "tiger AND cat" returns 143,000,000 web pages. The estimated size of the Google index is 43,000,000,000 at this moment. The resulting distance between "tiger" and "cat" is 0,6619220971 as

$$\begin{aligned} \mathrm{NWD}(x,y) &= \frac{max(log(567*10^6), log(251*10^7)) - log(143*10^6)}{log(43*10^9) - min(log(567*10^6), log(251*10^7))} \\ &= 0,6619220971. \end{aligned} \tag{12}$$

Comparing the concept "cat" with the concept "fisherman" gives a higher distance. The concept "fisherman" has 94,400,000 links and

"cat" and "fisherman" together 7,520,000. The resulting NWD value is 0,9492041527 as

$$\begin{aligned} \mathrm{NWD}(x,y) &= \frac{max(log(944*10^5), log(251*10^7)) - log(752*10^4)}{log(43*10^9) - min(log(944*10^5), log(251*10^7))} \\ &= 0,9492041527. \end{aligned} \tag{13}$$

When comparing the values of "cat-tiger" and "cat-fisherman", "cat-tiger" has a lower distance and is therefore more related than "cat-fisherman".

5 RESEARCH SET-UP

To represent the web data of concepts web pages are needed. These web pages are the input for the algorithms and have an impact on the results. For the retrieval of these web pages a software architecture is specified. From this retrieved data the text is extracted. This extracted text serves as input for the algorithms.

5.1 ACTIVITIES

The set-up for comparing concepts by using web data is divided in four activities.

1. *Collecting* web data that represents WordSimilarity-353 test collection concepts. This web data is the input for the algorithms Normalized Compression Distance, Jaccard index on keywords and Normalized Web Distance.
2. *Comparing* the collected web data by applying the three algorithms. These algorithms are impacted by the input data and their parameters.
3. *Gathering* the semantic relatedness scores from the algorithms for each concept-pair in the WordSimilarity-353 test collection.
4. *Evaluating* the results of the algorithms with the human assigned semantic relatedness scores in the WordSimilarity-353 test collection.

These activities and their relations are shown in figure 9.

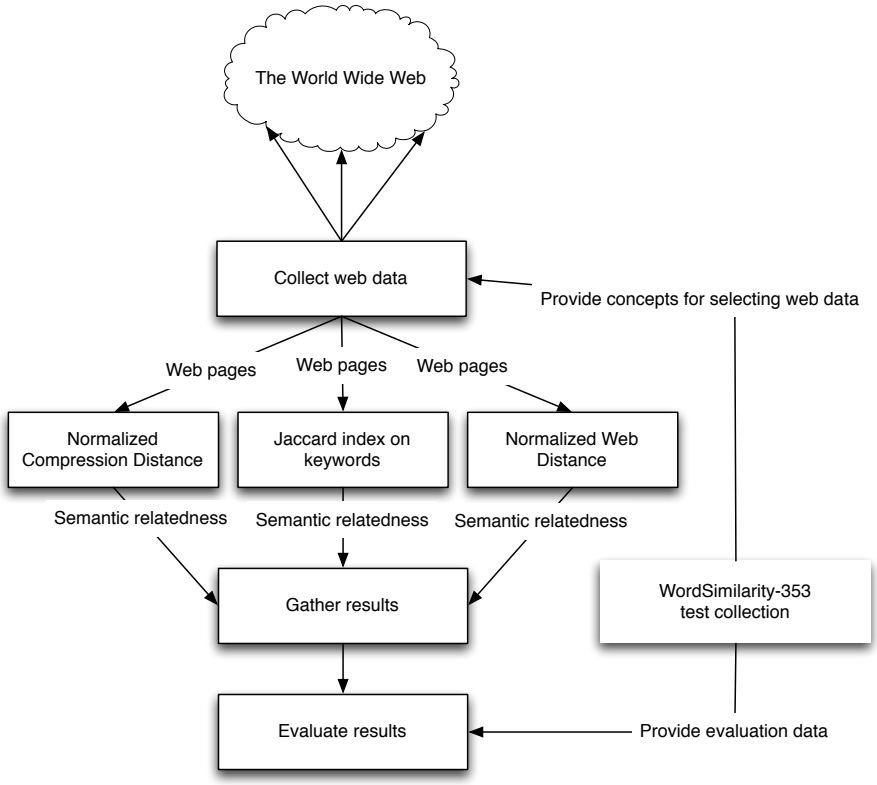

Figure 9: Research steps.

5.2 SOFTWARE ARCHITECTURE

For the retrieval and storage of a high volume of web pages a stable software architecture is a prerequisite. The software architecture is based on the architecture of the Apache Nutch project [24]. This project is proven to work with high volumes of data. The Nutch project is used for the retrieval and storage of web pages. In this thesis the architecture of this project is extended to support searching in these web pages and functionality is added for comparing concepts. The software architecture consists of five components. These components are shown on the left side of figure 10. Each component has specific tasks:

CRAWLER This component fetches web pages from the Internet.

STORAGE This component stores the fetched web pages in a database.

PROCESSOR This processor retrieves the pages from storage and processes these pages e.g. extract text or add extra information like keywords. The results of these processes are stored in the database.

SEARCH SERVER An index is build from the processed pages of the processor. This search server can be used to search in the content of web pages.

SEMANTIC RELATEDNESS COMPARISON APPLICATION This application uses the search server to retrieve web pages that are linked to concepts. These web pages are used to calculate the relatedness score between two concepts.

The architecture is implemented using different Open source programs. Apache Nutch[1] does the fetching and processing of web pages. This framework provides a crawler that fetches web pages and provides an interface for running long time processes. These processes can extract text from the fetched pages and retrieve keywords for each web page. These fetched web pages are stored in Apache HBase[2], a column oriented database. To search in these web pages a search server, ElasticSearch[3], is used. This search server returns the web pages that are associated with the concepts. All the algorithms for the semantic relatedness comparison are implemented in the client application written in the programming language Ruby[4]. The resulting architecture and its components is shown on the right side of figure 10.

This architecture makes it possible to fetch web pages from the Internet and build a search index of 704,562 web pages. How these pages are gathered is discussed in chapter 6. This index makes it possible to calculate relatedness values between concepts using the algorithms discussed in chapter 4. Before these algorithms are discussed the text extraction of web pages is explained.

1 Website: http://nutch.apache.org/
2 Website: http://hbase.apache.org/
3 Website: http://www.elasticsearch.org/
4 Website: http://www.ruby-lang.org/

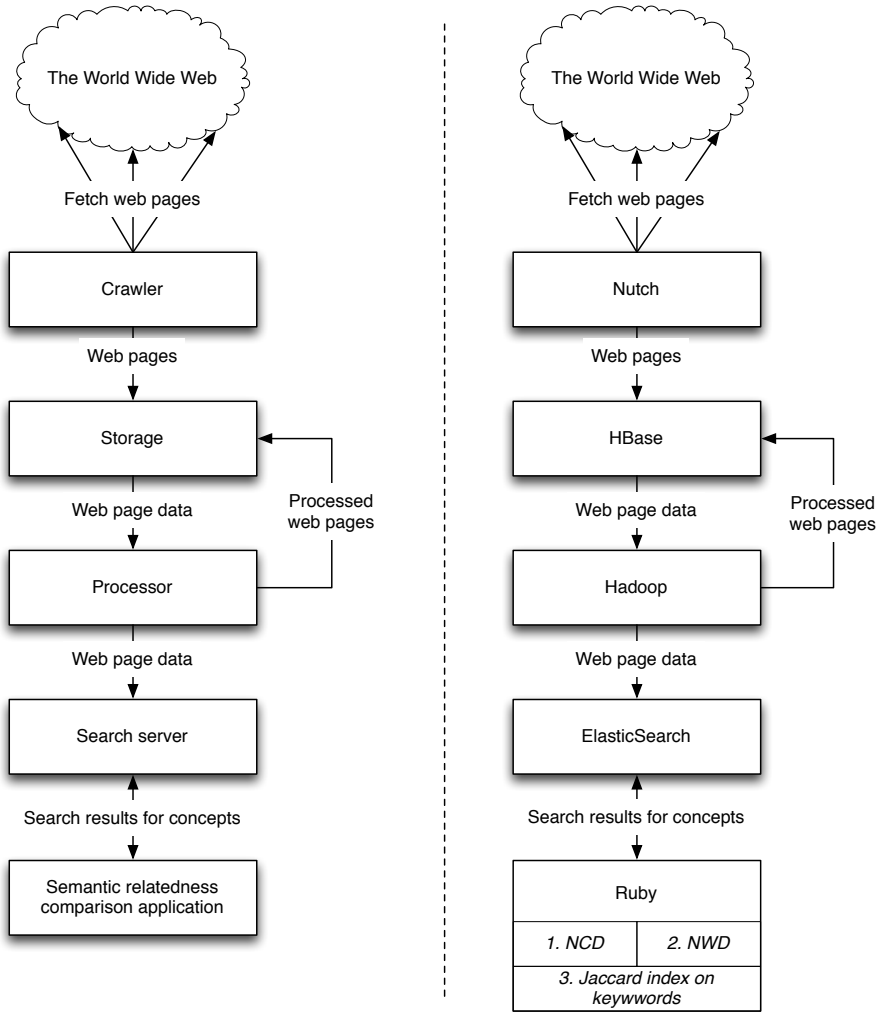

Figure 10: Architecture overview. The left overview shows the five components. The right one shows the implementation.

5.3 TEXT EXTRACTION

The extracting of text from web pages makes it possible to estimate semantic relatedness of concepts.

The structure of web pages is diverse and almost unique for each web site. This uniqueness makes it difficult to extract valuable information from these structures. To extract this information and use it for semantic relatedness a customised analyzer would be needed for almost every web site. Therefore only the text from web pages is used and the structure of web pages is ignored.

To extract text from a web page different methods can be used. The different content, structure and layout of web pages make it hard to extract all textual elements. In this research a DOM (Document Object Model[5]) parser is used to extract text from a web page. This technique processes all the elements on a web page. From these elements a DOM-tree is built that represents the web page. The visual text in this DOM-tree is used as the textual representation of the web page.

The textual representation of a web page is usually hard to read due to the loss of structure and layout of a web page during text extraction. E.g. the Wikipedia page about the chess piece king depicted in Figure 11. Although this article has a clean layout and a high focus on content, the text that remains after removing structure and layout is still hard to read. A part of the text from this article is shown in figure 12.

When reading the text, the first part can be identified as an introduction to chess. The second half is hard to read without seeing the rendered web page. The noise that is introduced by the use of text extraction on web pages is one of the downsides of using web data for semantic relatedness.

5 http://en.wikipedia.org/wiki/Document_Object_Model

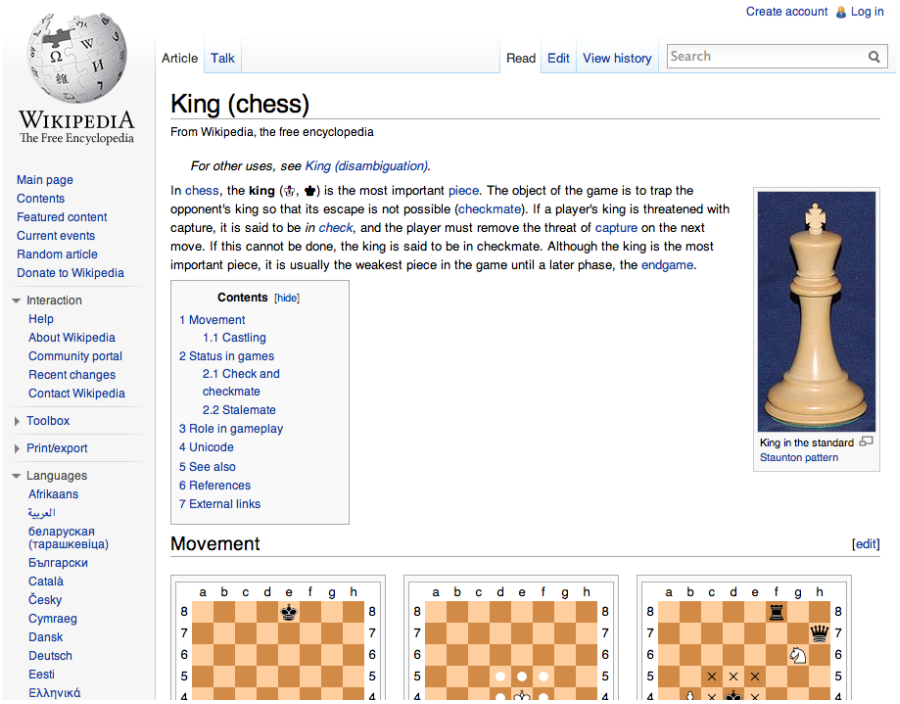

Figure 11: The Wikipedia article about chess piece king.

```
King (chess) - Wikipedia, the free encyclopedia King (chess) From
Wikipedia, the free encyclopedia Jump to: navigation , search For
other uses, see King (disambiguation) . King in the standard
Staunton pattern In chess , the king is the most important piece .
The object of the game is to trap the opponent's king so that its
escape is not possible ( checkmate ). If a player's king is
threatened with capture, it is said to be in check , and the player
must remove the threat of capture on the next move. If this cannot
be done, the king is said to be in checkmate. Although the king is
the most important piece, it is usually the weakest piece in the
game until a later phase, the endgame . Contents 1 Movement 1.1
Castling 2 Status in games 2.1 Check and checkmate 2.2 Stalemate 3
Role in gameplay 4 Unicode 5 See also 6 References 7 External links
[ edit ] Movement a b c d e f g h 8 8 7 7 6 6 5 5 4 4 3 3 2 2 1 1 a
b c d e f g h Initial placement of the kings. a b c d e f g h 8 8 7
7 6 6 5 5 4 4 3 3 2 2 1 1 a b c d e f g h Possible movements of the
unhindered king piece. a b c d e f g h 8 8 7 7 6 6 5 5 4 4 3 3 2 2 1
1 a b c d e f g h Possible movements of the king piece when hindered
by the borders or other pieces. The black king cannot move to the
squares under attack by the white bishop, the white knight, the
white queen, or the white pawn, and the white king cannot move to
the squares under attack by the black queen.
```

Figure 12: The extracted text from the Wikipedia article.

5.4 INPUT OF THE ALGORITHMS

The algorithms need web data to calculate semantic relatedness. This data is retrieved by a search query from the search server. This data is the textual representation of the web pages related to the concepts specified in the search query. The type of analysis performed on this data differs for each algorithm.

5.4.1 *Normalized Compression Distance*

The Normalized Compression Distance algorithm (par. 4.2) tries to find overlapping text patterns between two datasets to calculate a relatedness value. These overlapping text patterns are detected by a compressor. The data provided to this algorithm comes from the search server.

The input and the compressor have an impact on the Normalized Compression Distance. The input is full web pages or text fragments. The compressors are Bzip2, Zlib or Snappy. The impact of the input and the compressors on the NCD process is shown in figure 13.

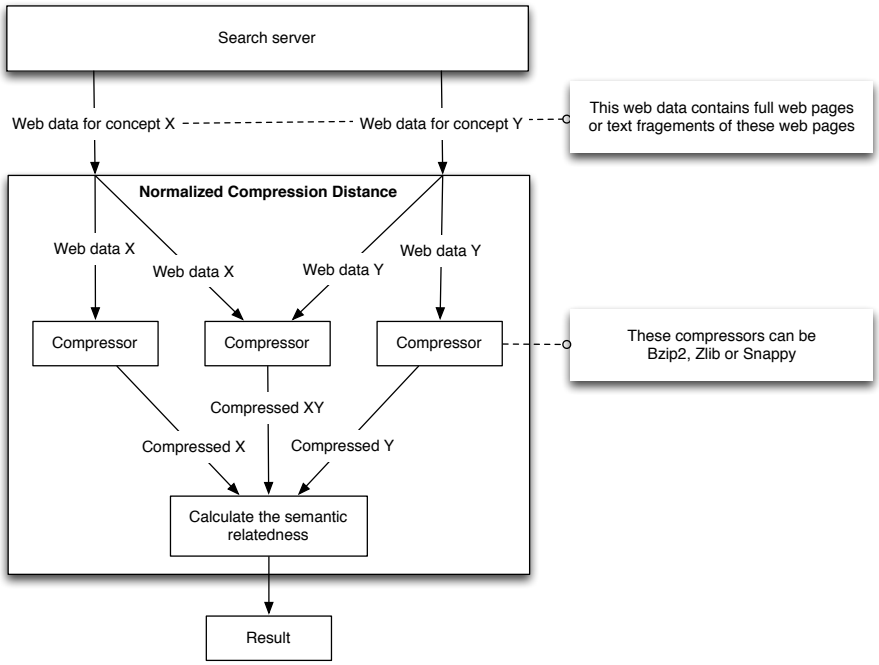

Figure 13: The NCD process.

The data provided to the NCD is a representation of the concepts that are analysed. These representations can be a collection of web pages or a collection of text fragments that contain the concepts. The search server provides this input.

The text fragments are more specific than full web pages. The use of text fragments is common for search engines. By using these text fragment the users are provided with an impression of the context for

each search result. E.g. Google uses text fragments, which is shown in figure 14.

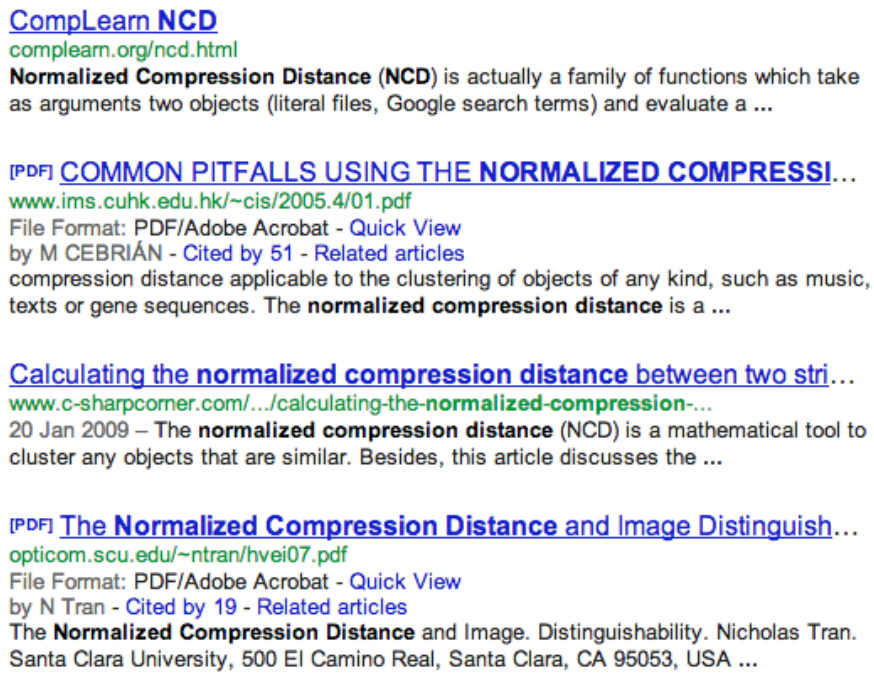

Figure 14: The use of text fragments by Google for the search query "Normalized Compression Distance".

5.4.2 *Normalized Web Distance*

The Normalized Web Distance uses the results provided by the search server to calculate a semantic relatedness score. With these results the co-occurrence of concepts is calculated. This technique is explained in more detail in 4.4. To calculate semantic relatedness based on co-occurrence three queries are executed on the search server. The first and second search query retrieve the number of web pages that are associated with respectively the first and the second concept. The third search query retrieves the number of web pages associated with both concepts. These three quantities are used by the NWD equation to calculate the relatedness between the first and second concept. The process executed to calculate this co-occurrence measurement is shown in figure 15.

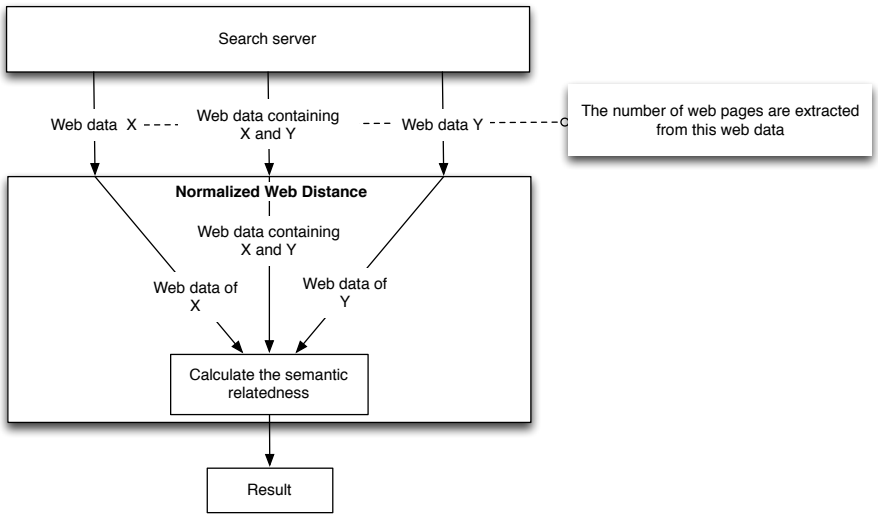

Figure 15: The NWD process overview.

5.4.3 *Jaccard index on keywords*

The third algorithm implemented makes use of keyword extraction. These keywords extracted from web pages are used to calculate a semantic relatedness value. An external keyword extractor performs the extraction of keywords from text. This keyword extractor has assigned a category to various keywords. Examples of categories are "American footbal", "animals", "Formula One", "newspapers", "people", "countries" and "places". In this research two specific categories of keywords are used beside general keywords. These categories are "places" and "people". Keywords from the category places include all kind of locations like cities and countries. The people category includes sports people, movie stars etc.

The Wikipedia article about the chess piece King illustrates this keyword analysis. This article is shown in figure 11. The results of this analysis of the Wikipedia article are:

GENERAL KEYWORDS "piece", "King"

CATEGORY COUNTRIES "Germany", "Germany", "Germany", "Germany", "Germany", "Germany"

CATEGORY CITIES "Role, West Pomeranian Voivodeship"

CATEGORY ENTERTAINMENT "Attack No. 1"

CATEGORY TELEVISION "Family Guy", "Family Guy", "Family Guy", "Family Guy", "Family Guy", "Family Guy"

CATEGORY PEOPLE "Don Most", "Shakira", "Shakira", "White Queen (Through the Looking-Glass)", "White King (Through the Looking-Glass)", "White King (Through the Looking-Glass)", "White King (Through the Looking-Glass)"

CATEGORY MUSIC "Island Records", "Links 2-3-4", "The Move"

CATEGORY WEBSITES "Wikipedia", "Wikipedia", "Wikipedia"

CATEGORY GAMES "Chess", "Chess", "Chess", "Chess"

The country Germany is an example of an unexpected keyword in this context. This relates to the functioning of this keyword extractor. E.g. the keyword Germany is found multiple times by its abbreviation "d e" in the text fragment "a b c d e f g h"(the representation of the columns of a chess board). To handle this noise in the returned keywords, the keyword extractor scores the keywords. The result of this scoring is clearly visible in the general keywords. These words give a good representation of the content of this Wikipedia article. Each web page that is used is analysed using this keyword extractor. These keywords are added to the index of the search server. With this search server not only the web pages but also the keywords associated with a search query can be analysed. The process to calculate semantic relatedness using keywords is shown in figure 16.

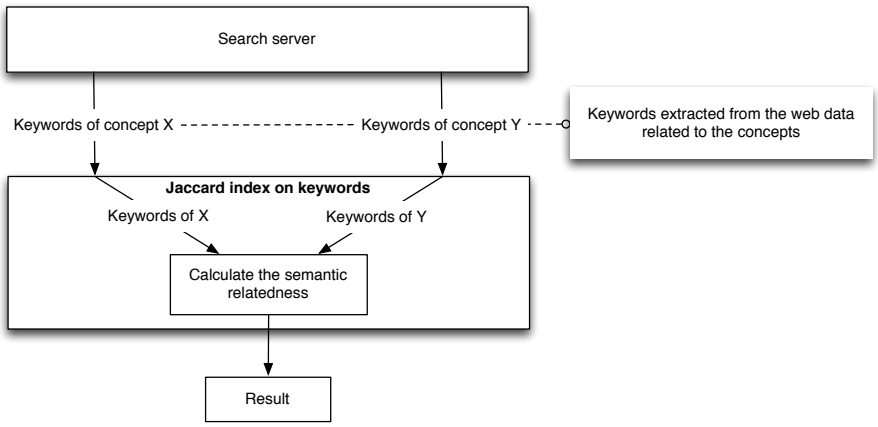

Figure 16: The keyword comparison overview.

The search server delivers two lists of keywords. Each list contains the keyword and the number of occurrences of this keyword. These lists are compared using four different comparisons. All these comparisons are based on the Jaccard index (par. 4.3.1).

6 TEST FRAMEWORK

To test the Normalized Compression Distance, Normalized Web Distance, Jaccard index on keywords and their different parameters the WordSimilarity-353 test collection is used. This dataset consists of 437 concepts grouped in 353 different concepts pairs. All these concept pairs have a human assigned relatedness score (3.1). These human scores will be compared to the scores of the algorithms in chapter 7.

For the algorithms to function properly input data is needed. The input data consists of web pages that contain the concepts of the WordSimilarity-353 test collection. Three data sources, Google, Wikipedia and IMDb, are used to collect these web pages. To explore the results of the algorithms a web application is built.

6.1 DATASET

The algorithms have to be able to work with heterogeneous data and have to provide stable results even with noisy data. To verify the quality of the different algorithms three data sources will be used. From these data sources[1] the web pages, which contain the concepts from the WordSimilarity-353 test collection, are collected. For collecting the first dataset the Google search engine is used to gather web pages that contain a concept. Google provides links to web pages of diverse websites. The second data source that is used for collecting web pages is Wikipedia. The different Wikipedia articles that contain concepts of the WordSimilarity-353 test collection dataset are gathered. The last data source used to collect web pages is the International Movie Database (IMDb). All the movie data related to concepts is collected. The process to collect data from the three sources is shown in figure 17. The figure illustrates that the Google dataset is a collection of web pages from different websites and the Wikipedia and IMDb datasets are collections of web pages from respectively Wikipedia and IMDb. The collected web pages are stored in a database for further processing (par. 5.2).

1 The data sources are publicly available on the Internet.

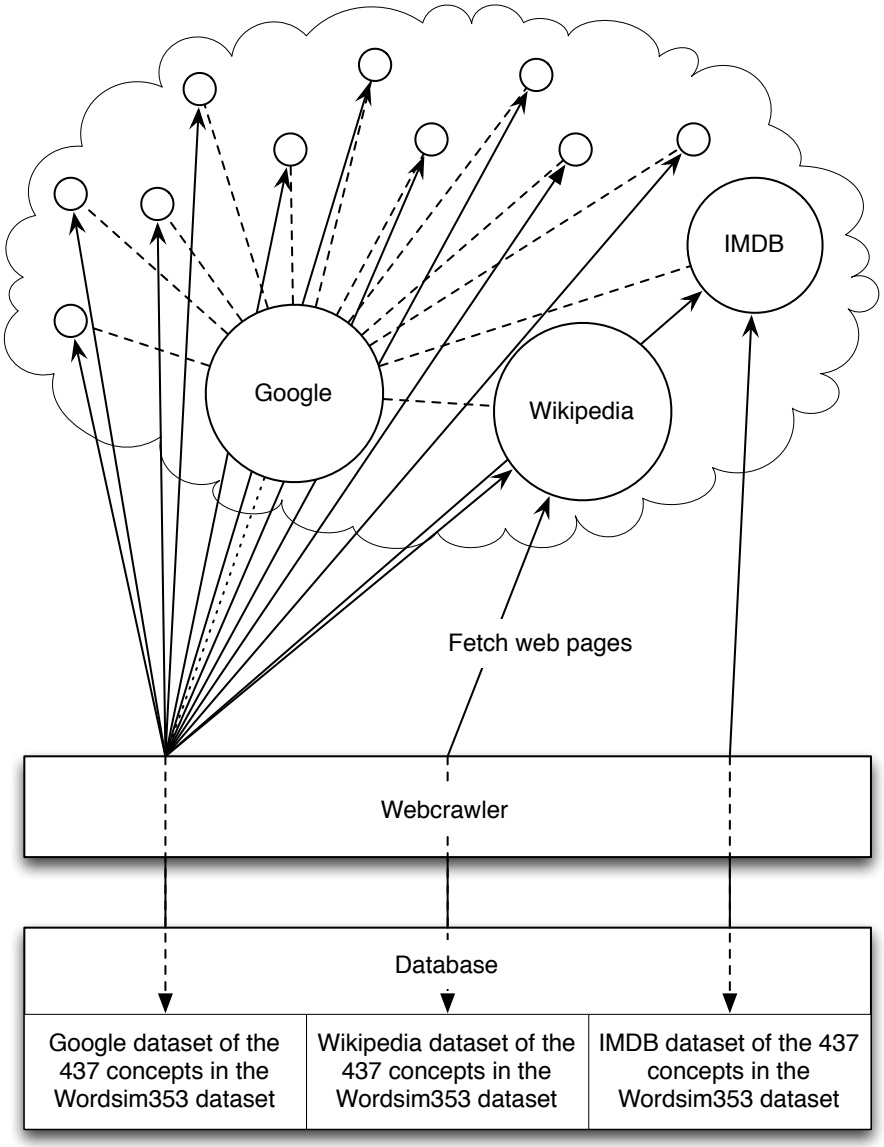

Figure 17: The process to collect the three datasets.

6.1.1 *Google index top 500 search results*

The first data source is based on the search results from Google. This data source is chosen for the high number of web sites that are indexed by Google. These diverse web sites symbolize the diversity of the Internet. Each of the 437 concepts in the WordSimilarity-353 test collection is entered as query in the Google search engine. For these search results the first 500 page urls for each concept are saved. These 218500 (500x437) page urls are used as seeding pages for the crawler. In total the crawler retrieved 174,714 pages for all the concepts in the WordSimilarity-353 test collection. This comes down to an average of 400 pages per concept. This is lower than the input of 500 pages per concept. This decrease in the number of pages comes from the type of search results that Google provides. Some of these results contain data that cannot be parsed by the crawler e.g. pdf files, images. Another cause for this lower number of web pages is an unresponsive server at the time of retrieval of a web page.

6.1.2 *Wikipedia top 500 search results*

The second data source used is Wikipedia. Wikipedia contains information on diverse topics. The number of topics covered by the English Wikipedia is currently more than 4.1 million. This data source is chosen for its higher quality of information on various topics. The search functionality provided by Wikipedia is used to gather web pages related to the 437 concepts. From these results 186,365 pages are parsed by the crawler. The average number of web pages per concept for this dataset is 426.

6.1.3 *IMDb search results*

IMDb stands for the Internet Movie Database[2]. This last data source is focussed on movies. Most of these movies are unrelated to the general concepts in the WordSimilarity-353 test collection. This data source is therefore chosen to see the impact of this unrelated information on the comparison of general concepts. The search functionality provided by IMDb is used to collect web pages related to the concepts. A search on IMDb gives a long list of results for each concept. E.g. the concept "Jaguar" returns search results for movie titles, actors names, characters, keywords and companies. In total 130 pages are linked to the word "jaguar". Another search on IMDb for "doctor" results in 2838 pages. The IMDb search interface provides two choices, a limited result set or all results. To gather all web pages related to a concept

2 This database started in 1990 by a group of international film fans and has now grown to a collection of 2 million movies/tv programs and more then 4 million cast en crew members.

all results are used. This results in a dataset of 343,483 IMDb pages with an average of 786 web pages per concept.

6.2 WEB APPLICATION

The three datasets Google, Wikipedia and IMDb make it is possible to calculate a relatedness value between two concepts. To be able to examine the results of these comparisons a web application is created for this research. This web application gives the user the possibility to enter two different concepts from the WordSimilarity-353 test collection dataset and calculate the relatedness value between these concepts. With this application the user has the opportunity to explore the data and the algorithms. This results in a better understanding of the algorithms and the data.

This functionality is shown in figure 18. The keywords associated with the concepts are shown in figure 19. These keywords give a good indication of keywords that impact the results of the different Jaccard index measurements.

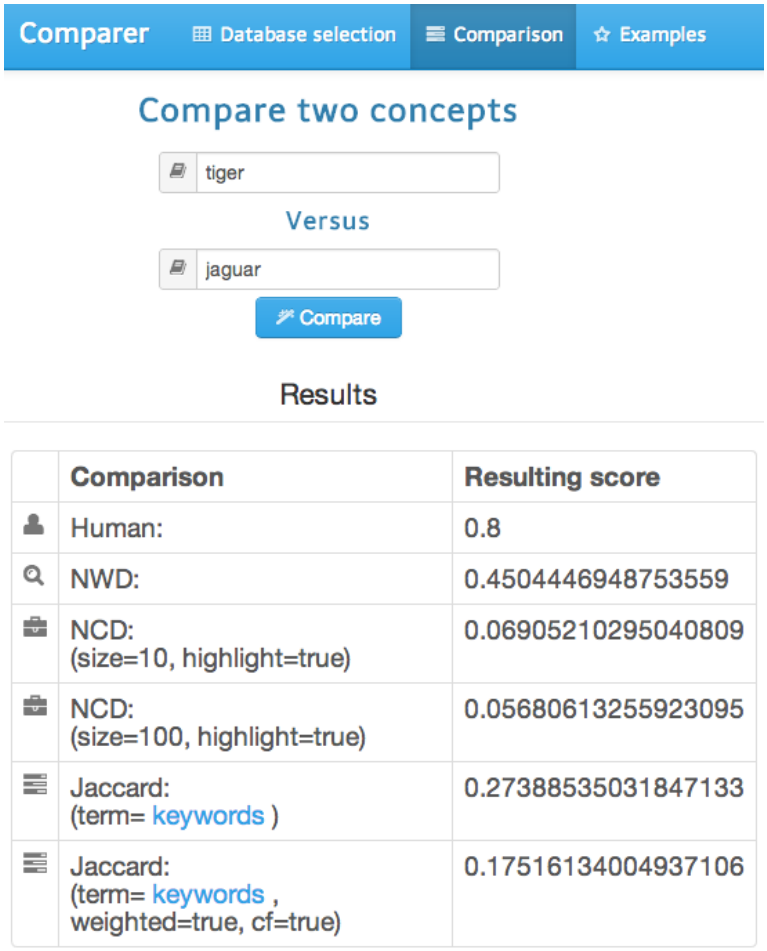

Figure 18: Partial screenshot of the user interface showing the results of the comparison between "Tiger" and "Jaguar".

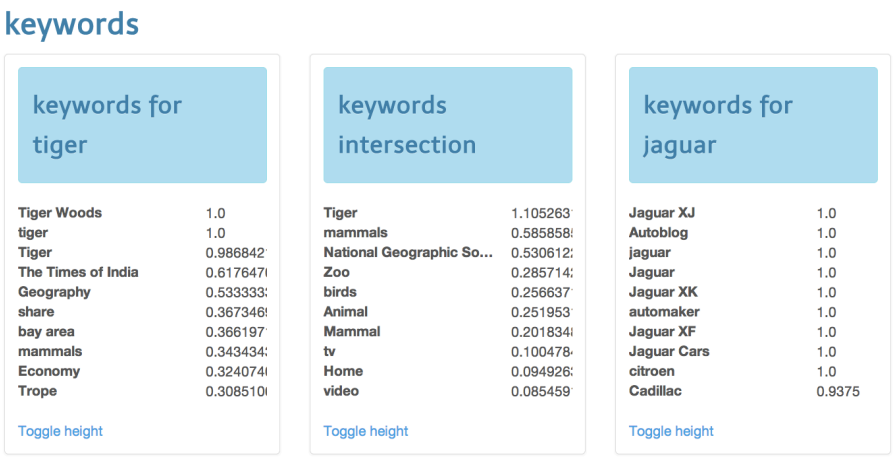

Figure 19: The weighted keywords from concepts “Tiger” and “Jaguar” and their intersection.

7 RESULTS

To evaluate the different algorithms for the concept pairs in the Word Similarity-353 test collection, their results are compared to the human assigned scores. The results are reviewed on three different attributes: accuracy, robustness and performance.

7.1 EVALUATION

To evaluate the different algorithms and their parameters, three datasets with web pages are used. The web pages in these datasets are related to the concepts of the WordSimilarity-353 Test Collection (par. 3.1). This test collection contains 353 concept pairs alongside a human assigned relatedness score. The dataset constructed with Google as data source contains 174,714 web pages from various websites. The datasets constructed with Wikipedia and IMDb as data source consist of respectively 186,365 and 343,483 web pages. The reason for using multiple datasets is to determine the impact of input data on the quality of the algorithms.

The concept pairs used to construct the datasets are input for the algorithms. The quality of each algorithm and parameter is measured for all three datasets. The quality measurement is based on the correlation between the human assigned relatedness score and the algorithm-assigned relatedness score of each concept-pair. The correlation measurement used is the Spearman correlation (par. 3.1). An example of this correlation measurement is given in appendix A. This example shows the calculation of the Spearman correlation for a set of ten concept pairs.

Before the results were gathered some tests were done on a smaller test set of 70 concept pairs. These tests showed some promising results for the Jaccard algorithms with Spearman scores that were up to 15% higher than the end results. This small test set increases the likelihood of results having the same order as the human assigned scores. These preliminary tests resulted in a list of parameters, which is tested on all 353 concept pairs in the WordSimilarity-353 Test Collection. The three algorithms and the parameters tested are listed in table 3.

The first algorithm NWD, Normalized Web Distance (par. 4.4), has no parameters. The second algorithm NCD, Normalized Compression Distance (par. 4.2), has three parameters. The first parameter is the number of pages used during compression. The second parameter

is the selected data (par. 5.4.1). The last parameter is the compressor used for compressing the data (par. 4.2.1).

The third algorithm Jaccard Index on keywords (par. 4.3) has also three parameters. The first parameter is the number of keywords that are compared. The second parameter is the keyword selection. This could be a keyword from the category "general", "people" or "places" (par 5.4.3). The last parameter, the weight factor, is the weight used during the calculation of the Jaccard Index. This weight factor is "None" (par. 4.3.1), "Normal" (par. 4.3.2), "Collection Frequency" (CF, par. 4.3.3) or "Inverse Document Frequency" (IDF, par. 4.3.4).

Table 3: The tested algorithms and their parameters.

Method	Size	Selection	Compressor	Weight factor
NWD	n/a	n/a	n/a	n/a
NCD	1, 2, 5, 10, 50, 100, 200, 300, 400, 500 (number of pages)	Content, Highlight	Bzip, Snappy, Zlib	n/a
Jaccard	10, 100, 1000 (number of keywords)	General, People, Places	n/a	None, Normal, CF, IDF

7.2 DATA SOURCES

The three datasets used are based on Google, Wikipedia and IMDb (par. 6.1). The highest result is achieved on the dataset that uses Wikipedia as data source. Closely followed by the dataset that uses Google as data source. The dataset that uses IMDb as data source had the lowest result. A complete overview of the three datasets and their Spearman scores for the algorithms is given in Appendix B. The three datasets and their best scoring algorithms will be discussed next.

7.2.1 *Data source Google*

The five best performing results for the dataset that uses Google as data source are given in table 4. The best scoring algorithm is the Normalized Web Distance (NWD) followed by two entries of the Normal-

ized Compression Distance (NCD). Between the NWD and the NCD algorithm is a gap. This gap is visible in all three datasets and shows that the NWD is overall the best performing algorithm. The difference between the two entries of the NCD is the number of web pages used during compression. This reduction in web pages from 200 to 100 web pages shows the limited impact of the input size on the resulting Spearman score (par. 7.3.1). The Jaccard index shows a good performance when the weighted Collection Frequency (CF) is used in combination with keywords in the category "places" or general keywords. The good performance of the keywords in the category "places" is unexpected. This shows the importance of location for semantic relatedness.

Table 4: Top 5 algorithms and parameters for the Google dataset.

Method	Size (*kw*= *keywords*)	Selection	Compressor	Weight factor	Spearman score
NWD	n/a	n/a	n/a	n/a	0.584299
NCD	200 pp.	Highlight	Bzip2	n/a	0.35305
NCD	100 pp.	Highlight	Bzip2	n/a	0.33238
Jaccard	1000 kw.	Places	n/a	CF	0.28543
Jaccard	100 kw.	General	n/a	CF	0.27972

7.2.2 *Data source Wikipedia*

The top five results for the dataset with Wikipedia as data source, given in table 5, is similar to the top five results of the previous dataset with Google as data source. There is a difference in the parameters of the Jaccard en NCD algorithms. The Jaccard index shows good performance in combination with general keywords and a keyword list of 100. The difference between the two Jaccard index entries is their weight factor. The weight factors are the Collection Frequency and the Inverse Document Frequency (IDF). An explanation for their similar performance could be that they both reduce the importance of generic keywords to filter noise.

The Normalized Compression Distance shows a good performance with the Bzip2 compressor. The difference between the two entries is their data selection and the number of web pages. In comparison to the previous dataset two types of selected data are shown: content and highlight. The good performance of NCD on the content could be due to the generic textual structure of Wikipedia pages. This generic structure makes it possible for the NCD to filter the shared elements (e.g. the text of the menu bar) as generic noise.

Table 5: Top 5 algorithms and parameters for the Wikipedia dataset.

Method	Size (*kw*= *keywords*)	Selection	Compressor	Weight factor	Spearman score
NWD	n/a	n/a	n/a	n/a	0.59627
Jaccard	100 kw.	General	n/a	CF	0.31896
NCD	100 pp.	Highlight	Bzip2	n/a	0.29002
Jaccard	100 kw.	General	n/a	IDF	0.28596
NCD	50 pp.	Content	Bzip2	n/a	0.27380

7.2.3 *Data source IMDb*

The dataset that uses IMDb as data source, table 6, has the lowest scores in comparison to the other datasets. The biggest difference in the results of this dataset and the results of the others is the dominance of the Jaccard index in the top five results. The Jaccard index is not only dominant in the top five but it is overall the best scoring algorithm in the dataset. The NCD algorithm appears for the first time on the 19th place with a score of 0,14.

The lower overall Spearman scores of this dataset were expected. The IMDb data source returns only movie facts related to the concepts. An unexpected result is the better performance of the Jaccard index in comparison to the NCD algorithm. A cause for the lower scores of the NCD algorithms could be the higher number of unique word patterns on the web pages of IMDb. These unique word patterns make it hard for the NCD algorithm to find textual patterns between web pages from IMDb.

Table 6: Top 5 algorithms and parameters for the dataset based on IMDb.

Method	Size (*kw*= *keywords*)	Selection	Compressor	Weight factor	Spearman score
NWD	n/a	n/a	n/a	n/a	0.34751
Jaccard	1000 kw.	Places	n/a	CF	0.22558
Jaccard	100 kw.	General	n/a	CF	0.21390
Jaccard	1000 kw.	General	n/a	IDF	0.18781
Jaccard	1000 kw.	People	n/a	CF	0.18497

7.3 ALGORITHMS AND PARAMETERS

The selected parameters for the Normalized Compression Distance and Jaccard index on keywords led to 97 combinations that are tested. There were in total 61 combinations for the NCD algorithm and 36 combinations for the Jaccard index.

7.3.1 *Normalized Compression Distance*

The Normalized Compression Distance scores are impacted by three parameters: the number of web pages, the selected data and the compressor. The results in previous section shows that the Bzip2 compressor is the only compressor in the top five results. The high Spearman scores of this compressor in comparison to the other compressors are probably caused by the higher compression ratio of the Bzip2 compressor in comparison to the Zlib and Snappy compressor.

To illustrate the impact of the number of web pages and the selected data on the results, the Bzip2 compressor is chosen as example in figure 20. In this figure the Spearman scores for the NCD are shown for the configured number of web pages and the content of the web pages as selected data. The figure shows three lines. The lines that reflect the general web data are the Wikipedia and Google data sources. These two lines show an ascending trend up to respectively 50 and 100 web pages. After these numbers of web pages the lines drop and the remaining results stabilise around zero. The IMDb line hovers around zero for all number of web pages.

The graphs showing the impact of all parameters on the Normalized Compression Distance are given in Appendix D.

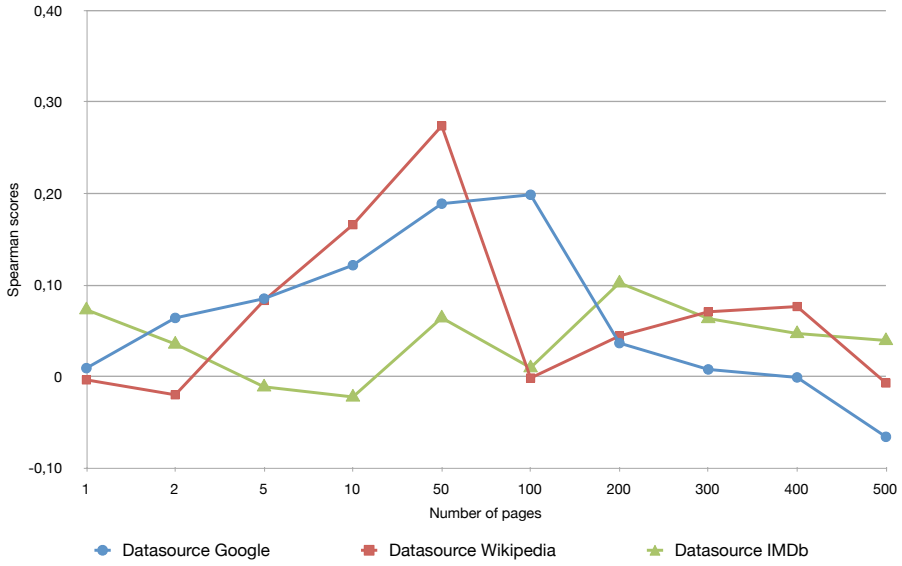

Figure 20: The results of the Normalized Compression Distance with content as selected data and Bzip2 as compressor.

The Spearman scores for the category "highlight" data are shown in figure 21. The Google and Wikipedia data sources have the highest scores in comparison to the IMDb data source. The scores of the Google and Wikipedia data sources show a fall after respectively 200 and 100 web pages. This trend is similar to the one in figure 20.

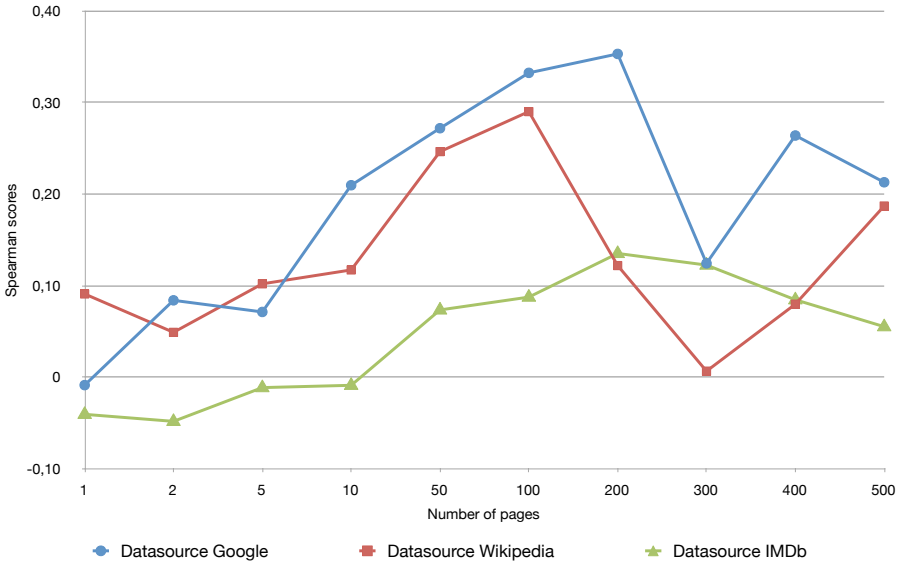

Figure 21: The results of the Normalized Compression Distance with highlight as selected data and Bzip2 as compressor.

An explanation for this dip in the scores is data that exceeds the window of analysis. The size of this window defines the data size analysed. When the input data size exceeds the window size, this overflow will be placed in the next window. This result is similar to the merging of two compressed datasets. An example of this behaviour is shown in figure 22. The first row contains two datasets of size 100, which fit both in the same window. The second row contains two datasets of size 200, which exceed the window size. A second window is needed for the remaining data. The third row contains two datasets of size 300, which fit together in two full windows. Almost no overlap exists between the two windows, which reduces the potential to discover textual patterns between these datasets. The last row contains two datasets of size 400, which fit in three windows. Due to the increased data sizes more overlap exists between the datasets in the second window. This increases the potential to discover textual patterns between these datasets.

The variation in the scores for the three datasets could be caused by the difference in writing styles. The Normalized Compression Distance is susceptible for different writing styles. This sensitivity is used for author recognition [9, p. 20] and plagiarism detection [25]. This explains the higher scores on the full content of Wikipedia articles, which are written in a formal structure [26]. The IMDb dataset

consists of short unstructured descriptions of movies. This results in lower scores.

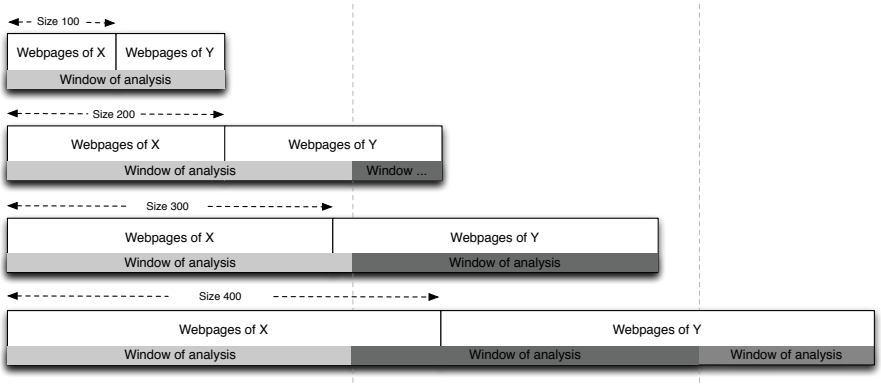

Figure 22: Example window sizes and compressions.

7.3.2 *Jaccard index on keywords*

The Jaccard index is tested with 36 combinations of different parameter configurations. The best performing parameters for the Jaccard index on keywords are the general keywords and the Collection Frequency as weight factor. These parameters are shown in figure 23, the lines in this figure are respectively the Google, Wikipedia, and IMDb data sources. All three data sources show a peak at 100 keywords. This trend is also visible in other parameter combinations (Appendix E).

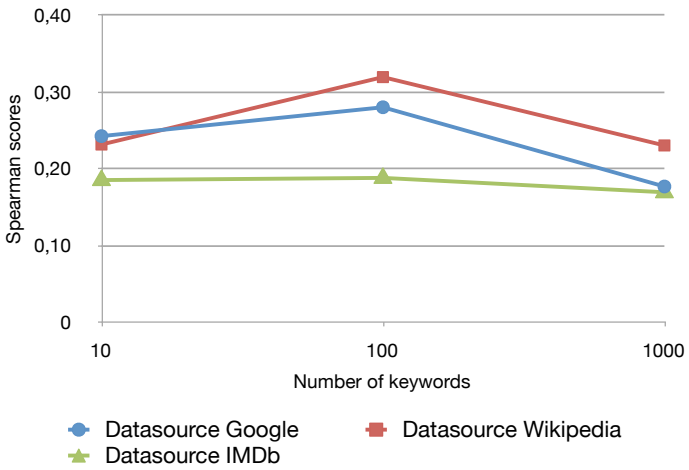

Figure 23: The results of the Jaccard index on keywords.

An explanation for the lower scores around 1000 keywords could be the introduction of extra noise by increasing the number of keywords. This heightens the chance that synonymous keywords describe the same concept, e.g. car and automobile. Another reason for

this increase in noise could be the higher number of specialized keywords that describe a concept. E.g. the various keywords used to describe the automobile Jaguar and its parts makes the comparison of the animals jaguar and tiger more difficult.

7.4 REVIEW

These results can be viewed from three different angles: the accuracy of the algorithms, their robustness with different data sources and the execution time of the algorithms.

7.4.1 *Accuracy*

The highest accuracy on all three datasets is achieved by the Normalized Web Distance (table 4, 5 and 6). This algorithm uses the co-occurrence of concepts to calculate their relatedness. The second place is for the Normalized Compression Distance. The Normalized Compression Distance discovers shared text patterns between web pages to calculate their relatedness. Last but not least is the Jaccard index. This algorithm follows the NCD algorithm closely, but results in slightly lower Spearman scores.

7.4.2 *Robustness*

The three different datasets give an overview of the accuracy of the algorithms when different data sources are used. The datasets based on the data sources Google and Wikipedia show that the difference between global web data and encyclopaedic data is of limited impact on the end results. This is in contrast to the movie related dataset, which results in lower Spearman scores. The Normalized Web Distance accomplishes the highest scores on all three datasets. The Normalized Compression Distance Spearman scores are so much lower for the IMDb data source that it is not robust. The short sentences in the IMDb dataset could be an explanation for this pattern. The NCD is sensitive for writing style, which could cause lower scores for the IMDb dataset. The results of the Jaccard Index are surprising. It shows that the weighted variants with the Collection frequency and the Inverse Document Frequency positively impact the results by lowering the importance of generic keywords.

7.4.3 *Performance*

The performance in the sense of time it takes for an algorithm to compare two concepts is not discussed until now. The algorithms have to retrieve two lists of web pages to calculate the semantic relatedness

of two concepts. The retrieval time of these lists is negligible, because all three algorithms have to retrieve these lists. The Normalized Web Distance can directly use these lists and is the fastest algorithm of all three with a computation time of a few nano seconds.

The Normalized Compression Distance on the other hand has to compress all web pages before the semantic relatedness can be calculated. The compression of web pages has the largest impact on the performance of this algorithm. The time it takes to compress web pages is given in figure 24. The time for a NCD comparison with a size of 100 is still under one second. The higher the number of web pages, the longer it takes for the NCD algorithms to calculate the semantic relatedness value. This higher number of web pages decreases the accuracy of the NCD. Depending on the dataset the advised number of web pages lies between fifty and two hundred.

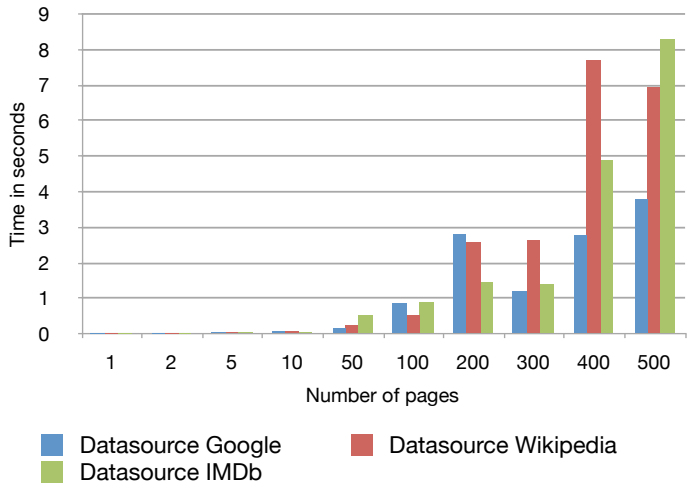

Figure 24: The performance of the Normalized Compression Distance with highlight as selected data and Bzip2 as compressor.

The Jaccard index has no heavy computational calculations. The time it currently takes to calculate this score is around one second. The current implementation is far from optimal and, if implemented efficiently, the calculation could probably be reduced to a few milliseconds. A bigger bottleneck is the time it takes to perform keyword analysis. It takes about two seconds to extract keywords from one web page (1 CPU; 3 GB of memory). For the Jaccard index to work efficiently an analysis of all web pages in the dataset is needed. The size of the dataset largely impacts the usefulness of the Jaccard index. E.g. the time it took to analyse all three datasets is 49 hours. The datasets contain a total of 704,562 web pages and the available number of CPU's is eight (704,562 * 2 / 8 / 3600 = 48,928). The high number of hours involved in this comparison process leads to higher costs compared to the other algorithms.

8 CONCLUSION

During this research three algorithms are implemented to calculate semantic relatedness of concepts with web data. To validate the accuracy of the algorithms the WordSimilarity-353 test collection, a golden standard, is used. This test collection makes it possible to compare the results of an algorithm to the human assigned semantic relatedness scores.

The Normalized Web Distance achieves the highest Spearman scores. This algorithm uses the co-occurrence of concepts to calculate their semantic relatedness. Lower Spearman scores are achieved by the Normalized Compression Distance. This algorithm discovers shared textual patterns in the web pages of concepts to calculate their semantic relatedness. To discover these textual patterns different compressors are used. The noise, the size of the input and the compression efficiency of the used compressor impacts the accuracy of this algorithm. A similar accuracy as the NCD is achieved by the Jaccard index on keywords. This algorithm uses specialised software to extract keywords on the web pages of concepts. These keywords are used to calculate their overlap with the Jaccard index. The accuracy of this algorithm is improved by applying weights to keywords. These weights filter generic keywords and amplify unique keywords. A disadvantage of this algorithm is the expensive process to extract keywords from web pages.

To answer the research question of this thesis, first the two sub questions have to be answered.

1. "What is the added value of Normalized Compression Distance on calculating semantic relatedness?"

The added value of the Normalized Compression Distance lies in the use of the context of concepts to calculate their semantic relatedness, although the results of this algorithm are lower than the NWD algorithm. The NCD shows that it is possible to calculate semantic relatedness on context. The scores of the NCD are influenced by the different writing styles on web pages. To achieve better scores further research into better compressors and cleaning of input data will improve the accuracy of this algorithm and decrease the sensitivity to writing style.

2. "What is the added value of keyword extraction on calculating semantic relatedness?"

The keywords extracted from the content are the added value of this algorithm. These keywords provide insights and let end users

discover new relationships between concepts. The disadvantage is the extensive analysis of the content for this algorithm to function properly. Therefore the use of this algorithm is only recommended when the added value of keywords improves the insights of the end user. When this is not the case, the NCD algorithm with a similar accuracy should be used.

The research question is defined as:

> "How can the context of concepts in web data be used to calculate semantic relatedness?"

The research shows that the context of concepts can be used in different ways to calculate semantic relatedness. The shared textual patterns of the Normalized Compression Distance and the overlapping keywords with the Jaccard index gives can be used to calculate semantic relatedness. The use of these algorithms depends on the application. The better performance of the Normalized Compression Distance and the higher scores on general web data make it a good candidate for applications with automated semantic relatedness calculations. For applications that provide exploratory insights in semantic relatedness, the Jaccard index on keywords is advised for its detailed and extensive output and the insights that this output can provide.

BIBLIOGRAPHY

[1] *Oxford English Dictionary*. Oxford University Press, 2013.

[2] Alexander Budanitsky and Graeme Hirst. "Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures." In: *Workshop on wordnet and other lexical resources, second meeting of the north american chapter of the association for computational linguistics*. 2001.

[3] Alexander Budanitsky and Alexander Budanitsky. *Lexical Semantic Relatedness and Its Application in Natural Language Processing*. Tech. rep. 1999.

[4] Philip Resnik. "Using information content to evaluate semantic similarity in a taxonomy." In: *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*. IJCAI'95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995, pp. 448–453. ISBN: 1-55860-363-8, 978-1-558-60363-9.

[5] Christiane Fellbaum. "WordNet and wordnets." In: *Encyclopedia of Language and Linguistics*. Ed. by Keith Brown. Oxford: Elsevier, 2005, pp. 665–670.

[6] P. M. Roget. *Roget's Thesaurus of English words and phrases*. Available from Project Gutemberg, Illinois Benedectine College, Lisle IL (USA), 1852.

[7] Michael Strube and Simone Paolo Ponzetto. "WikiRelate! computing semantic relatedness using wikipedia." In: *proceedings of the 21st national conference on Artificial intelligence - Volume 2*. AAAI'06. Boston, Massachusetts: AAAI Press, 2006, pp. 1419–1424. ISBN: 978-1-57735-281-5.

[8] David Milne and Ian H. Witten. "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links." In: *AAAI 2008*.

[9] R. Cilibrasi and P.M.B. Vitanyi. "Clustering by compression." In: *Information Theory, IEEE Transactions on* 51.4 (2005), pp. 1523 –1545. ISSN: 0018-9448.

[10] Herbert Rubenstein and John B. Goodenough. "Contextual correlates of synonymy." In: *Commun. ACM* 8.10 (Oct. 1965), pp. 627–633. ISSN: 0001-0782.

[11] George A. Miller and Walter G. Charles. "Contextual correlates of semantic similarity." In: *Language and Cognitive Processes* 6.1 (1991), pp. 1–28.

[12] Yossi Matias Lev Finkelstein Evgeniy Gabrilovich. "Placing search in context: the concept revisited." In: *ACM Trans. Inf. Syst.* 20.1 (Jan. 2002), pp. 116–131. ISSN: 1046-8188.

[13] Eneko Agirre et al. "A study on similarity and relatedness using distributional and WordNet-based approaches." In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL '09. Boulder, Colorado: Association for Computational Linguistics, 2009, pp. 19–27. ISBN: 978-1-932432-41-1.

[14] Joseph Reisinger and Raymond J. Mooney. "Multi-prototype vector-space models of word meaning." In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 109–117. ISBN: 1-932432-65-5.

[15] Rudi Cilibrasi and Paul M. B. Vitányi. "Normalized Web Distance and Word Similarity." In: *CoRR* abs/0905.4039 (2009).

[16] Jorge Gracia and Eduardo Mena. "Web-Based Measure of Semantic Relatedness." In: *Proceedings of the 9th international conference on Web Information Systems Engineering*. WISE '08. Auckland, New Zealand: Springer-Verlag, 2008, pp. 136–150. ISBN: 978-3-540-85480-7.

[17] Zelig Harris. "Distributional structure." In: *Papers in Structural and Transformational Linguistics*. Dordrecht, Holland: D. Reidel Publishing Company, 1970, pp. 775–794.

[18] Magnus Sahlgren. "Special issue of the Italian Journal of Linguistics." In: *Rivista di Linguistica* 20.1 (2008), pp. 33–53.

[19] Manuel Cebri An, Manuel Alfonseca, and Alfonso Ortega. "Common pitfalls using normalized compression distance: what to watch out for in a compressor." In: *Communications in Information and Systems* 5 (2005), pp. 367–384.

[20] Ming Li et al. "The Similarity Metric." In: *IEEE TRANSACTIONS ON INFORMATION THEORY*. 2003, pp. 863–872.

[21] A Block sorting Lossless et al. *A Block-Sorting Lossless Data Compression Algorithm*. Tech. rep. Digital SRC Research Report, 1994.

[22] Manu Konchady. *Building Search Applications: Lucene, LingPipe, and Gate*. First. Mustru Publishing. ISBN: 0615204252.

[23] Rudi L. Cilibrasi and Paul M.B. Vitanyi. "The Google Similarity Distance." In: *IEEE Transactions on Knowledge and Data Engineering* 19 (2007), pp. 370–383. ISSN: 1041-4347.

[24] Mike Cafarella and Doug Cutting. "Building Nutch: Open Source Search." In: *Queue* 2.2 (Apr. 2004), pp. 54–61. ISSN: 1542-7730.

[25] Efstathios Stamatatos. "A survey of modern authorship attribution methods." In: *Journal of the American Society for Information Science and Technology* 60.3 (2009), pp. 538–556. ISSN: 1532-2890.

[26] W. Emigh and S.C. Herring. "Collaborative Authoring on the Web: A Genre Analysis of Online Encyclopedias." In: *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*. 2005, 99a.

APPENDICES

A

EXAMPLE OF THE SPEARMAN CORRELATION

The Spearman correlation is used to calculate the quality of an algorithm in comparison to the human assigned scores. Spearman correlation equation is

$$\rho(X,Y) = 1 - \frac{6\sum_{i=1}^{n}(R(x_i) - R(y_i))^2}{n(n^2-1)}. \tag{14}$$

The inputs of this equation are two equal sized sets of data, X and Y. The $R(x_i)$ stands for the rank of $x_i \in X$, $R(y_i)$ is the rank of $y_i \in Y$ and n is the size of the sets. The result of this equation is a value between minus one and one. The magnitude of this result represent the strength of the correlation and a positive result represent a positive correlation, i.e. when set X goes up and set Y does the same. A negative result represents the inverse, i.e. set X goes down, set Y goes down.

An example of data used to calculate the Spearman correlation is given in table 7. In this table the results of ten concepts-pairs are given. The "Human" column shows the human assigned relatedness scores of each concept-pair. The "NWD" column gives relatedness scores for the Normalized Web Distance of each concept-pair. The "$R(x_i)$" and the "$R(y_i)$" columns represent the ranks of the human assigned scores and the NWD scores. The "Δ" and "Δ^2" represent the difference, $R(x_i) - R(y_i)$, and the distance squared, $(R(x_i) - R(y_i))^2$.

Table 7: Spearman example for 10 concepts and their score.

Concept pair	Human	NWD	$R(x_i)$	$R(y_i)$	Δ	Δ^2
tiger - tiger	10.0000	1.0125	1	1	0	0
king - queen	8.5800	0.5105	2	4	-2	4
computer - software	8.5000	0.7053	3	2	1	1
tiger - jaguar	8.0000	0.4504	4	5	-1	1
dollar - yen	7.7800	0.5336	5	3	2	4
train - car	6.3100	0.3703	6	6	0	0
hospital - infrastructure	4.6300	0.1431	7	8	-1	1
school - center	3.4400	0.3234	8	7	1	1
peace - insurance	2.9400	0.1286	9	9	0	0
professor - cucumber	0.3100	-0.0115	10	10	0	0

To calculate the Spearman correlation value of this table the variables of the Spearman equation are replaced with the values of the example in

$$\rho(X,Y) = 1 - \frac{6 * 12}{10(10^2 - 1)} = 1 - \frac{72}{990} = 0.9272727. \quad (15)$$

The number 12 in the equation represents the summed total of the Δ^2 column and the number 10 represents the size of the table. The resulting value, 0.9272727 shows a strong correlation between the human assigned scores and the NWD scores. In this thesis all the 353 concepts-pairs from the Wordsimilarity-353 Test Collection are used to calculate the Spearman correlation of each algorithm.

B

SPEARMAN SCORES OF THE THREE DATA SOURCES.

A complete overview of the three datasets and their Spearman scores for the algorithms is shown in table 8. The first column shows the methods used for calculating semantic relatedness. The second column shows the input size for these methods. For the Jaccard index this is the number of keywords and for the Normalized Compression Distance it is the number of web pages. The third column shows the selection of data processed. Those are keyword categories for the Jaccard index and page selection for the NCD. The fourth column shows the compressor, only applicable for the NCD. The fifth column gives the weight factor used in the Jaccard index. The sixth, seventh and eighth column show the Spearman scores for respectively the Google, Wikipedia and IMDb datasets.

Table 8: The Spearman scores of all algorithms and parameters for the three datasets.

Method	Size	Selection	Compressor	Weight factor	Data source Google	Data source Wikipedia	Data source IMDb
Jaccard	10	People	n/a	n/a	0,064682959	0,104634377	0,058227997
Jaccard	100	People	n/a	n/a	0,082362084	0,083925013	0,107193337
Jaccard	1000	People	n/a	n/a	0,066732241	0,057661234	0,127934415
Jaccard	10	People	n/a	CF	0,063281056	0,09094251	0,039548477
Jaccard	100	People	n/a	CF	0,085589906	0,088106648	0,134632246
Jaccard	1000	People	n/a	CF	0,099679393	0,104678368	0,182181019
Jaccard	10	People	n/a	IDF	0,068605288	0,099350384	0,053187624
Jaccard	100	People	n/a	IDF	0,090914001	0,095776161	0,123490814
Jaccard	1000	People	n/a	IDF	0,079015521	0,060930114	0,136331036
Jaccard	10	People	n/a	Normal	0,067386781	0,09614391	0,023601629
Jaccard	100	People	n/a	Normal	0,094489316	0,097228603	0,098726263
Jaccard	1000	People	n/a	Normal	0,076976674	0,055275167	0,11449783
Jaccard	10	Places	n/a	n/a	0,044684948	0,091674869	0,091335493
Jaccard	100	Places	n/a	n/a	0,091176104	0,054029379	0,100626318
Jaccard	1000	Places	n/a	n/a	0,151793168	0,082780438	0,127226882
Jaccard	10	Places	n/a	CF	-0,000675614	0,092251522	0,056357133
Jaccard	100	Places	n/a	CF	0,108162563	0,075700536	0,156766703
Jaccard	1000	Places	n/a	CF	0,285438455	0,226250644	0,213906459
Jaccard	10	Places	n/a	IDF	0,024325872	0,108533381	0,084001332
Jaccard	100	Places	n/a	IDF	0,064229617	0,055453722	0,096047471
Jaccard	1000	Places	n/a	IDF	0,135947874	0,081242539	0,136980801
Jaccard	10	Places	n/a	Normal	0,030217209	0,113633295	0,085991892

Table 8: The Spearman scores of all algorithms and parameters for the three datasets.

Method	Size	Selection	Compressor	Weight factor	Data source Google	Data source Wikipedia	Data source IMDb
Jaccard	100	Places	n/a	Normal	0,063318431	0,056387073	0,096234142
Jaccard	1000	Places	n/a	Normal	0,120056679	0,072912484	0,129137099
Jaccard	10	General	n/a	n/a	0,179511026	0,187240557	0,225587238
Jaccard	100	General	n/a	n/a	0,18290042	0,248950022	0,163398514
Jaccard	1000	General	n/a	n/a	0,120050882	0,136272245	0,154409486
Jaccard	10	General	n/a	CF	0,242160185	0,23124007	0,184975005
Jaccard	100	General	n/a	CF	0,279722672	0,318966489	0,187810253
Jaccard	1000	General	n/a	CF	0,176603005	0,23003193	0,169090563
Jaccard	10	General	n/a	IDF	0,209645437	0,13178022	0,170111077
Jaccard	100	General	n/a	IDF	0,245451026	0,285969003	0,177870209
Jaccard	1000	General	n/a	IDF	0,184109106	0,181255511	0,184900869
Jaccard	10	General	n/a	Normal	0,192442912	0,104089643	0,147318398
Jaccard	100	General	n/a	Normal	0,231564168	0,272702788	0,148431532
Jaccard	1000	General	n/a	Normal	0,184507136	0,177110569	0,151741607
NCD	1	Content	Bzip2	n/a	0,009146576	-0,003675504	0,072945835
NCD	2	Content	Bzip2	n/a	0,064005026	-0,019993987	0,035464304
NCD	5	Content	Bzip2	n/a	0,085068906	0,083098739	-0,011291819
NCD	10	Content	Bzip2	n/a	0,121585166	0,165848623	-0,022356046
NCD	50	Content	Bzip2	n/a	0,188835065	0,273800576	0,064109785
NCD	100	Content	Bzip2	n/a	0,198542266	-0,001672463	0,00961731
NCD	200	Content	Bzip2	n/a	0,036514282	0,044296603	0,10229046
NCD	300	Content	Bzip2	n/a	0,007735943	0,070681032	0,063338755
NCD	400	Content	Bzip2	n/a	-0,001006942	0,076428734	0,047011746
NCD	500	Content	Bzip2	n/a	-0,065899488	-0,007090337	0,039372515
NCD	1	Content	Snappy	n/a	0,029800287	-0,016276948	0,066819268
NCD	2	Content	Snappy	n/a	0,037914071	-0,0236551	0,035955976
NCD	5	Content	Snappy	n/a	0,023415777	-0,029669747	0,002666242
NCD	10	Content	Snappy	n/a	0,070858359	-0,051905552	0,051305506
NCD	50	Content	Snappy	n/a	0,067487243	-0,051021511	-0,098210174
NCD	100	Content	Snappy	n/a	0,003690304	-0,089752103	-0,016098121
NCD	200	Content	Snappy	n/a	-0,036896898	-0,025149691	0,006645657
NCD	300	Content	Snappy	n/a	-0,047468703	-0,035651111	-0,020723482
NCD	400	Content	Snappy	n/a	-0,100045778	-0,04037973	0,000628009
NCD	500	Content	Snappy	n/a	-0,095155791	-0,057834127	-0,032806109
NCD	1	Content	Zlib	n/a	0,027327535	-0,008768325	0,071075857

Table 8: The Spearman scores of all algorithms and parameters for the three datasets.

Method	Size	Selection	Compressor	Weight factor	Data source Google	Data source Wikipedia	Data source IMDb
NCD	2	Content	Zlib	n/a	0,074529293	-0,020719049	0,042595222
NCD	5	Content	Zlib	n/a	0,07492289	0,068249271	-0,003404126
NCD	10	Content	Zlib	n/a	0,184673618	0,099735592	-0,037100619
NCD	50	Content	Zlib	n/a	0,074205877	-0,017151987	-0,078364186
NCD	100	Content	Zlib	n/a	0,046434616	-0,071700933	-0,145427483
NCD	200	Content	Zlib	n/a	-0,014289799	-0,075808773	-0,065696517
NCD	300	Content	Zlib	n/a	-0,043009484	-0,065858703	-0,120468486
NCD	400	Content	Zlib	n/a	-0,063386906	-0,070075394	-0,101138792
NCD	500	Content	Zlib	n/a	-0,073272047	-0,076869868	-0,097352459
NCD	1	Highlight	Bzip2	n/a	-0,008794583	0,090795124	-0,040704238
NCD	2	Highlight	Bzip2	n/a	0,083835531	0,048880155	-0,048389847
NCD	5	Highlight	Bzip2	n/a	0,071148425	0,1018674	-0,01150884
NCD	10	Highlight	Bzip2	n/a	0,20958726	0,117053107	-0,008983299
NCD	50	Highlight	Bzip2	n/a	0,271870444	0,246311401	0,073356278
NCD	100	Highlight	Bzip2	n/a	0,332382612	0,290028691	0,087374766
NCD	200	Highlight	Bzip2	n/a	0,35305958	0,121811531	0,135080133
NCD	300	Highlight	Bzip2	n/a	0,124272824	0,006481016	0,122355241
NCD	400	Highlight	Bzip2	n/a	0,263916185	0,079671356	0,084458766
NCD	500	Highlight	Bzip2	n/a	0,212824426	0,186906229	0,05505801
NCD	1	Highlight	Snappy	n/a	0,017684853	0,12537641	-0,018508058
NCD	2	Highlight	Snappy	n/a	0,09908303	0,048431109	-0,066133286
NCD	5	Highlight	Snappy	n/a	0,073988447	0,10966602	0,007716641
NCD	10	Highlight	Snappy	n/a	0,191584787	0,015565596	-0,065575117
NCD	50	Highlight	Snappy	n/a	0,00451269	0,037389116	-0,039363103
NCD	100	Highlight	Snappy	n/a	0,01669919	-0,004616699	-0,12179107
NCD	200	Highlight	Snappy	n/a	-0,04094131	-0,044374626	-0,115274862
NCD	300	Highlight	Snappy	n/a	-0,035064978	-0,06712918	-0,05384646
NCD	400	Highlight	Snappy	n/a	-0,073539401	-0,056464211	-0,126362141
NCD	500	Highlight	Snappy	n/a	-0,077730721	-0,065993198	-0,092355531
NCD	1	Highlight	Zlib	n/a	0,013598498	0,082972633	-0,03641116
NCD	2	Highlight	Zlib	n/a	0,074792214	0,013973202	-0,068885669
NCD	5	Highlight	Zlib	n/a	0,085681365	0,097272662	0,012980787
NCD	10	Highlight	Zlib	n/a	0,217956941	0,075249444	0,010536339
NCD	50	Highlight	Zlib	n/a	0,129348731	0,038963027	-0,007435306
NCD	100	Highlight	Zlib	n/a	0,091324308	0,010071062	-0,066984863

Table 8: The Spearman scores of all algorithms and parameters for the three datasets.

Method	Size	Selection	Compressor	Weight factor	Data source Google	Data source Wikipedia	Data source IMDb
NCD	200	Highlight	Zlib	n/a	0,006473923	-0,042852891	-0,091647452
NCD	300	Highlight	Zlib	n/a	-0,013329984	-0,059568245	-0,059892344
NCD	400	Highlight	Zlib	n/a	-0,045120762	-0,062768309	-0,126524054
NCD	500	Highlight	Zlib	n/a	-0,047266687	-0,063232632	-0,133612345
NWD	n/a	n/a	n/a	n/a	0,584299513	0,596274108	0,347510927

C

CONCEPT PAIRS

The concept pairs of the WordSimilarity-353 Test Dataset are shown in table 9. The table shows the three algorithms and their normalized value for each concept pair. In the brackets next to the values of the algorithms is the index/rank of these values given. The difference between this index and the index of the human assigned values is assigned a colour between green (small difference) and red (large difference). The parameters for the Normalized Compression Distance are 200 pages with the selection of highlighted content and the Bzip2 compressor. The provided parameters for the Jaccard index on keywords are 1000 keywords from the category places and the Collection Frequency as weight factor.

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
1. tiger - tiger	1.000	0.770 (2)	1.000 (1)	1.000 (1)
2. fuck - sex	0.943	0.303 (157)	0.115 (218)	0.073 (269)
3. midday - noon	0.927	0.381 (68)	0.164 (23)	0.107 (81)
3. journey - voyage	0.927	0.352 (91)	0.136 (85)	0.106 (84)
5. dollar - buck	0.920	0.298 (165)	0.081 (336)	0.034 (350)
6. money - cash	0.913	0.343 (100)	0.154 (33)	0.121 (35)
7. coast - shore	0.908	0.356 (88)	0.168 (22)	0.136 (21)
8. money - cash	0.906	0.343 (100)	0.154 (33)	0.121 (35)
9. money - currency	0.902	0.282 (187)	0.139 (73)	0.119 (38)
10. football - soccer	0.901	0.640 (5)	0.131 (113)	0.159 (11)
11. magician - wizard	0.900	0.382 (66)	0.114 (229)	0.089 (191)
12. type - kind	0.895	0.316 (136)	0.122 (183)	0.102 (101)
13. gem - jewel	0.894	0.411 (47)	0.137 (83)	0.104 (92)
14. car - automobile	0.892	0.380 (69)	0.136 (87)	0.087 (200)
15. street - avenue	0.885	0.259 (220)	0.217 (5)	0.172 (8)
16. asylum - madhouse	0.884	0.337 (111)	0.143 (56)	0.106 (83)
17. boy - lad	0.880	0.288 (177)	0.146 (47)	0.094 (144)
18. environment - ecology	0.878	0.278 (193)	0.159 (26)	0.123 (34)
19. furnace - stove	0.876	0.393 (64)	0.135 (91)	0.102 (104)
20. seafood - lobster	0.867	0.535 (16)	0.157 (28)	0.110 (71)
21. mile - kilometer	0.863	0.378 (72)	0.142 (59)	0.158 (13)
22. Maradona - football	0.859	0.423 (43)	0.083 (334)	0.078 (243)
23. OPEC - oil	0.856	0.410 (49)	0.211 (6)	0.160 (10)
24. king - queen	0.855	0.417 (45)	0.139 (69)	0.111 (69)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
25. murder - manslaughter	0.850	0.430 (39)	0.253 (4)	0.209 (5)
26. computer - software	0.846	0.554 (14)	0.154 (36)	0.120 (37)
26. money - bank	0.846	0.344 (98)	0.135 (93)	0.116 (50)
28. vodka - gin	0.842	0.572 (11)	0.149 (44)	0.087 (198)
28. Jerusalem - Israel	0.842	0.489 (20)	0.186 (9)	0.144 (16)
30. planet - star	0.841	0.414 (46)	0.141 (64)	0.118 (42)
31. calculation - computation	0.840	0.455 (29)	0.134 (98)	0.075 (258)
32. money - dollar	0.838	0.308 (149)	0.102 (288)	0.051 (337)
33. law - lawyer	0.834	0.350 (93)	0.181 (13)	0.138 (20)
34. championship - tournament	0.832	0.571 (12)	0.154 (32)	0.116 (48)
35. weather - forecast	0.830	0.440 (32)	0.388 (2)	0.248 (3)
35. seafood - food	0.830	0.271 (204)	0.129 (131)	0.093 (158)
38. nature - environment	0.827	0.321 (129)	0.097 (313)	0.104 (96)
38. network - hardware	0.827	0.286 (181)	0.133 (106)	0.105 (88)
38. FBI - investigation	0.827	0.495 (19)	0.144 (53)	0.097 (125)
40. man - woman	0.826	0.429 (40)	0.155 (31)	0.118 (45)
41. money - wealth	0.823	0.294 (168)	0.136 (84)	0.119 (39)
42. psychology - Freud	0.817	0.436 (34)	0.120 (189)	0.112 (62)
43. news - report	0.812	0.228 (264)	0.112 (239)	0.072 (272)
45. Harvard - Yale	0.809	0.482 (22)	0.174 (15)	0.126 (28)
45. war - troops	0.809	0.393 (63)	0.140 (68)	0.098 (123)
45. vodka - brandy	0.809	0.488 (21)	0.101 (300)	0.055 (333)
47. physics - proton	0.808	0.341 (104)	0.112 (237)	0.094 (151)
47. bank - money	0.808	0.344 (98)	0.135 (93)	0.116 (50)
49. planet - galaxy	0.807	0.301 (161)	0.127 (143)	0.095 (137)
51. planet - moon	0.803	0.375 (73)	0.137 (79)	0.089 (183)
51. psychology - psychiatry	0.803	0.473 (24)	0.176 (14)	0.143 (18)
51. stock - market	0.803	0.404 (53)	0.147 (45)	0.083 (223)
53. credit - card	0.801	0.499 (18)	0.201 (7)	0.184 (7)
53. planet - constellation	0.801	0.245 (244)	0.138 (78)	0.104 (94)
55. hotel - reservation	0.798	0.373 (79)	0.101 (299)	0.071 (277)
56. planet - sun	0.797	0.290 (171)	0.139 (71)	0.104 (91)
58. closet - clothes	0.795	0.411 (48)	0.170 (19)	0.113 (60)
58. tiger - feline	0.795	0.300 (163)	0.121 (186)	0.096 (132)
58. tiger - jaguar	0.795	0.375 (75)	0.125 (157)	0.077 (245)
60. soap - opera	0.789	0.450 (31)	0.099 (308)	0.070 (285)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
60. planet - astronomer	0.789	0.331 (118)	0.142 (60)	0.082 (232)
62. planet - space	0.787	0.331 (120)	0.144 (54)	0.112 (63)
62. movie - theater	0.787	0.397 (59)	0.124 (160)	0.100 (111)
64. treatment - recovery	0.786	0.332 (117)	0.107 (268)	0.109 (75)
65. liquid - water	0.784	0.318 (133)	0.124 (158)	0.092 (162)
66. life - death	0.783	0.330 (122)	0.133 (101)	0.091 (171)
67. baby - mother	0.780	0.400 (57)	0.106 (273)	0.063 (307)
68. aluminum - metal	0.778	0.403 (54)	0.137 (80)	0.089 (184)
69. lobster - food	0.776	0.210 (282)	0.110 (250)	0.081 (233)
69. cell - phone	0.776	0.251 (229)	0.169 (20)	0.117 (46)
71. dollar - yen	0.773	0.433 (37)	0.128 (133)	0.033 (351)
72. money - deposit	0.768	0.249 (234)	0.106 (275)	0.106 (85)
72. wood - forest	0.768	0.379 (70)	0.140 (66)	0.112 (67)
74. television - film	0.767	1.000 (1)	0.153 (39)	0.131 (25)
76. admission - ticket	0.764	0.331 (119)	0.131 (118)	0.083 (228)
76. game - team	0.764	0.371 (81)	0.114 (223)	0.059 (322)
76. psychology - mind	0.764	0.339 (106)	0.140 (65)	0.123 (33)
78. Arafat - terror	0.759	0.381 (67)	0.102 (292)	0.092 (166)
78. Jerusalem - Palestinian	0.759	0.591 (9)	0.101 (294)	0.063 (308)
80. profit - loss	0.757	0.269 (209)	0.146 (46)	0.131 (24)
80. dividend - payment	0.757	0.286 (180)	0.069 (343)	0.096 (131)
82. computer - keyboard	0.756	0.294 (169)	0.131 (115)	0.119 (40)
83. boxing - round	0.755	0.289 (174)	0.116 (212)	0.090 (182)
84. rock - jazz	0.753	0.359 (85)	0.142 (61)	0.100 (112)
84. century - year	0.753	0.220 (271)	0.119 (195)	0.084 (217)
86. computer - internet	0.752	0.362 (84)	0.133 (102)	0.108 (77)
87. money - property	0.751	0.270 (206)	0.116 (208)	0.082 (231)
88. announcement - news	0.750	0.147 (334)	0.116 (207)	0.075 (257)
88. tennis - racket	0.750	0.338 (109)	0.135 (92)	0.110 (73)
90. day - dawn	0.747	0.187 (310)	0.131 (111)	0.093 (155)
90. canyon - landscape	0.747	0.300 (164)	0.116 (210)	0.095 (136)
92. food - fruit	0.746	0.303 (158)	0.132 (108)	0.108 (79)
93. currency - market	0.744	0.310 (147)	0.101 (295)	0.067 (298)
93. telephone - communication	0.744	0.290 (173)	0.133 (105)	0.098 (124)
95. psychology - cognition	0.742	0.477 (23)	0.174 (16)	0.133 (23)
96. seafood - sea	0.741	0.276 (194)	0.104 (281)	0.048 (341)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
96. marathon - sprint	0.741	0.353 (90)	0.100 (307)	0.066 (300)
98. book - library	0.740	0.247 (239)	0.119 (194)	0.094 (153)
98. book - paper	0.740	0.237 (252)	0.127 (145)	0.091 (169)
100. Mexico - Brazil	0.738	0.521 (17)	0.171 (18)	0.158 (12)
102. jaguar - cat	0.736	0.315 (138)	0.109 (259)	0.067 (297)
102. psychology - depression	0.736	0.433 (38)	0.000 (353)	0.086 (207)
102. media - radio	0.736	0.318 (134)	0.129 (129)	0.102 (103)
104. fighting - defeating	0.735	0.334 (114)	0.129 (127)	0.099 (115)
106. dollar - profit	0.732	0.226 (265)	0.097 (315)	0.039 (347)
106. hundred - percent	0.732	0.328 (123)	0.107 (270)	0.072 (275)
106. bird - crane	0.732	0.315 (140)	0.145 (50)	0.089 (185)
106. movie - star	0.732	0.375 (74)	0.116 (211)	0.087 (203)
109. physics - chemistry	0.729	0.573 (10)	0.186 (10)	0.166 (9)
109. tiger - cat	0.729	0.372 (80)	0.129 (126)	0.097 (129)
111. country - citizen	0.725	0.234 (256)	0.127 (142)	0.086 (206)
112. money - possession	0.723	0.224 (267)	0.123 (168)	0.097 (128)
113. jaguar - car	0.721	0.302 (159)	0.132 (109)	0.104 (95)
114. cup - drink	0.719	0.398 (58)	0.114 (227)	0.090 (180)
115. psychology - health	0.716	0.233 (257)	0.129 (125)	0.100 (110)
116. museum - theater	0.712	0.343 (102)	0.122 (175)	0.087 (199)
117. summer - drought	0.709	0.208 (286)	0.102 (291)	0.068 (295)
118. investor - earning	0.706	0.222 (268)	0.128 (136)	0.097 (126)
118. phone - equipment	0.706	0.185 (312)	0.106 (276)	0.078 (242)
120. bird - cock	0.703	0.266 (213)	0.109 (261)	0.065 (302)
121. tiger - carnivore	0.701	0.315 (137)	0.126 (147)	0.093 (154)
121. company - stock	0.701	0.297 (166)	0.112 (238)	0.058 (328)
124. game - victory	0.696	0.314 (143)	0.106 (274)	0.061 (315)
124. liability - insurance	0.696	0.374 (77)	0.143 (55)	0.233 (4)
124. stroke - hospital	0.696	0.355 (89)	0.117 (201)	0.091 (173)
127. psychology - anxiety	0.693	0.457 (28)	0.006 (352)	0.093 (159)
127. tiger - animal	0.693	0.325 (126)	0.131 (117)	0.108 (80)
127. doctor - nurse	0.693	0.403 (56)	0.124 (165)	0.101 (109)
129. game - defeat	0.690	0.306 (152)	0.105 (280)	0.062 (313)
130. FBI - fingerprint	0.687	0.435 (36)	0.138 (77)	0.091 (170)
132. opera - performance	0.681	0.240 (249)	0.122 (177)	0.103 (100)
132. street - block	0.681	0.232 (259)	0.125 (154)	0.093 (157)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
132. money - withdrawal	0.681	0.189 (306)	0.067 (345)	0.083 (222)
134. drink - eat	0.680	0.471 (26)	0.174 (17)	0.135 (22)
136. cup - tableware	0.678	0.254 (226)	0.100 (301)	0.067 (296)
136. psychology - fear	0.678	0.321 (131)	0.105 (279)	0.076 (252)
136. tiger - mammal	0.678	0.347 (96)	0.131 (112)	0.091 (177)
136. drug - abuse	0.678	0.453 (30)	0.184 (11)	0.156 (15)
140. concert - virtuoso	0.673	0.264 (217)	0.099 (309)	0.062 (311)
140. football - basketball	0.673	0.640 (6)	0.127 (140)	0.185 (6)
140. student - professor	0.673	0.344 (97)	0.134 (97)	0.088 (194)
142. computer - laboratory	0.670	0.272 (201)	0.151 (43)	0.139 (19)
143. television - radio	0.669	0.709 (3)	0.163 (24)	0.129 (27)
143. love - sex	0.669	0.307 (150)	0.117 (206)	0.080 (237)
145. problem - challenge	0.667	0.332 (116)	0.110 (248)	0.059 (321)
146. movie - critic	0.665	0.424 (42)	0.133 (103)	0.070 (284)
146. Arafat - peace	0.665	0.378 (71)	0.086 (331)	0.065 (303)
148. bed - closet	0.664	0.338 (110)	0.123 (167)	0.093 (160)
149. psychology - science	0.663	0.325 (124)	0.151 (42)	0.116 (49)
151. lawyer - evidence	0.661	0.325 (125)	0.122 (184)	0.095 (140)
151. bishop - rabbi	0.661	0.288 (178)	0.123 (173)	0.081 (234)
151. fertility - egg	0.661	0.406 (50)	0.130 (122)	0.100 (113)
153. precedent - law	0.657	0.284 (184)	0.123 (171)	0.083 (226)
154. minister - party	0.655	0.321 (130)	0.103 (285)	0.068 (291)
154. football - tennis	0.655	0.600 (8)	0.082 (335)	0.115 (54)
156. professor - doctor	0.654	0.339 (108)	0.125 (152)	0.086 (211)
157. cup - coffee	0.650	0.435 (35)	0.124 (159)	0.112 (64)
157. psychology - clinic	0.650	0.311 (145)	0.129 (130)	0.113 (59)
159. government - crisis	0.648	0.351 (92)	0.126 (150)	0.099 (116)
159. water - seepage	0.648	0.239 (250)	0.121 (187)	0.103 (98)
161. space - world	0.645	0.215 (277)	0.135 (95)	0.098 (122)
162. Japanese - American	0.642	0.246 (242)	0.113 (233)	0.084 (218)
163. dividend - calculation	0.640	0.297 (167)	0.089 (326)	0.050 (338)
164. luxury - car	0.639	0.304 (153)	0.128 (137)	0.095 (142)
164. victim - emergency	0.639	0.267 (211)	0.138 (74)	0.118 (41)
166. tool - implement	0.638	0.287 (179)	0.109 (264)	0.073 (263)
167. street - place	0.636	0.251 (231)	0.156 (29)	0.124 (32)
167. competition - price	0.636	0.226 (266)	0.105 (278)	0.064 (305)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
169. psychology - doctor	0.634	0.280 (188)	0.102 (293)	0.073 (268)
170. gender - equality	0.633	0.464 (27)	0.197 (8)	0.143 (17)
171. listing - category	0.629	0.201 (294)	0.129 (128)	0.101 (107)
173. governor - office	0.625	0.248 (237)	0.169 (21)	0.116 (47)
173. video - archive	0.625	0.199 (296)	0.119 (197)	0.077 (249)
173. oil - stock	0.625	0.325 (127)	0.124 (162)	0.096 (134)
173. discovery - space	0.625	0.274 (198)	0.142 (62)	0.098 (119)
177. shower - thunderstorm	0.622	0.289 (176)	0.093 (321)	0.049 (339)
177. record - number	0.622	0.276 (195)	0.131 (110)	0.086 (209)
177. train - car	0.622	0.318 (132)	0.115 (216)	0.073 (265)
179. brother - monk	0.618	0.286 (182)	0.130 (120)	0.094 (146)
181. disaster - area	0.616	0.204 (289)	0.100 (304)	0.089 (193)
181. family - planning	0.616	0.216 (275)	0.143 (57)	0.112 (68)
181. production - crew	0.616	0.368 (82)	0.137 (82)	0.108 (78)
181. nature - man	0.616	0.314 (142)	0.100 (303)	0.101 (106)
184. food - preparation	0.613	0.197 (297)	0.104 (282)	0.058 (326)
184. skin - eye	0.613	0.394 (62)	0.138 (75)	0.114 (56)
188. preservation - world	0.610	0.108 (345)	0.126 (146)	0.090 (179)
188. lover - quarrel	0.610	0.275 (197)	0.091 (325)	0.025 (352)
188. game - series	0.610	0.552 (15)	0.096 (318)	0.061 (314)
188. movie - popcorn	0.610	0.261 (218)	0.111 (245)	0.069 (286)
188. bread - butter	0.610	0.554 (13)	0.146 (48)	0.077 (247)
191. dollar - loss	0.600	0.270 (205)	0.029 (351)	0.048 (340)
192. weapon - secret	0.597	0.396 (61)	0.135 (96)	0.085 (212)
193. precedent - antecedent	0.595	0.336 (112)	0.129 (124)	0.098 (121)
194. shower - flood	0.594	0.221 (270)	0.091 (324)	0.057 (330)
196. announcement - warning	0.591	0.251 (230)	0.124 (164)	0.088 (196)
196. arrival - hotel	0.591	0.342 (103)	0.088 (330)	0.059 (323)
196. registration - arrangement	0.591	0.117 (341)	0.096 (319)	0.084 (221)
198. game - round	0.588	0.333 (115)	0.102 (287)	0.057 (329)
198. baseball - season	0.588	0.420 (44)	0.110 (251)	0.072 (274)
200. drink - mouth	0.586	0.374 (76)	0.128 (134)	0.079 (240)
202. life - lesson	0.584	0.196 (298)	0.117 (204)	0.075 (256)
202. grocery - money	0.584	0.357 (87)	0.111 (246)	0.086 (204)
202. energy - crisis	0.584	0.311 (146)	0.123 (172)	0.101 (105)
204. king - rook	0.582	0.216 (274)	0.146 (49)	0.103 (99)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
204. cucumber - potato	0.582	0.473 (25)	0.141 (63)	0.095 (141)
206. equipment - maker	0.581	0.210 (283)	0.107 (267)	0.073 (270)
206. reason - criterion	0.581	0.240 (248)	0.121 (188)	0.084 (220)
208. cup - liquid	0.580	0.312 (144)	0.110 (256)	0.098 (118)
209. deployment - withdrawal	0.578	0.307 (151)	0.068 (344)	0.063 (309)
210. tiger - zoo	0.577	0.437 (33)	0.122 (179)	0.083 (227)
211. precedent - example	0.575	0.247 (241)	0.121 (185)	0.086 (205)
211. journey - car	0.575	0.236 (254)	0.114 (228)	0.083 (225)
213. smart - stupid	0.571	0.357 (86)	0.114 (224)	0.061 (316)
214. plane - car	0.567	0.266 (214)	0.108 (266)	0.071 (281)
215. planet - people	0.565	0.315 (141)	0.153 (40)	0.129 (26)
216. lobster - wine	0.560	0.301 (162)	0.115 (217)	0.092 (167)
217. money - laundering	0.555	0.284 (185)	0.339 (3)	0.330 (2)
219. summer - nature	0.553	0.202 (291)	0.102 (289)	0.099 (114)
219. OPEC - country	0.553	0.171 (320)	0.089 (327)	0.060 (318)
219. decoration - valor	0.553	0.339 (107)	0.126 (149)	0.086 (208)
219. Mars - scientist	0.553	0.403 (55)	0.138 (76)	0.102 (102)
222. tiger - fauna	0.552	0.303 (155)	0.124 (166)	0.073 (266)
223. psychology - discipline	0.548	0.363 (83)	0.117 (202)	0.097 (127)
224. glass - metal	0.546	0.405 (51)	0.123 (170)	0.090 (181)
225. alcohol - chemistry	0.544	0.250 (233)	0.098 (312)	0.092 (165)
226. disability - death	0.536	0.209 (285)	0.122 (178)	0.106 (82)
227. change - attitude	0.533	0.229 (263)	0.110 (252)	0.113 (58)
228. arrangement - accommodation	0.530	0.182 (314)	0.080 (337)	0.074 (260)
229. territory - surface	0.523	0.248 (236)	0.113 (230)	0.084 (216)
231. exhibit - memorabilia	0.520	0.230 (262)	0.124 (161)	0.110 (72)
231. size - prominence	0.520	0.173 (318)	0.101 (298)	0.060 (320)
231. credit - information	0.520	0.150 (332)	0.116 (215)	0.082 (230)
233. territory - kilometer	0.517	0.189 (305)	0.077 (340)	0.045 (342)
234. man - governor	0.514	0.216 (273)	0.120 (192)	0.094 (147)
234. death - row	0.514	0.266 (212)	0.152 (41)	0.095 (135)
236. doctor - liability	0.508	0.160 (328)	0.101 (296)	0.070 (282)
237. impartiality - interest	0.505	0.205 (288)	0.112 (242)	0.079 (238)
238. energy - laboratory	0.497	0.276 (196)	0.127 (139)	0.094 (149)
239. secretary - senate	0.494	0.428 (41)	0.153 (37)	0.125 (29)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
240. death - inmate	0.491	0.214 (278)	0.114 (222)	0.068 (290)
242. travel - activity	0.488	0.214 (279)	0.123 (169)	0.091 (176)
242. doctor - personnel	0.488	0.205 (287)	0.088 (328)	0.068 (293)
242. cup - food	0.488	0.274 (199)	0.110 (254)	0.089 (188)
242. monk - oracle	0.488	0.152 (330)	0.061 (346)	0.060 (319)
245. journal - association	0.485	0.396 (60)	0.130 (121)	0.091 (174)
246. street - children	0.482	0.249 (235)	0.128 (135)	0.084 (219)
246. car - flight	0.482	0.315 (139)	0.116 (209)	0.090 (178)
248. space - chemistry	0.476	0.265 (216)	0.127 (144)	0.110 (70)
249. situation - conclusion	0.469	0.389 (65)	0.100 (306)	0.060 (317)
250. tiger - organism	0.465	0.166 (324)	0.107 (269)	0.062 (310)
252. consumer - energy	0.463	0.303 (156)	0.143 (58)	0.114 (55)
252. word - similarity	0.463	0.235 (255)	0.103 (284)	0.068 (294)
252. peace - plan	0.463	0.232 (258)	0.130 (123)	0.108 (76)
254. ministry - culture	0.456	0.261 (219)	0.156 (30)	0.115 (53)
255. hospital - infrastructure	0.450	0.159 (329)	0.101 (297)	0.078 (244)
256. smart - student	0.449	0.189 (307)	0.131 (116)	0.105 (86)
257. investigation - effort	0.446	0.304 (154)	0.116 (214)	0.072 (273)
258. image - surface	0.443	0.280 (189)	0.110 (258)	0.066 (299)
259. life - term	0.437	0.255 (224)	0.145 (51)	0.099 (117)
261. start - match	0.434	0.237 (253)	0.126 (151)	0.087 (201)
261. board - recommendation	0.434	0.164 (326)	0.122 (182)	0.085 (214)
261. computer - news	0.434	0.110 (344)	0.110 (255)	0.073 (267)
263. lad - brother	0.433	0.250 (232)	0.118 (198)	0.069 (287)
264. food - rooster	0.429	0.150 (331)	0.110 (253)	0.072 (271)
265. observation - architecture	0.425	0.232 (260)	0.126 (148)	0.072 (276)
265. coast - hill	0.425	0.317 (135)	0.161 (25)	0.118 (43)
268. benchmark - index	0.411	0.190 (301)	0.114 (225)	0.076 (253)
268. deployment - departure	0.411	0.168 (321)	0.109 (262)	0.077 (246)
268. attempt - peace	0.411	0.373 (78)	0.095 (320)	0.036 (348)
270. consumer - confidence	0.399	0.268 (210)	0.182 (12)	0.158 (14)
271. focus - life	0.392	0.247 (240)	0.133 (104)	0.094 (145)
271. start - year	0.392	0.273 (200)	0.122 (180)	0.092 (163)
273. development - issue	0.383	0.265 (215)	0.129 (132)	0.093 (156)
274. day - summer	0.380	0.256 (222)	0.136 (86)	0.095 (143)
275. theater - history	0.377	0.212 (281)	0.118 (200)	0.096 (133)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
277. chance - credibility	0.374	0.259 (221)	0.107 (271)	0.079 (241)
277. profit - warning	0.374	0.174 (317)	0.123 (174)	0.098 (120)
277. media - trading	0.374	0.201 (293)	0.117 (203)	0.089 (187)
277. situation - isolation	0.374	0.293 (170)	0.098 (310)	0.042 (346)
280. precedent - information	0.371	0.084 (348)	0.109 (263)	0.076 (254)
281. architecture - century	0.363	0.289 (175)	0.158 (27)	0.112 (61)
282. population - development	0.360	0.271 (203)	0.140 (67)	0.109 (74)
283. stock - live	0.358	0.190 (302)	0.112 (240)	0.064 (304)
286. morality - marriage	0.354	0.340 (105)	0.039 (350)	0.092 (164)
286. peace - atmosphere	0.354	0.231 (261)	0.113 (236)	0.062 (312)
286. minority - peace	0.354	0.301 (160)	0.134 (99)	0.115 (52)
286. atmosphere - landscape	0.354	0.283 (186)	0.116 (213)	0.069 (288)
286. cup - object	0.354	0.186 (311)	0.076 (341)	0.055 (334)
289. music - project	0.348	0.638 (7)	0.139 (72)	0.104 (93)
289. report - gain	0.348	0.188 (309)	0.125 (153)	0.094 (150)
291. seven - series	0.341	0.405 (52)	0.120 (190)	0.095 (139)
292. experience - music	0.332	0.702 (4)	0.139 (70)	0.103 (97)
293. school - center	0.329	0.285 (183)	0.130 (119)	0.092 (168)
294. five - month	0.322	0.322 (128)	0.135 (90)	0.112 (66)
294. announcement - production	0.322	0.190 (303)	0.122 (176)	0.105 (89)
297. morality - importance	0.315	0.309 (148)	0.135 (88)	0.112 (65)
297. delay - news	0.315	0.111 (343)	0.091 (323)	0.055 (332)
297. money - operation	0.315	0.245 (243)	0.128 (138)	0.094 (152)
299. governor - interview	0.309	0.241 (247)	0.119 (196)	0.089 (189)
300. practice - institution	0.303	0.269 (208)	0.132 (107)	0.114 (57)
301. century - nation	0.300	0.350 (94)	0.137 (81)	0.087 (202)
302. coast - forest	0.299	0.330 (121)	0.153 (38)	0.125 (30)
303. shore - woodland	0.292	0.255 (225)	0.154 (35)	0.125 (31)
304. drink - car	0.288	0.278 (192)	0.113 (232)	0.071 (279)
305. prejudice - recognition	0.284	0.279 (190)	0.124 (163)	0.094 (148)
305. president - medal	0.284	0.279 (191)	0.109 (260)	0.083 (229)
307. viewer - serial	0.280	0.247 (238)	0.114 (226)	0.085 (215)
308. peace - insurance	0.277	0.148 (333)	0.040 (349)	0.064 (306)
308. Mars - water	0.277	0.243 (245)	0.115 (221)	0.091 (175)
310. cup - artifact	0.275	0.105 (346)	0.092 (322)	0.044 (345)
311. media - gain	0.271	0.210 (284)	0.115 (219)	0.091 (172)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
312. precedent - cognition	0.264	0.119 (340)	0.125 (155)	0.092 (161)
313. announcement - effort	0.258	0.238 (251)	0.125 (156)	0.089 (186)
314. line - insurance	0.252	0.167 (322)	0.107 (272)	0.070 (283)
314. crane - implement	0.252	0.083 (350)	0.112 (243)	0.088 (195)
316. drink - mother	0.248	0.336 (113)	0.120 (191)	0.085 (213)
317. opera - industry	0.246	0.129 (339)	0.117 (205)	0.089 (192)
318. listing - proximity	0.238	0.137 (337)	0.103 (286)	0.105 (90)
318. volunteer - motto	0.238	0.067 (351)	0.083 (333)	0.054 (335)
320. precedent - collection	0.232	0.131 (338)	0.113 (235)	0.071 (278)
320. Arafat - Jackson	0.232	0.189 (308)	0.051 (348)	0.058 (327)
322. cup - article	0.222	0.221 (269)	0.108 (265)	0.079 (239)
323. problem - airport	0.220	0.213 (280)	0.052 (347)	0.000 (353)
323. sign - recess	0.220	0.115 (342)	0.113 (234)	0.101 (108)
325. reason - hypertension	0.213	0.141 (335)	0.097 (314)	0.071 (280)
326. direction - combination	0.207	0.347 (95)	0.072 (342)	0.068 (292)
327. Wednesday - news	0.204	0.161 (327)	0.085 (332)	0.036 (349)
328. cup - entity	0.197	0.087 (347)	0.077 (338)	0.058 (325)
329. glass - magician	0.189	0.190 (304)	0.100 (305)	0.068 (289)
329. cemetery - woodland	0.189	0.290 (172)	0.145 (52)	0.118 (44)
331. possibility - girl	0.175	0.243 (246)	0.135 (89)	0.075 (259)
332. cup - substance	0.173	0.201 (295)	0.097 (316)	0.077 (250)
333. forest - graveyard	0.166	0.215 (276)	0.111 (244)	0.080 (236)
335. month - hotel	0.162	0.192 (300)	0.077 (339)	0.052 (336)
335. energy - secretary	0.162	0.253 (227)	0.122 (181)	0.095 (138)
335. stock - egg	0.162	0.184 (313)	0.115 (220)	0.088 (197)
337. precedent - group	0.158	0.166 (325)	0.110 (257)	0.059 (324)
338. production - hike	0.156	0.084 (349)	0.103 (283)	0.074 (262)
339. holy - sex	0.142	0.269 (207)	0.102 (290)	0.073 (264)
339. stock - phone	0.142	0.138 (336)	0.110 (249)	0.080 (235)
341. drink - ear	0.111	0.252 (228)	0.120 (193)	0.075 (255)
341. stock - CD	0.111	0.272 (202)	0.111 (247)	0.086 (210)
343. delay - racism	0.098	0.178 (316)	0.112 (241)	0.077 (248)
345. lad - wizard	0.071	0.193 (299)	0.131 (114)	0.089 (190)
345. monk - slave	0.071	0.256 (223)	0.134 (100)	0.096 (130)
345. stock - life	0.071	0.201 (292)	0.118 (199)	0.083 (224)
345. stock - jaguar	0.071	0.203 (290)	0.100 (302)	0.066 (301)

Table 9: Results for the concept pairs.

	Human	NWD (#)	NCD (#)	Jaccard (#)
348. sugar - approach	0.067	0.172 (319)	0.127 (141)	0.105 (87)
349. rooster - voyage	0.040	0.000 (353)	0.113 (231)	0.074 (261)
350. noon - string	0.032	0.166 (323)	0.088 (329)	0.044 (343)
350. chord - smile	0.032	0.219 (272)	0.098 (311)	0.077 (251)
352. professor - cucumber	0.008	0.050 (352)	0.096 (317)	0.044 (344)
353. king - cabbage	0.000	0.181 (315)	0.105 (277)	0.056 (331)

D

NCD RESULTS FOR THE THREE DATA SOURCES

The resulting Spearman scores for the Normalized Compression Distance are divided in three figures. Each figure shows the results for one dataset. Figure 25 shows the results of NCD algorithm on dataset with Google as data source. The results for the dataset with Wikipedia as data source is given in figure 26. The last figure 27 shows the results for the dataset constructed with IMDb as data source.

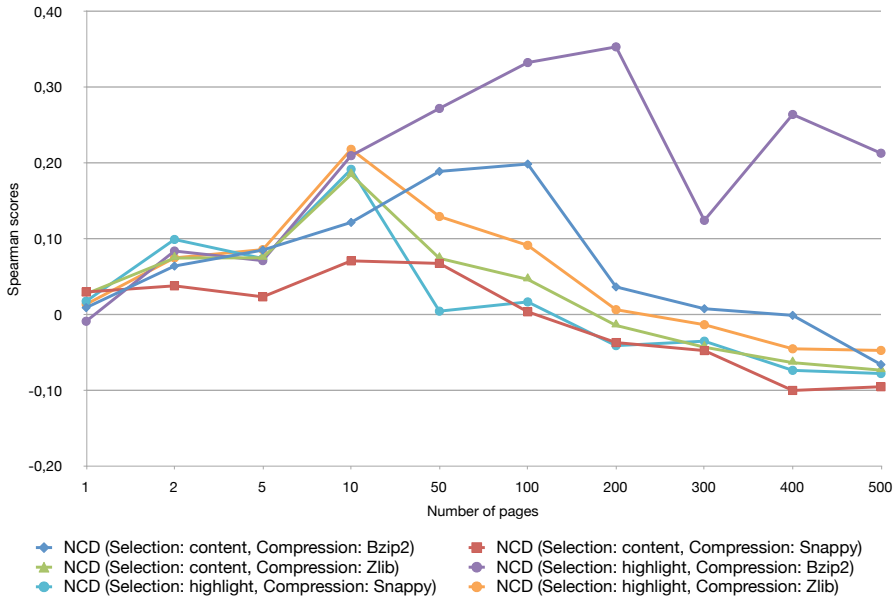

Figure 25: NCD results for the Google data source.

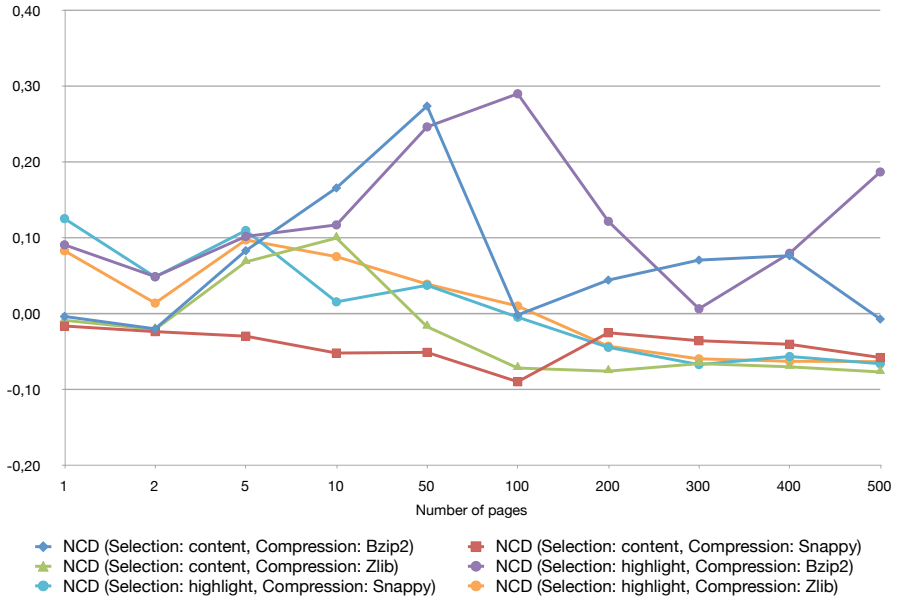

Figure 26: NCD results for the Wikipedia data source.

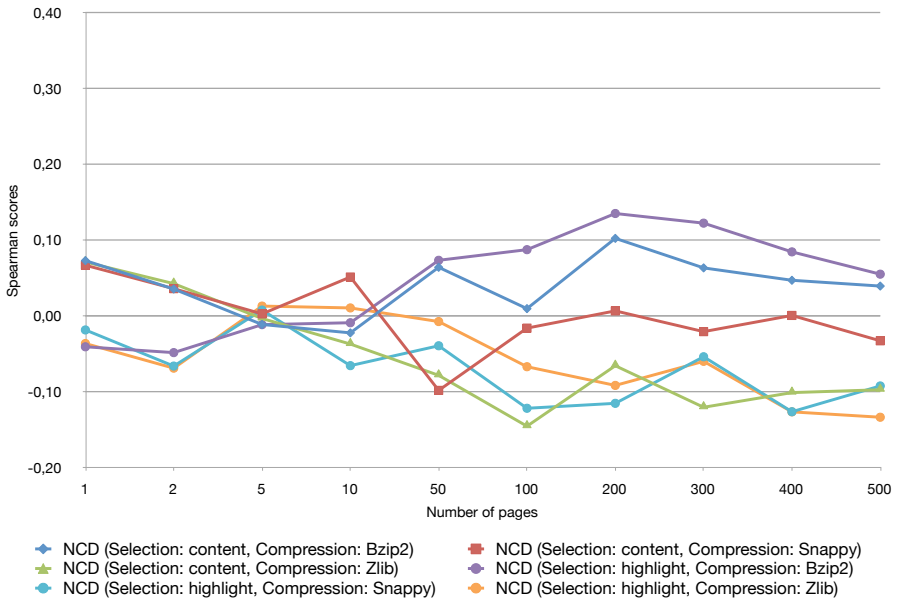

Figure 27: NCD results for the IMDb data source.

E

JACCARD RESULTS FOR THE THREE INPUT DATA SOURCES

The resulting Spearman scores for the Jaccard index on keywords are divided in three figures. Each figure shows the results for one dataset. The figure 28 shows the results the algorithm on the dataset that uses Google as data source. The results for the dataset with Wikipedia as data source is given in figure 29. The last figure 30 shows the results for the dataset constructed with IMDb as data source.

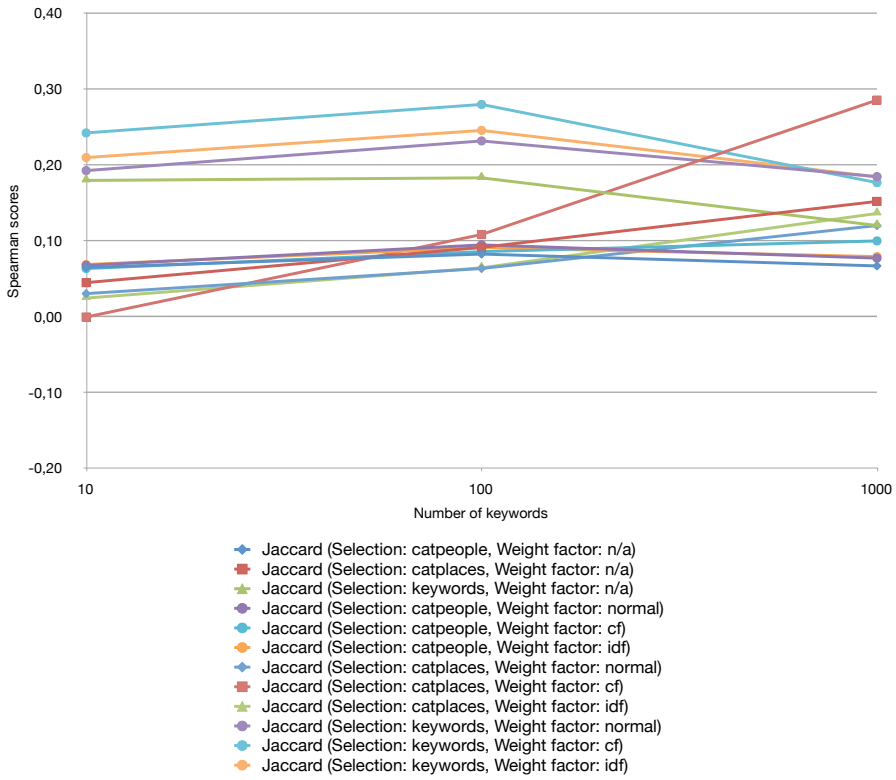

Figure 28: Jaccard results for the Google data source.

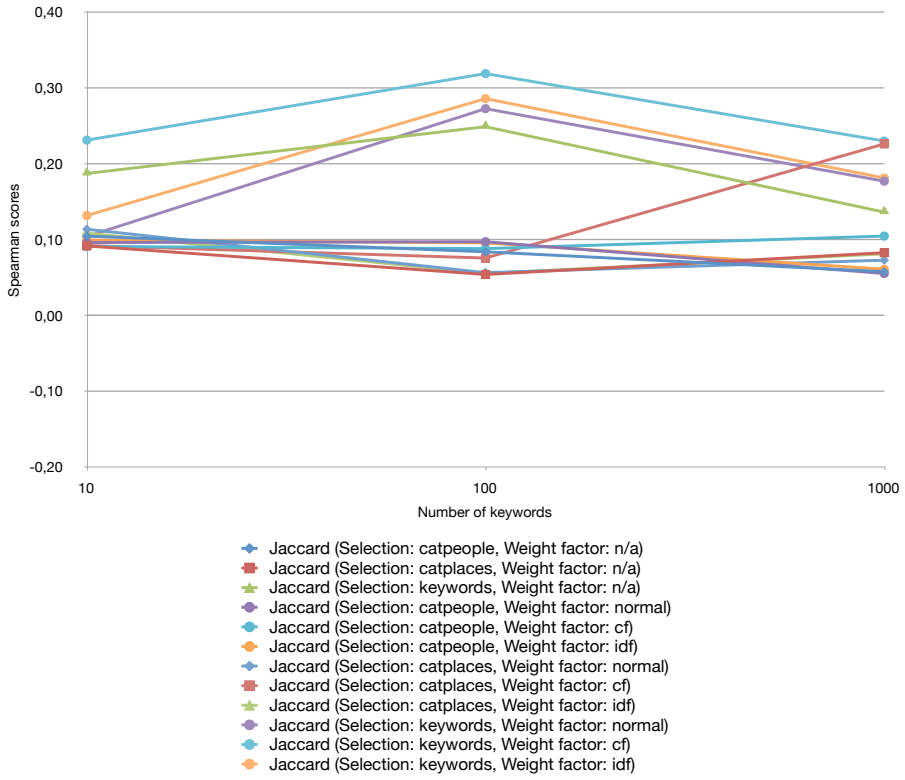

Figure 29: Jaccard results for the Wikipedia data source.

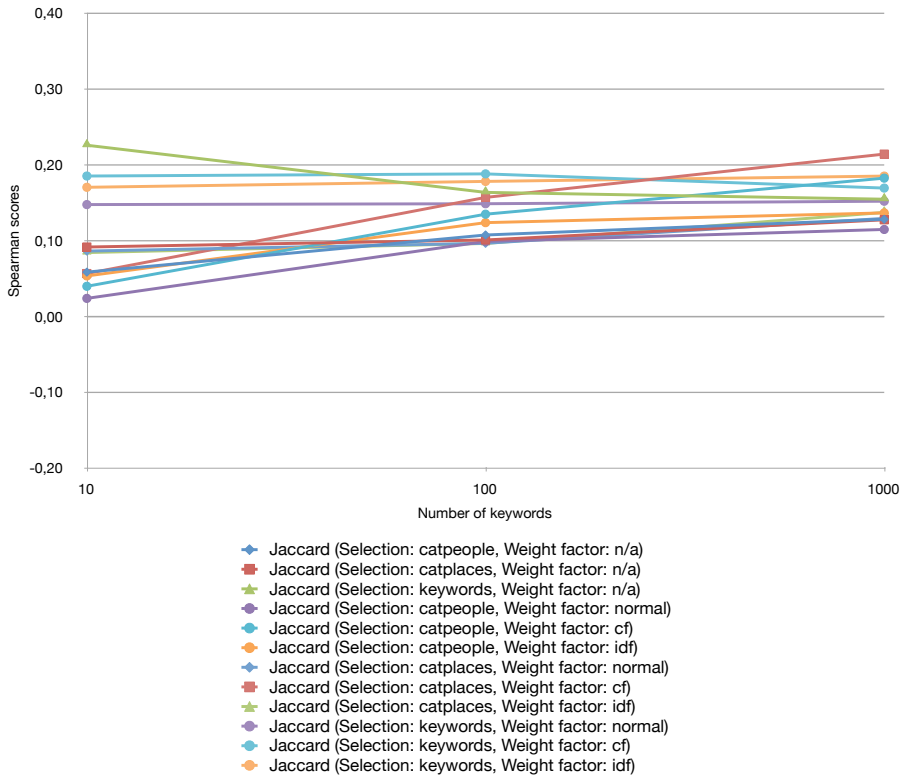

Figure 30: Jaccard results for the IMDb data source.