Proposal for a STS Evaluation Framework for STS based Applications

Studienarbeit

Philip Beyer
Wirtschaftsinformatik

TECHNISCHE UNIVERSITÄT DARMSTADT

UBIQUITOUS KNOWLEDGE PROCESSING

Submission: 06.03.2015

Supervised by: Nils Reimers and Prof. Dr. Iryna Gurevych

Fachgebiet Ubiquitous Knowledge Processing

Fachbereich Informatik

Technische Universität Darmstadt

Hochschulstraße 10

64289 Darmstadt

http://www.ukp.tu-darmstadt.de

Table of Content

List of Figures

List of Tables

List of Abbreviations

AEG	Automatic Essay Grading
CG	Commutative Gain
DCG	Discounted Commutative Gain
IR	Information Retrieval
LTS	Lexical Text Similarity
MAD	Mean Absolute Difference
MSD	Mean Square Difference
nCG	normalized Cumulative Gain
nDCG	normalized Discounted Cumulative Gain
PCC	Pearson product-moment correlation
RTE	Recognizing Textual Entailment
SCC	Spearman's rank correlation coefficient
STS	Semantic Text Similarity

1 Introduction

Semantic Text Similarity (STS) is a concept based on the notion that different text can be similar to each other. The more similar texts are the higher is the value for STS between them. In recent years the number and quality of systems that rate the STS between texts have increased, as has the number of applications where such system can be used. Common examples for STS based applications are Automatic Essay Grading, Plagiarism Detection, Automated Text Summarization, or Link Discovery. These applications are strongly dependent on the quality of the STS measure they use, but provide additional features as well. An Automatic Essay Grading system might check the spelling of an essay and a Plagiarism Detection system has to find a way to separate correctly cited from not cited quotes. In theory, a STS based application can be developed independently from the used STS measure. In practise, however, it is common that the development of a STS based application implies the implementation of a new STS measure. One reason for this might be, that "… only a few text similarity measures proposed in the literature are released publicly, and those then typically do not comply with any standardization.", (Bär et al., 2013). Another reason might be that it is unclear how well an existing STS measure performs within new applications. For practitioners who develop STS based applications, this is an important question, because the integration of STS measures in an application is a complex and time-consuming task. Trying out many different STS measures is not a practical solution. For the selection of the STS measure, practitioners mainly relay on the evaluation done by publishers of a new STS measure. The base of quantitative information about a STS measure can be rather limited and influenced by the setup of the evaluation, but other sources to base the selection of STS measures can be hard to obtain.

To evaluate STS measures it is most common to apply the Pearson product-moment correlation (PCC) to calculate the correlation between given values of similarity, the so called "gold standard", and the result of the STS measure. Despite this common practise, Agirre et al. (2013) states in the discussion of the results of the SemEval 2013 task about STS that: "Evaluation of STS is still an open issue." and that beside the Pearson correlation "…other alternatives need to be considered, depending on the requirements of the target application."

A STS measure that always returns the correct similarity value between two texts has a perfect Pearson correlation value of 1 when comparing calculated similarity values to "correct" gold standard values. Such a measure will always perform better than any other STS measure in a reasonably formulated STS based application. In practise, however, such a

perfect correlation is unlikely to be achieved. A developer of a STS based application is still likely to pick the STS measure with the highest Pearson correlation for his application.

The intention of this work is to show that a STS system with a higher Pearson correlation value will not always outperform a STS system with a lower Pearson correlation when used in a STS based application. In addition, a framework of other evaluation methods is proposed, that can help practitioners to pick a STS measure for their applications. A requirement for the framework to be useful is, that developers of new STS measures publish the result of multiple evaluation methods or that comparative studies for STS measures are conducted.

Besides the difficulties to find a suitable evaluation method, some other issues are unsolved in the context of STS measures. Due to the costly process of creation, not many datasets or corpora are available to test STS systems. The content of existing corpora varies in what is compared between words, phrases, sentence, and paragraphs. The inter annotator agreement for setting the gold standard for corpora, the upper limit on how well a STS system can perform, varies and can be quite low. There is no robust definition what similarity in the context of texts or even words means or how it can be determined. The distribution of similarity values in a corpus mostly does not mirror the distribution of the dataset of actual applications. Although these issues are worse being addressed, they are out of scope for this work.

The content of the remaining work is structured as follows: In chapter 2 an introduction to the field of STS is given and the research question is formulated. Chapter 3 compares the ranking of different STS measures according to different evaluation methods and formulates the STS Evaluation Framework. The framework organises requirements of STS based applications and maps them to evaluation methods. In chapter 4, the framework is evaluated by three experiments. Chapter 5 concludes this work with a summary and suggestions for future work.

2 State of the Art

In this chapter the main concepts related to Semantic Text Similarity (STS) are summarized. After a short introduction of STS, applications that are based to some extend on STS system are described. Then fundamental text similarity methods are listed together with the most common methods to evaluate those systems. This chapter concludes with a formulation of the research question.

2.1 Semantic Text Similarity & Use Cases

To understand Semantic Text Similarity and the difficulties such a concept implies, a deconstruction of the term is helpful. Starting with the last term, *similarity* describes to what degree the characteristics of one thing matches the characteristic of another thing. The second term, *text*, means any combination of words, from a single word to whole encyclopaedia. In combination, *text similarity* describe to what degree the characteristics of two combinations of words match.

In case of single words we call a perfect match a synonym. Miller and Charles (1991) use a formulation they attribute to Leibniz that identifies such a perfect match. “[Two] words are said to be synonyms if one can be used in a statement in place of the other without changing the meaning of the statement (the conditions under which the statement would be true or false).” If word similarity would be dichotomic in such way that a text is either similar or not, this definition would be of sufficient. Unfortunately, word similarity is rather a continuous measure where two terms can be more or less similar. The above formulation does not seem to be expandable to include continuous outcomes. Without another formulation, an objective method to assign the similarity of two words to a value does not exist.

In theory, the above formulation could be expanded from single words to phrases and sentences. However, with an increase of the extent of the text the unambiguousness of finding perfect match seems to fade and clearly the above formulation was never intended to include more than single words.

The first term, *semantic*, is well known as one of three levels of the semiotic, the theory of signs in which Morris (1938) differentiates between syntax, semantic, and pragmatic. In the context of STS, *semantic* means that the meaning of a word is important, not the composition of characters. Lexical Text Similarity (LTS) on the other hand means that the composition of the characters is important which could be attributed to the syntax rather than to the semantic level of the sign theory. It is easy to construct two sentences with a high STS but a

low LTS, simply by replacing all possible terms of the first sentence with synonyms to construct the second sentence. In a realistic environment however, people tend to use the same words to describe the same thing as long as they are similar skilled in using the used language. So the values of LTS can be an indicator for the values of STS, but STS has the stronger claim to capture the "real" similarity of two texts (Hliaoutakis et al., 2006).

The term *Semantic Text Similarity* therefore means to what degree does the meaning of two texts match. Unfortunately no objective definition for this degree of matching exists. Therefore, Semantic Text Similarity can be defined as: two texts are similar in a semantic way to the extent that humans perceive the meaning of two texts to be similar (Bär, 2013 and Dagan et al., 2006 for textual entailment). Hence the only way to evaluate the predictions of STS systems is to test them against human judgment of similarity.

STS has a wide range of applications outside the academic world. Some of the most common applications are introduced in the following sections.

2.1.1 Automatic Essay Grading

Letting a student write an essay on a topic or a question is among the best ways to evaluate his learning success due to the comprehensive nature of an essay. Compared to other assessing methods, like answering multiple choice questions, for an essay you do not only need to remember facts but also have to organize information and come to a conclusion. A downside of essays is the time-consuming evaluation and grading process. Furthermore, the grading can be subjective; two challenges multiple choice questionnaires do not cope with (Valenti et al., 2003).

The aim of Automatic Essay Grading (AEG) systems is to automatically give a score to an essay. This can overcome the before mentioned challenges. It can be more objective because it would not give a score based on the authorship of an essay. And the whole (or most of the) grading process would be done by an algorithm.

A common approach for AEG is to extract from a set of already graded essays features that reflect the score (e.g. the appearance of certain words or the length of a paragraph). Then identify these features in the essays to be graded and compute the similarity between these two sets of features (Ben-Simon and Bennett, 2007). In such an approach, the content could be one of the features used or even then only feature (Foltz et al., 1999). A STS measure can be used to determine the similarity between these texts.

2.1.2 Plagiarism Detection

Zu Eissen and Stein (2006) define plagiarism as the use of someone else information, language, or writing, when done without proper acknowledgment of the original source. A wider definition could include thoughts and conclusions as well.

The aim of plagiarism detection is to find out whether a given text is plagiarised. When the text contains sections that are similar to sections that are found in other texts within a collection of related texts, a plagiarism is likely. This method finds plagiarised information, language, and writing but is likely to struggle with the wider definition of plagiarism, the usage of existing thoughts and conclusions.

Plagiarism detection systems usually work in three steps. At first the collection of texts that contain possible plagiarism is build, for example by a forward or backward search of cited sources. Then the actual similarity check is carried out, where similar passages are collected. And lastly the similar passages are somehow processed to make them comparable to the checked text (Potthast et al., 2012). A STS system can then compare the texts and assess the similarity between them.

2.1.3 Recognizing Textual Entailment

Recognizing Textual Entailment (RTE) is the task to determine if the meaning of one text can be inferred from the meaning of the other text (Tatu and Moldovan, 2005). The content of a summary of a text, for example, should be inferred by the content of the complete text.

Question answering, information extraction, summarization, and machine translation evaluation are examples for applications that employ RTE (Dagan et al., 2006). It can be seen as a specific directed case of semantic similarity.

STS on itself is a not a proficient indicator for RTE. Especially because RTE is a binary task while STS is continuous and RTE is unidirectional while STS is bidirectional. But STS can be used as one of larger set of clues to determine textual entailment.

2.1.4 Link Discovery

“Links between pages are essential for navigation, but most systems require authors to manually identify each link. Authors must identify both the anchor and the target page in order to build a knowledge network.” (Huang et al., 2008) In a dynamic setup, where documents are constantly added, edited or even deleted, keeping up with a link structure requires a lot of work. Link Discovery is the task to automatically identify related texts within

a collection of documents. The target text as well as the anchor text can be of different size. Examples are the linking of one text to the same text in another language or the linking of a word to its definition. Automatic Link Discovery is especially popular within the Wikipedia as well as the Linking Open Data community (Lu et al., 2009 and Hassanzadeh et al., 2009).

Typically, the anchor and the target text are semantically similar; therefore STS can be helpful to identify links. When linking texts of vastly different sizes (e.g. a word to an article) just using STS might not be enough and other clues like the context or an existing link structure can be used. Reducing the STS to just the headlines or certain paragraphs are common technics in this field (He, 2009).

2.1.5 Clustering, Tagging, Classification

Clustering, tagging, as well as classification are used to structure sets of documents; either by splitting them up into clusters, tag every document with one or more terms, or by assigning each document to a class (Aggarwal and Zhai, 2012, Davide and Fabio, 2012, and Rubin et al., 2012). Common methods, like the k-means algorithm, involve finding homogeneous sets of documents. One way to define homogeneity in this context is in terms of STS. Therefore STS systems can play be a crucial part in determining how to cluster, tag, or classify a document.

2.1.6 Automated Text Summarization

Barzilay and Elhadad (1999) define summarization as “the process of condensing a source text into a shorter version preserving its information content.” Automated Text Summarization aims to automatically create a summary of a text. Building upon the early work of Jones (1999) and Hovy and Lin (1998), Nenkova et al. (2011) distinguishes between *extractive* and *abstractive* summaries, *single* versus *multi-document* summaries, *indicative* and *informative* summaries, *paragraph, keyword, or headline* summaries, and *generic, query focused*, as well as *update* summaries.

STS can be used for automatic text summarization in different ways. For example can a high STS score of one sentence to either all other sentences individually or to the complete text as a whole be a good indicator that this sentence is “central” and therefore is a good pick to put in an extractive summary. Another possibility is to compare all candidates that are selected to be part of the summary to each other. Sentence with a high STS might not add much content to the summary and therefore could be excluded (Gupta and Lehal, 2010). STS can be helpful when it comes to the evaluation of Automated Text Summarization as well. Lin (2004)

proposes ROUGE (Recall-Oriented Understudy for Gisting Evaluation), a tool that measures the text similarity between summaries to evaluate their quality.

2.1.7 Information Retrieval

According to Salton and McGill (1983) Information Retrieval is concerned with all the activities related to the organization of, processing of, and access to information of all forms and formats. Since the rise of the World Wide Web as well as personal computers, the definition got more focused on large collections of digital documents. For Manning et al. (2008), Information Retrieval is "*finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)*." Many of the before mentioned applications fall at least under the first definition and consequently STS can be used in many different ways.

2.2 Common Text Similarity Measures

This section describes common text similarity measures in a simplified way. The list of similarity measures described here is based on the selection by Bär (2013). Most of the listed text similarity measures are based on lexical similarity and only a few implement a way to extend the similarity to semantic similarity. Some of the measures use pre-processing steps such as *tokenization*, *sentence splitting*, *lemmatization*, or *stopword filtering* (Bär, 2013).

2.2.1 String Distance Measures

String Distance Measures have in common that they are applied to two strings and compute the similarity of these strings by the common appearance and order of characters. This makes them useful not only to compare human written prose text but also for DNA sequences and programming code.

2.2.1.1 Greedy String Tiling

Wise (1996) divides the length of shared substrings of two texts in relation to the total text length. This implementation is used by YAP3, a tool to detect plagiarism in programming code of student submissions. The shared substrings should be of maximal length, meaning that finding a long common substring is preferred against finding multiple shorter once. The substring also has a minimal size, typically three. YAP3 also applies some pre-processing that is meant to reduce the code to its actual functionality. For a text similarity measure, these steps, like deleting all words that are not used by the programming language itself, seem less useful.

2.2.1.2 Longest Common Subsequence and Longest Common Substring

Allison and Dix (1986) propose an algorithm to calculate edit operations with a particularly good time complexity for generic tasks. The Longest Common Subsequence refers to the longest string two texts have in common, when gaps between the series in characters are allowed.

Gusfield (1997) proposes an algorithm to determine the length of the Longest Common Substring in linear time complexity with the usage of suffix trees. The Longest Common Substring, unlike the Longest Common Subsequence, does not allow for gaps in between two strings.

2.2.2 N-Gram Models

N-Gram Models count either characters (Kešelj et al., 2003) or words (Lyon et al., 2001) of texts and weight terms with the tf-idf[1] scheme. The n-gram indicates that not every single character or word is used by itself, but a combination of n. A common value of n is three, which would lead to triplets of characters or words. The n-grams are transformed into a vector, called a feature vector, and two of those vectors can then be compared to by calculating the cosine between them. Another possibility is comparing the feature vectors using the Jaccard coefficient (Real and Vargas, 1996 and Huang, 2008) or any other set based similarity measurement.

2.2.3 Compositional Measures

Bär (2013) defines Compositional Measures as a method that tokenizes the input texts, computes pairwise word similarity between all words, and aggregates the resulting scores to an overall similarity score. The method to compute the pairwise word similarities can be one of the before mentioned text similarity measures that work well with single words. Bär refers to a measure by Mihalcea et al. (2006), which he has also implemented in an open source framework (DKPro) for STS measures, as an example. It depends on the text similarity method that is used whether the Compositional Measure is lexical or semantic.

2.3 Evaluation of STS Measures

The main topic of this work is the evaluation of STS measures. The following section focuses on the different ways to evaluate such measures. The difference between intrinsic and

[1] Tf-idf stands for *term frequency* and *inverse document frequency*. Term frequency is the count how often a term appears in the document at hand, while inverse document frequency is the inverse of the count in how many documents of a corpus this term appears.

extrinsic evaluation will be discussed with a focus on common intrinsic evaluation methods. Common datasets (or corpora) that are used to evaluate STS measures are described as well.

2.3.1 Intrinsic vs. Extrinsic Evaluation

A measure or a system, in this case a STS measure, can be evaluated in two different ways according to Galliers and Jones (1993): *intrinsically* or *extrinsically*. *Intrinsic* refers to an evaluation where the quality of a systems output is of concern, its "objective". In case of a STS measure, how well can the text similarity be measured is the intrinsic question. *Extrinsic* evaluation is concerned with how well a larger setup that uses a system performs, so the "function" of a system is evaluated. In STS terms that is how well an application performs that uses a STS measure. The applications can be one of the STS based applications that were discussed in section 2.1. The extrinsic question would evaluate the quality of a summary, grades of an essay, or the finding of links between texts. Typically the intrinsic evaluation is easier to conduct, because only the STS measure and a dataset are needed. For the extrinsic evaluation, a STS based application is needed as well. STS based applications also might incorporate other sub-systems then a STS measure which also have an impact on the overall performance. It might be difficult to differentiate between the performance of the STS measure and the rest of the application. But in the end a STS measure is developed to work well in an application and a measure that only works well in an isolated setting (intrinsic evaluation) might not be that useful. The measurement used to evaluate a STS measure extrinsically depends on the application that makes use of the STS measure. For the intrinsic evaluation a few measures have been established that are regularly used and will be discussed in the following section.

2.3.2 Common Intrinsic Evaluation Methods

In the following section the most common intrinsic methods to evaluate STS measures are described. These measures fall under three categories: correlations, binary classification, and cumulative gains. This list is not intended to be exhaustive but to give an overview of the measures that are commonly used and will be used within this work.

2.3.2.1 Correlation

Correlations describe the relationships between two (or more) continuous variables. For the purpose of evaluating STS measures, the two variables would be the "gold standard" similarity score and the score the STS measure produces. The idea behind this approach is the higher the correlation coefficient, the better the STS measure. The two most common

correlations used for the evaluation of STS measures are the Pearson product-moment correlation coefficient (PCC) and the Spearman's rank correlation coefficient, of which the PCC seems to be the dominate evaluation metric overall (Agirre et al., 2013, Eneko Agirre, In prep., and Zesch, 2010).

The PCC measures the linear correlation between two variables. In other words how closely related is one vector of variables to another vector, when a linear function is applied to one of the two vectors. The correlation value does not make any statement about the linear function. Simplified a correlation value of 1 (the maximum) means that for high values of one vector, the related values of the other vector are high as well and the same for low values. A correlation value of 0 means that, the height of the value of one vector leads to no conclusion about the height of the related value of the other vector. A correlation value of -1 (the minimum) means that, a high value of one vector relates to a low value of the other vector. Typically the PCC of two vectors $\vec{x}$ and $\vec{y}$ with length n is denoted with a r (rho).

$$r = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n(\sum x_i^2)(\sum x_i)^2}\sqrt{n(\sum y_i^2)(\sum y_i)^2}} \quad (1)$$

The usage of the PCC for the evaluation of STS measures has been questioned before. Zesch (2010) refers to Anscombe (1973) to list three limitations:

- The PCC is sensitive to outliers.
- The PCC can only measure a *linear* relationship between vectors.
- The PCC needs the two vectors to be normally distributed.

Another, rather practical, issue is the unknown linear function. If the two vectors have different codomains, without the knowledge of the linear function one cannot determine the actual values that are correlated. For example if the gold standard vector $\vec{g} \in [0,4]$ and the values of the STS measure are $\vec{m} \in [0,1]$. The PCC value can still be high. If the values of $\vec{m}$ would be needed in an application, a linear function would have to be applied first to get to the values of a vector $\vec{m}'$ that has similar values to $\vec{g}$ and could be used in the application. In this case just using a linear regression between $\vec{m}$ and $\vec{g}$ would produce this linear function and $\vec{m}'$can easily be determined. But if the STS measure is used in an application without a known “gold standard” getting $\vec{m}'$ is not trivial.

To overcome the three before mentioned limitations Zesch (2010) recommends to use Spearman's rank correlation coefficient (SCC), which is also commonly used for the evaluation of STS measures. The SCC does not use the actual values to compute a correlation,

but the ranking of the values. The vectors $\vec{m}$ and $\vec{g}$ would first be transformed into vectors that contain the ranking of the values within the vectors. For ties, where the values are the same, the arithmetic mean between all individual ranks is used. The formula to compute the correlation is the same as the formula for the PCC (1) just with the transformed vectors. In case no ties occur the formula can be simplified to:

$$\rho = \frac{1 - 6\sum d_i^2}{n(n^2 - 1)}$$

Where n is the length of the vectors and d_i is the difference in rank between m_i and g_i.

The SCC is much more robust against outliers, can cope with non-linear relationships, and does not require any special distribution of variables. The issue with the unknown linear function does not apply because the actual values are not used. This leads to the main disadvantage. No conclusion about how well the *values* of two vectors correlate can be drawn. Hauke and Kossowski (2011) show that a negative correlation between values (PCC) can still imply a positive correlation between ranks (SCC).

2.3.2.2 Binary Classification

Another way to evaluate STS measures is by testing how well the measure can be used to classify the similarity of two texts. The idea behind this is not to evaluate continuous value mappings (e.g. 0.9 → 0.95 or 0.1 → 0.95) but a binary classification (e.g. $t \rightarrow t$ or $t \rightarrow f$). The class for the evaluation of STS measures are commonly *similar* (t) and *not similar* (f). A mapping between two binary vectors can result in four combinations: true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). A typical measure to evaluate classifications is the F1 measure, which is the harmonic mean between the recall and the precision of a classifier (van Rijsbergen, 1979). Another important measure is the accuracy.

$$precision = \frac{tp}{tp+fp} \qquad recall = \frac{tp}{tp+fn} \qquad accuracy = \frac{tp+tn}{tp+tn+fp+fn}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

In case the precision or the recall is more important, the F measure can be adapted accordingly. These measures are not unchallenged, e.g. by Powers (2011), but should work good enough for most scenarios.

Problematic is that typically STS measures do not produce binary but continuous values. In this case a mapping from continuous to binary values is needed. Usually it is not straight forward what value the delimiter for the mapping has.

2.3.2.3 Cumulative Gain

Järvelin and Kekäläinen (2000) describes the Cumulative Gain (CG) and the Discounted Cumulative Gain (DCG) which later was extended to the normalised Cumulative Gain (nCG) and the normalised Discounted Cumulative Gain (nDCG) by Kekäläinen (2005). CG measures are common in Information Retrieval (IR) evaluation, especially in the evaluation of search result lists.

Let $\vec{g}$ be a vector of similarity scores for a set of texts ordered from highest to lowest (the gold standard). And let $\vec{m}$ be the vector that denotes similarity scores for the same set of texts produced by a STS measure, also ordered from highest to lowest. Then $\vec{v}$ is defined as the vector with the values of $\vec{g}$ but the order of $\vec{m}$. The CG is defined as ($|\vec{g}| = |\vec{m}| = |\vec{v}| = n$):

$$CG_i = \begin{cases} \vec{v}_i \,, if\ i = 1 \\ CG_{i-1} + \vec{v}_i \,, otherwise \end{cases}$$

The CG sums up all similarity scores up to i. A higher CG means that texts with higher similarity scores where ranked better. For $i = n$, the CGs for all STS measures are the same because that's the sum of all similarity scores of $\vec{g}$, therefore a i smaller n is chosen. Within this work the notation for a CG measure, that is only applied to the first i elements of a vector, is CG@ i.

Under the assumption that highly similar text should be ranked higher, the DCG was introduced, that applies a ranked based discount factor to the CG.

$$DCG_i = \begin{cases} \vec{v}_i \,, if\ i = 1 \\ DCG_{i-1} + \dfrac{\vec{v}_i}{log_b i} \,, otherwise \end{cases}$$

A typical value for b is 2 and for the rest of this work 2 will be used.

To make the comparison of CG measures possible over different set of texts, a normalisation of the CG is useful. The normalisation is done by computing the CG of $\vec{v}$ and dividing it by the CG of the vector $\vec{g}$, which produces an optimal score:

$$nCG_i = \frac{CG_i(\vec{v})}{CG_i(\vec{g})} \qquad nDCG_i = \frac{DCG_i(\vec{v})}{DCG_i(\vec{g})}$$

2.3.3 Datasets for Intrinsic Evaluation

For the evaluation of a STS measure a dataset is needed, that consists of pairs of texts aligned with a human annotation of how similar each pair of text is. This human annotation is generally called *gold standard*. Typically the gold standard is an average of the annotations of

more than one person. How well the different annotators agree on the similarity of a text pair is the therefore the upper bound of the quality of a STS measure and is called the inter annotator agreement. Different datasets use different scales for the similarity score. Some use definitions for only the extremums of the scale (*"highly unrelated" and "highly related",* Lee et al., 2005), others map different definitions of similarity to different values on the scale (Agirre et al., 2012). The range of most of the similarity scores is either between 0.0 and 1.0 (Mihalcea et al., 2006) or is a five point Likert scale (Jurgens et al., 2014). This section describes a dataset that will be used within this work for intrinsic evaluation as well as common datasets for the evaluation of STS measures.

Agirre et al. (2012) introduces a dataset for SemEval 2012 task 6, that is a composition of five sub datasets. The first corpus is the Microsoft Paraphrase Corpus, as introduced by Dolan et al. (2004). It contributes 1500 text pairs. The next is the Microsoft Video Description Paraphrase Corpus by Chen and Dolan (2011), also contributing 1500 text pairs. The third and fourth dataset are taken from the 2007 ACL Workshops on Statistical Machine Translation (Callison-Burch et al., 2007), 1,193 from the first dataset and 399 from the second. The last dataset was a mapping of glosses from WordNet (Miller and Fellbaum, 1998) and OntoNotes (Hovy et al., 2006) from which 750 pairs were taken. Each text pair was annotated on a 0.0 to 5.0 scale for text similarity (gold standard).

The task 6 of SemEval2012 was to develop a STS measure to compute the STS for each text pair. For the first three sub datasets, about 50% of the pairs were given as training data and the rest was used for test data. The last two dataset were "surprise" datasets were no training data was provided. Each participant was allowed to submit up to three different runs and 35 teams submitted 88 different runs. The main measure to evaluate the STS systems was the PCC, once used overall, once used with normalization, and once with a weighted factor for the size of each sub dataset. As can be seen in Figure 1, the distribution of similarity scores of the test data ($n = 3108$) is biased towards higher similarity.

Other common datasets for the evaluation of STS measures that will not be used in this work were introduced by Li et al. (2006) and Lee et al. (2005). Li et al. (2006) developed a dataset of 65 sentence pairs based on prior work of Rubenstein and Goodenough (1965) where 65 word pairs were used. The initial 65 sentence pairs were reduced to 30 pairs to lessen a bias towards dissimilar pairs in the frequency distribution. Similarity was rated on a scale between 0.0 and 4.0. Lee et al. (2005) introduced a set of 50 sentences in which every sentence was compared to every other sentence on a 1 to 5 scale, resulting in 1225 distinct text pairs.

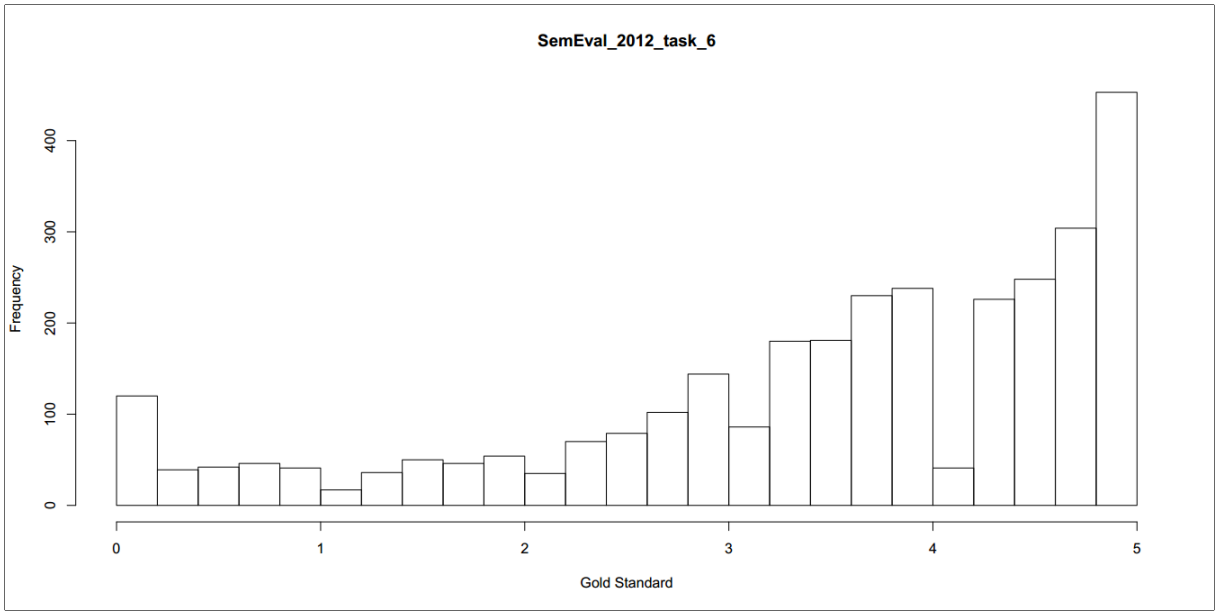

Figure 1: SemEval 2012 Task 6 Gold Standard Histogram of Test Data

2.4 Research Question

Within the field of Semantic Text Similarity (STS) the question, which evaluation method to use, remains unsolved. The common evaluation methods, like the Pearson product-moment correlation coefficient (PCC) and the Spearman's rank correlation coefficient (SCC), have been criticised, but alternatives, like the F1 measure or the nDCG, have their own weaknesses. A single evaluation method that reflects all performance aspects of a STS measure probably does not exist. From the perspective of a practitioner the challenge is to pick the best STS measure for a STS based application. An answer to the following two questions would assist him in this task:

1. Do different intrinsic evaluation methods (like correlation, classification, and cumulative gain) come to the same conclusion when used for the evaluation of STS measures? Especially, are the PCC or the SCC the best indicators for the performance of a STS measure?
2. Do different STS based applications have different requirements for the employed STS measure and can these requirements be aligned to specific evaluation methods? Can therefore an intrinsic evaluation predict the outcome of an extrinsic evaluation, which is the performance of a STS based application?

Answering these questions would enable practitioners who develop STS based applications to a more sophisticated decision on which STS measure to use. After deciding on their

requirements on a STS measure, they could pick one of the STS measures which performs best within the evaluation method that is aligned to the requirements. The requirements for the STS measures are closely related to the task at hand and therefore should not be difficult to determine. A prerequisite would be that developers of STS measures publish the performance of STS measures not only on the basis of one evaluation method (typically the PCC or SCC) but on many different.

3 Evaluation of STS Systems

There are various evaluation methods which all evaluate different properties of semantic text similarity (STS) measures. The Pearson's product-moment correlation coefficient (PCC) measures the linear relation of STS scores to a gold standard, while the normalised Discounted Cumulative Gain (nDCG) evaluates the ranking of STS scores with a high emphasis on the higher ranks. Still, the evaluation of STS measures is dominated by only a few measures, namely the PCC and Spearman's rank correlation coefficient (SCC). The first research question, stated in section 2.4, asks whether evaluation methods come to different results when comparing STS measures. That would imply that one evaluation method (e.g. the dominant PCC) is not sufficient to evaluate all properties of a STS measure. The second research question asks whether different STS based applications have different requirements on STS measures and therefore different properties of STS measures are important to different STS based applications. Followed by the question whether a mapping of evaluation methods to requirements would be possible, which would allow to predict the performance of different STS measures when used by a STS based application.

This chapter consists of two parts. In the first part (section 3.1) the rankings of STS measures with respect to different evaluation methods are compared, which will help to answer the first research question. In the second part (section 3.2) requirements for STS based applications are listed and the STS Evaluation Framework is proposed that maps evaluation methods to requirements, which will be tested in the following chapter. The intension of the framework is to help to answer the second research question.

3.1 Ranking of STS Measures with Different Evaluation Methods

The Pearson product-moment correlation coefficient (PCC) and the Spearman's rank correlation coefficient (SCC) are the dominate evaluation methods for STS measures. Within this section further possible evaluation methods are introduced and (as well as the PCC and SCC) applied to a dataset of submissions to SemEval 2012 task 6. This allows the comparison of the ranking of different STS measures according to different evaluation methods.

3.1.1 Setup of Comparison

The SemEval 2012 task 6, as described in section 2.3.3, was a challenge with the task to determine the semantic textual similarity for pairs of short sentences in a range from zero to five. 35 teams form within the academic community submitted 88 different runs (different STS measures or at least different setups for the measures) and the scores for each text pair of

each run are publically available.[2] In the official shared task (Agirre et al., 2012) only the PCC was used to determine the ranking of the different submission. Section 3.1.2 computes the ranking for other evaluation methods, like SCC, and compares the ranking based on PCC with rankings based on other evaluation methods.

3.1.2 Evaluation Methods

For the comparison of the rankings, a variety of evaluation methods will be applied to the submissions. The selection of evaluation methods include all evaluation methods that are part of the STS Evaluation Framework that will be proposed in section 3.2, as well as additional measures for a broader overview.[3] The following evaluation method will be used for the ranking of the different STS measures submitted to the SemEval 2012 task 6 challenge:

Pearson product-moment correlation coefficient (PCC)

The formula for the PCC from section 2.3.2.1 applied to the each submission and the gold standard. The PCC is part of the STS Evaluation Framework and the dominant measure to evaluate STS measures.

Spearman's rank correlation coefficient (SCC)

The formula for the SCC from section 2.3.2.1 applied to the ranks of each submission and the ranks of the gold standard. The SCC is part of the STS Evaluation Framework and a common measure to evaluate STS measures.

nDCG@all (nDCG_All)

The formula for nDCG from section 2.3.2.3 applied to each submission and the complete gold standard. The nDCG@all (*all* indicates that the measure is applied to the complete dataset) is a common measure especially for Information Retrieval and can be used as an alternative to the SCC.

Mean of nDCG@3,5,10 (nDCG_Avg)

The mean of the nDCGs as described in section 2.3.2.3 applied to the highest ranked 3, 5, and 10 values of each submission and the respective gold standard. Unlike the nDCG@all, only the ranking of the 3, 5, and 10 most similar text pairs are of importance. That means the gold standard values of only the top 3, 5, and 10 ranking texts according to the respective STS measure are summed up. This reflects a use case where only a subset of the results is used, for example in form of a result list of limited length.

[2] http://www.cs.york.ac.uk/semeval-2012/

[3] Compositional measures of the STS Evaluation Framework are excluded.

Mean of nCG@3,5,10 (nCG_Avg)

The mean of the nCGs as described in section 2.3.2.3 applied to the highest ranked 3, 5, and 10 values of each submission and the respective gold standard. Unlike the nDCG_Avg, the ranking of the most similar text pairs is not important. Instead only the correct selection of highly similar text pairs is crucial. This calculation is equivalent to nDCG_Avg, but without a discount factor. The measure is also part of the STS Evaluation Framework.

Accuracy of low similarity (Acc_low)

The formula for accuracy from section 2.3.2.2 applied to each submission and the gold standard. A similarity score of below 1.5 (on a 0.0 to 5.0 scale) was arbitrarily chosen as a threshold for low similarity. Any submission value of below 1.5 similarity score with a corresponding gold standard also being lower then 1.5 was considered *true positive*. Such a measure helps to evaluate the ability of a STS measure to detect dissimilar text pairs.

F1 of low similarity (F1_low)

The formula for precision, recall, and F1 from section 2.3.2.2 applied to each submission and the gold standard. Any submission value of below 1.5 similarity score with a corresponding gold standard also being lower then 1.5 was considered *true positive*. This measure is similar to Acc_low and part of the F1_hmean (see below) which in turn is part of the STS Evaluation method.

Accuracy of high similarity (Acc_high)

The formula for accuracy from section 2.3.2.2 applied to each submission and the gold standard. A similarity score of above 3.5 (on a 0.0 to 5.0 scale) was arbitrarily chosen as a threshold for high similarity. Any submission value of above 3.5 similarity score with a corresponding gold standard also being above 3.5 was considered *true positive*.

F1 of high similarity (F1_high)

The formula for precision, recall and F1 from section 2.3.2.2 applied to each submission and the gold standard. Any submission value of below 3.5 similarity score with a corresponding gold standard also being above 3.5 was considered *true positive*. Like the F1_low, this measure is part of the F1_hmean, which is part of the STS Evaluation method.

Macro average of accuracy for low and high similarity (Acc_macro)

The unweighted mean of Acc_low and Acc_high. Such a measure helps to evaluate the ability of a STS measure to detect dissimilar as well as similar text pairs.

Harmonic mean of accuracy for low and high similarity (ACC_hmean)

The harmonic mean of Acc_low and Acc_high. This measure is similar to Acc_macro, but uses a different method to calculate the mean.

Macro average of F1 for low and high similarity (F1_macro)

The unweighted mean of F1_low and F1_high. Unlike the Acc_macro this measure is based on F1 scores, the purpose remains the same.

Harmonic mean of F1 for low and high similarity (F1_hmean)

The harmonic mean of F1_low and F1_high. The measure is part of the STS Evaluation Framework. The capability to evaluate the ability of a STS measure to differentiate between similar and dissimilar text pairs makes this measure valuable for classification tasks.

Table 13 shows all submissions ranked with the above stated evaluation methods, while Table 1 only shows the Mean Absolut Difference (MAD) as a measure of differences between the rankings of the different evaluation methods. For the MAD, the sum of the absolute differences in rank of each STS measure was calculated and divided by the number of STS measures. That was done for every combination of two evaluation methods. The Mean Squared Difference (Table 14) as well as the SCC (Table 15) between ranks were also calculated and showed to similar results.

Mean Absolut Difference	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean
PCC	0.0	6.6	19.0	29.4	28.9	9.2	13.8	14.6	17.2	11.4	12.5	11.3	12.5
SCC	6.6	0.0	19.6	29.1	28.6	13.4	17.5	13.3	16.5	11.3	12.0	13.9	15.2
nDCG_All	19.0	19.6	0.0	20.6	19.8	18.6	21.9	21.5	21.7	20.7	21.1	20.3	20.8
nDCG_Avg	29.4	29.1	20.6	0.0	1.3	30.2	26.4	31.4	31.9	31.8	31.8	27.5	26.3
nCG_Avg	28.9	28.6	19.8	1.3	0.0	29.8	25.8	30.9	31.3	31.2	31.2	26.9	25.8
Acc_low	9.2	13.4	18.6	30.2	29.8	0.0	12.7	14.2	15.3	9.4	11.2	9.1	10.8
F1_low	13.8	17.5	21.9	26.4	25.8	12.7	0.0	20.4	21.8	17.8	18.7	8.4	5.9
Acc_high	14.6	13.3	21.5	31.4	30.9	14.2	20.4	0.0	4.2	4.9	3.3	13.6	16.1
F1_high	17.2	16.5	21.7	31.9	31.3	15.3	21.8	4.2	0.0	7.3	6.0	15.0	17.6
Acc_macro	11.4	11.3	20.7	31.8	31.2	9.4	17.8	4.9	7.3	0.0	1.9	10.6	13.3
Acc_hmean	12.5	12.0	21.1	31.8	31.2	11.2	18.7	3.3	6.0	1.9	0.0	11.6	14.3
F1_macro	11.3	13.9	20.3	27.5	26.9	9.1	8.4	13.6	15.0	10.6	11.6	0.0	3.3
F1_hmean	12.5	15.2	20.8	26.3	25.8	10.8	5.9	16.1	17.6	13.3	14.3	3.3	0.0

Table 1: Mean Absolut Difference between ranks of submissions to SemEval 2012 task 6

The MAD between PCC and nDCG_Avg is 29.4. This means that the in average the ranks of a STS measure submitted to the Sem Eval 2012 task 6 challenge evaluated by PCC is 29.4 ranks apart from the same STS measure ranked by nDCG_Avg. Except for the SCC (6.6) and the Acc_low (9.2), all differences between an evaluation method and the PCC is higher than 10. For the SCC, the other commonly applied evaluation method, all evaluation methods, with the exception of the PCC, have MAD scores of higher than 10 as well.

3.1.3 Conclusion

Table 1 shows that different evaluation methods result in different rankings for STS measures. Therefore one evaluation method is not sufficient to evaluate all properties of a STS measure. But it can't be determined which evaluation method is the best indicator for the performance of a STS measure. It can be assumed that for different kinds of STS based applications, different requirements for the applied STS measure exist. And therefore different evaluation methods could be the best indicators for the performance of STS measures with respect to the application that uses the STS measure.

3.2 Proposal for a Framework to Map Requirements to Evaluation Methods for STS Based Applications

It has been shown in section 3.1 that different evaluation methods lead to different rankings of STS measures. Within this section the STS Evaluation Framework, a framework to organise requirements of STS based applications and align them to evaluation methods is proposed.

The framework was developed under the theory that different STS based applications have different requirements on STS measures and, therefore, different properties of STS measures are important to different STS based applications. Furthermore, that it is possible to align requirements of STS based applications to evaluation methods for STS measures.

Three dimensions on requirements of STS based applications are proposed: *Cardinality*, *Set of Interest*, and *Information*. For various combinations of these categories, evaluation methods for STS measures are then proposed. This alignment of requirements and evaluation methods constitutes the STS Evaluation Framework. The framework, and thereby the theory, will then be tested in chapter 4.

3.2.1 Dimensions of Requirements for STS Measures within STS Based Applications

Three dimensions will be used to structure the different requirements for STS measures: *Cardinality*, *Set of Interest*, and *Information*. With the help of these three proposed dimensions, and their respective subcategories, requirements and evaluation methods will be aligned.

Cardinality describes how many texts are compared to how many other texts. It consist of two sub categories *1:1* and *1:n*. *1:1* in this context means that exactly one text is compared with exactly one other text and only the result of this single comparison is of interest. This may happen multiple times with different texts but the result of one comparison will not be used in relation to the result of any other comparison. Whereas *1:n* means that one text will be compared with a whole set of other texts and the results of these comparison will be used in some way together.

A third option, *n:n*, where two sets of texts are compared, would theoretically be possible as well, but no example of this could be found in a study that will be shown in section 3.2.2 and will, therefore, not be considered further.

Set of Interest describes which of the elements of the result set will be used. It has three sub categories: *All*, *K-best*, and *Threshold*. *All* in this context means that all results of all comparisons will be used in some form. *K-best* describes the case where only the "k" best results will be used in some way. And *Threshold*, as the name already indicates, is used when only results of values over a certain threshold will be used.

Other sub categories would be possible as well. For example the *K-worst* results or some kind of an *interval* that consists of all the results, that are not quite similar or dissimilar. As with the possible subcategory of *Cardinality* of *n:n*, none of those sub categories appeared in the study in section 3.2.2 and therefore are not followed.

Information describes the type of information from the result set that is of interest. It has three sub categories: *Value, Rank,* and *Classification.* The case where the actual value of the result of a comparison is of interest falls in the category *Value. Rank* on the other hand is used if only the rank of each comparison is used in some way. *Classification* means that a simple classification, texts are similar or not, is used.

3.2.2 Study of Requirements for STS Based Applications

Within this section a short study on the requirements of STS based applications is presented. This list is not intended to be exhaustive or representative, but to demonstrate that different requirements on STS measures exist. To conduct this study some STS based applications were analysed and classified according to their requirements on STS measures (see section 3.2.1).

Table 2 depicts the classification of different STS based applications according to the before mentioned dimensions. The 16 analysed papers describe 25 different STS based applications. For the papers describing more than one application a comment was added to help

differentiate between the applications. For some applications it was not clear what requirement existed or contradicting statements for requirements existed. In those cases multiple or none requirements were ticked.

Domain	Publication	Comment	Cardinality		Set of Interest			Information		
			1:1	1:n	All	K-best	Thres-hold	Value	Rank	Classi-fication
Automatic Essay Grading	Attali and Burstein (2006)		X	0	0	0	0	X	0	0
	Foltz et al. (1999)		0	X	X	0	0	X	0	0
	Valenti et al. (2003)	IEA	X	0	0	0	0	X	0	0
	Valenti et al. (2003)	ETSI	X	0	0	0	0	X	0	0
	Valenti et al. (2003)	E-Rater	X	0	0	0	0	X	0	0
	Valenti et al. (2003)	C-Rater	X	0	0	0	0	X	0	0
	Valenti et al. (2003)	Automark	0	X	0	X	0	0	0	0
	Valenti et al. (2003)	PS-ME	0	X	0	0	0	X	0	0
	Mohler and Mihalcea (2009)		X	0	0	0	0	X	0	0
Clustering	Steinbach et al. (2000)	K-Means	0	X	0	X	0	0	0	X
	Steinbach et al. (2000)	Hierachicaly	X	X	X	0	0	X	0	0
	Song et al. (2009)		X	X	X	0	0	X	0	0
	Strehl et al. (2000)		0	X	0	X	0	0	0	X
IR	Hliaoutakis et al. (2006)	Expansion	0	X	0	0	X	0	0	X
	Hliaoutakis et al. (2006)	IR	0	X	0	X	0	0	X	0
Link Detection	Lu et al. (2009)		0	X	0	0	X	0	0	X
	Jin et al. (2007)		0	X	0	X	0	X	0	0
	Milne and Witten (2008)		0	X	0	X	0	0	0	X
	He (2009)	Run2	0	X	0	X	0	0	0	0
	He (2009)	Run3	0	X	0	0	X	0	0	X
	Knoth et al. (2011)		0	X	0	X	0	0	0	0
Plagiarism	Barrón-Cedeño et al. (2013)		0	X	0	0	X	0	0	0
	Stein et al. (2007)		X	0	0	0	X	0	0	0
Summarization	Hovy and Lin (1998)	Summac, 98	0	X	0	X	0	0	0	0
	Hovy and Lin (1998)	Marcu, 97	X	0	0	0	0	0	0	X
Count			**10**	**17**	**3**	**9**	**5**	**11**	**1**	**7**

Table 2: STS Based Applications and Requirements

Table 2 shows that indeed different requirements on STS measures within STS based applications exist. Because the collection of information as well as the evaluation was not rigorous at this point it was refrained from any further conclusion based on this study.

3.2.3 Common Combinations of Requirements

Within this section, a mapping of requirements to evaluation methods is proposed. As it was stated in section 3.2, the theory is that different properties of STS measures are important to different STS based applications and that it is possible to align requirements of STS based applications to evaluation methods for STS measures. For example should the SCC of STS measure be more important to STS based application were a ranking is involved then the PCC. Instead of mapping one evaluation method to one requirement, one evaluation method will be mapped to one combination of requirements. This will lead to a more distinct alignment of requirements. Whether this alignment is sufficient will be tested exemplarily in chapter 4.

With the three dimensions and the corresponding sub categories 18 different combinations are possible.[4] However, some of these combinations can be disregarded because they are not plausible.

For any combination that involves a *Cardinality* of *1:1* only a *Set of Interest* of *All* is useful, because the result set contains only one result. In addition, the *Information* can't be *Rank,* because the result set only contains one element. This reduces the possible combinations with *Cardinality* of *1:1* to *(1:1, All, Value)* and *(1:1, All, Classification)*. The combination *(1:n, K-best, Classification)* corresponds with the question: Of the k most similar texts, how many should be classified as similar? This seems to be an unlikely question and the circumstances of such a question coming up within an actual application are hard to imagine and will therefore be excluded. Similar to that, the combination *(1:n, Threshold, Classification)* corresponds to the question: Of all results with a similarity score higher than a certain value (for the *Threshold*), how many have a score that is higher than another value (for the *Classification*). Applying a classification twice is again an unlikely requirement for an actual application and therefore this won't be considered as well.

The remaining nine combinations will be described in this chapter. For each combination one evaluation metric will be proposed that fits the requirements of this particular combination best.

(1:1, All, Value) is the most common combination that was found in section 3.2.2. It describes the case where an application needs the value of the similarity score of two texts. One example is an Automatic Essay Grading system where a student's essay is compared to an ideal essay, for example written by a teacher. In this case, the higher the similarity score, the

[4] 2(*Cardinality*) * 3(*Set of Interest*) * 3(*Information*) = 18

better the grade for the essay gets and a low similarity score, in turn, would lead to a low essay grade. Of cause to evaluate a STS system one would need to consider not a single instance of such a comparison but many.

This combination is also common for the intrinsic evaluation of STS systems. For example, the SemEval 2012 task 6 is such a combination (see section 2.3.3). Here, more than 3000 pairs had to be compared with the goal to have a score as similar as possible to a gold standard. So here the combination *(1:1, All, Value)* was used more than 3000 times. The SemEval 2012 task 6 is, in fact, an intrinsic and not an extrinsic evaluation. But the used evaluation method, the PCC, seems to be a good metric to evaluate this combination as it provides the linear dependence between the values of two variables.

(1:1, All, Classification) is used for classification decisions, that are based purely on the score of two texts that are compared to each other. An example of a binary decision would be to accept or reject an answer to a question based on how similar the answer is to an ideal answer. Compared to the *(1:1, All, Value)* combination, the main difference is, that the actual value of the similarity score is not further used. The answer, to use the example again, is either accepted or rejected, not partially accepted or rejected based on the value of the similarity score. The task at hand is therefore to identify similar and dissimilar texts, not *how* similar or dissimilar these texts are.

As with the former combination, not a single instance of a classification would be used for evaluation, but many. With classification, the issue occurs that not all applications have the same requirements on how similar texts have to be to be considered similar. A plagiarism detection system would probably want a higher threshold than a text clustering system where texts that are quite different could still be in the same cluster. One possible solution would be to provide the evaluation results for many different levels of threshold so that the most fitting for the purpose can be selected. The most common metrics to evaluate classification problems are recall, precision, and the F1 measure which is a combination of both. To minimize the amount of metrics only the F1 measure will be used, although for some application recall or precision might be more relevant. To overcome the challenge of different thresholds for similarity, one F1 measure can be calculated for very similar scores and one can be calculated for dissimilar scores. The harmonic mean of both F1 measures can be a good indicator for classifications.

(1:n, All, Value) describes the case when one text is compared to many texts and the similarity score of all these comparisons is used in some way. Similar to the *(1:1, All, Value)* case, an

AEG system can be a good example for this combination. The difference is that the essay of the student is not compared to only one 'perfect' essay, but to many. It is also possible to compare this essay to a particularly bad essay. The score for the students essay would be a combination of the similarity scores of all comparisons. This means that every single score of the comparisons is equally valuable for the application. Therefore, the PCC is a good way to evaluate such a system. Although a single comparison of one text to many is enough to calculate a correlation coefficient, it is probably better to evaluate the comparisons of many instances of a 1:n comparison to increase the sample size.

(1:n, All, Rank) is used when one text is compared to many and the order of how similar these texts are to the one text is important. The rank is often important when a list of results is shown to humans. For example in plagiarism detection, the most likely candidate for plagiarism should be presented first. In case STS is the only indicator for plagiarism, the most likely candidate would be the one with the highest similarity score.

Typical measures to evaluate the similarity of two rankings are Spearman's rank correlation coefficient (SCC) and the normalized Discounted Cumulative Gain (nDCG). The correlation coefficient values only the ranking but all ranks the same, while the nDCG incorporates the actual similarity score and values the rank of very similar texts higher than the ranking of not so similar texts. Depending on the task either the SCC or the nDCG should perform well.

(1:n, All, Classification) tries to identify texts that are similar to one text. An example for an application that could benefit from this combination is a text tagging system. A tagging system could assign a tag to a text if a similar text already has this tag assigned to. Again, the problem rises that different levels of thresholds could be useful for different applications. The same rational as with the *(1:1, All, Classification)* can be applied and so will the same evaluation metric, the harmonic mean of low and high F1 measures.

(1:n, K-best, Value) is used by applications that only consider the k-most similar pairs and need to know the actual similarity score of those pairs. The size or the quality of the set that is compared to the one text is most likely not important to determine the value of k. If size and quality would be relevant, the *(1:n, Threshold, Value)* combination would be much more useful. The value of k is more likely determined by some other requirements of the application. A common value for k is 1, if only the most similar text is important for the application. If some kind of human interaction is needed, a small k is also appropriate so the amount of human interaction can be reduced. The subsequent usage of a computational intensive algorithm is another reason for a small value of k.

This divides the task in two sub tasks. First, an evaluation metric needs to consider if the right k most similar texts have been identified and, secondly, how well the similarity scores have been calculated compared to the actual similarity scores.

For the first task, typical classification evaluation methods like accuracy, recall, precession, or an F1 measure could be used. This would have the disadvantage that a text, wrongfully identified as belonging to the set of the k most similar texts, penalises the result of the evaluation quite considerably even if the wrongfully identified text is actually just slightly less similar than the right one. For example, a combination with a k of one and the STS system identifies a text with a similarity score of 4.995 as the most similar, while the actual most similar text has a score of 4.996. The typical classification evaluation methods would score such a result quite low. Although for an application it might not be important to actually find the most similar text, finding a text that is nearly as similar is good enough. Therefore a measure like the normalized Cumulative Gain can be more useful. It puts the similarity score of the actual most similar texts in relation to the similarity score of the texts that the STS system identifies as the most similar. The example above would lead to a quite high nCG of $\frac{4.995}{4.996} = 0.9998$ where 1 would be the optimal result. For the second problem the requirements are not different from the *(1:n, All, Value)* combination, so the PCC can be used.

As a last step the two measures, the nCG and the PCC, need to be combined. Similar to the F1 measure, that uses the harmonic mean between the recall and the precision, the harmonic mean between the nCG and the PCC can be used. This has the advantage that only if both measures perform well the combined measure performs well too. This combined measure needs to be calculated for different values of k. The values for the different k's depend on the composition and size of the dataset. As the PCC can become negative, the absolute value will be used.

(1:n, K-best, Rank) has similar properties as the *(1:n, K-best, Value)* combination above. The difference is that the rank of the result is important and not the actual similarity score. This is the combination that would be used to identify for example the one most similar pair of texts. Here again two different task occur. Can the k-most similar text pairs be identified and are these pairs in the right order? A combined measure like the one proposed for the *(1:n, K-best, Value)* combination is suitable. The only difference is that this time the absolute value of SCC in combination with the nCG will be used instead of the PCC because the rank, not the value, is important. Another suitable evaluation method for this task would be the nDCG because it values the rank. As described in section 2.3.2.3, the penalty for high ranking text pairs in the

wrong order is higher than for low ranking text pairs in the wrong order. If this constraint is acceptable or even wanted, the mean of the nDCG@k with different values for k can be used.

(1:n, Threshold, Value) is a combination useful for applications where the similarity scores of all text pairs with a score higher than t is needed. In a plagiarism detection scenario such a combination could be applied. By using a threshold parts of the text where no plagiarism occurred can be approved without the need of human interaction. There are two challenges with this combination, similar to the combination where the *Set of Interest* is K-best. First, all text pairs with a similarity score of t or higher need to be identified and then the similarity scores need to be right as well. The first challenge can be addressed by using the harmonic mean of low and high F1 measures as it is used for combination with *Information* being *Classification*. The second challenge can be handled by also applying the absolute value of the PCC.

(1:n, Threshold, Rank) is similar to the previous combination with the difference that the order of text pairs in the set of pairs with a similarity score higher than t is of interest. Every time search results that are based on a similarity score are listed, this combination can be useful. For search results, the order of similarity should be represented by the order the results are listed, therefore the rank is important. The threshold helps to limit the list of results to the actually useful results. The only difference to the evaluation metric that is recommended for *(1:n, K-best, Value)* combination, is that absolute value of the SCC in combination with the harmonic mean of low and high F1 measures will be used instead of the PCC.

Table 3 summarizes the recommendation for evaluation methods with respect to different combinations of requirements on STS measures within STS based applications (STS Evaluation Framework). The PCC as well as the SCC are the two evaluation methods recommended most often. This is in line with the task for this paper to build a framework that *extends* on the common evaluation methods, not to *replace* these measures.

Combination	Measure 1	Measure 2
(1:1, All, Value)	PCC	
(1:1, All, Classification)	F1_hmean	
(1:n, All, Value)	PCC	
(1:n, All, Rank)	SCC	nDCG
(1:n, All, Classification)	F1_hmean	
(1:n, K-best, Value)	Harmonic mean of nCG@K and |PCC|	
(1:n, K-best, Rank)	Harmonic mean of nCG@K and |SCC|	nDCG@K
(1:n, Threshold, Value)	Harmonic mean of F1_hmean and |PCC|	
(1:n, Threshold, Rank)	Harmonic mean of F1_hmean and |SCC|	

Table 3: STS Evaluation Framework

3.3 Utilisation of STS Evaluation Framework

The goal of the STS Evaluation Framework is to predict the performance of a STS based application by evaluating the STS measure that is used within the application. The used evaluation method dependents on the requirements the STS based application has on the STS measure. Of cause the prediction of the performance of the STS based application can only go so far as the performance depends on the STS measure. Therefore, it would rather be possible to compare the performance of different STS measures within a STS based application than to make a statement about the performance of the STS based application itself.

For example, a text tagging system assigns a tag to every text that is similar to an archetype of text. The requirements for a STS measure according to this framework would be *(1:n, All, Classification)*. *1:n* because one text needs to be compared to a set of texts. *All*, because every text within the set can receive a tag, therefore all texts are of interest. And *Classification*, because it is a binary decision whether the text is similar enough to the archetype text that a tag can be assigned to it. From Table 3 we can take that the harmonic mean of the F1 scores of high and low similarity (F1_hmean) is suitable for this combination of requirements. According to the theory of the framework, a STS measure that has a higher F1_hmean score will perform better within the tagging system. A practitioner, who is developing a tagging system, could now choose a STS measure that is known to perform well according to the F1_measure and would not need to test different STS measures. Or at least could test fewer STS measures and pick measures with a higher likelihood to perform well. Figure 2 shows the generic process of how to apply the STS Evaluation Framework.

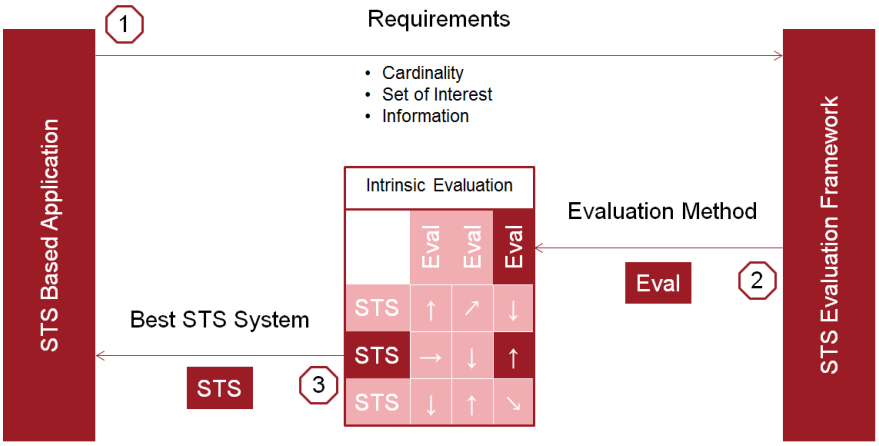

Figure 2: Utilisation of STS Evaluation Framework. 1. Requirements of STS based application are assessed by the STS Evaluation Framework. 2. STS Evaluation Framework suggests evaluation method. 3. The best STS measure according to the suggested evaluation measure is used by the STS based application.

3.4 Conclusion

Within this chapter it was shown that the ranking of STS measures is based on the used evaluation method (section 3.1.3) and that, according to the STS based application that applies the measure, different requirements on STS measures exists (section 3.2.2). A framework was proposed that maps requirements to evaluation methods (Table 3). The framework so far is just a proposal and needs to be tested. Therefore, in the next chapter three short experiments in form of extrinsic evaluations will be discussed that test the framework.

4 Assessment of the STS Evaluation Framework

Section 3.1 showed that the performance ranking of STS measures differs depending on the selected evaluation method. Section 3.2 introduced the STS Evaluation Framework, a framework, that maps requirements of STS based application to evaluation methods (see Table 3). Depending on the combination of requirements of the STS based applications, the intrinsic rankings of STS measures according to certain evaluation methods should resemble the rankings of the STS measures according to extrinsic evaluations. This would allow for a prediction of the performance of STS measure when used in STS based applications.

The following chapter evaluates this framework by conducting three experiments. In each experiment the intrinsic rankings of different STS measures for one dataset are compared to the rankings of STS measures when used in STS based applications (extrinsic evaluation) with another dataset.

4.1 Setup

For the intrinsic evaluation the SemEval 2012 task 6 dataset is utilized (Agirre et al., 2012, section 2.3.3). The selection and the implementations of the used STS measures are based on the publically available open source framework DKPro Similarity as presented by Bär (2013). Detailed information about the implementation as well as the stopword list can be found online.[5] The following STS measures are part of a compositional measure (see section 2.2.3) that ranked first place in the SemEval 2012 task 6 and was used as a baseline in the SemEval 2013 shared task: Semantic Textual Similarity (Agirre et al., 2013). These measures are used for the intrinsic evaluation as well as for the three extrinsic evaluations:

CharacterNGramMeasure {2, 3, 4}

The Character n-gram with values for n of 2, 3, and 4 (section 2.2.2). For the comparison the Jaccard correlation is used.

GreedyStringTiling {3}

Substrings of minimal length 3, composed by the Greedy String Tiling algorithm (section 2.2.1.1).

LongestCommonSubstringComparator

The length of the longest continuous common substring (section 2.2.1.2).

[5] https://code.google.com/p/dkpro-similarity-asl/

LongestCommonSubsequenceComparator

The length of the longest not continuous common substring (section 2.2.1.2).

LongestCommonSubsequenceNormComparator

The length of the longest not continuous common substring (section 2.2.1.2), normalised by the length of the first text (Clough and Stevenson, 2011).

WordNGramContainmentMeasure {1, 2} stopword filtered

The Word n-gram with values for n of 1 and 2 (section 2.2.2). For the comparison the Containment measure is used and a stopword filter is applied before calculation.

WordNGramJaccardMeasure {1, 3, 4}

The Word n-gram with values for n of 1, 3, and 4 (section 2.2.2). For the comparison the Jaccard correlation is used.

WordNGramJaccardMeasure {2, 4} stopword filtered

The Word n-gram with values for n of 2 and 4 (section 2.2.2). For the comparison the Jaccard correlation is used. A stopword filter is applied before calculation.

The SemEval 2012 task 6 dataset has a gold standard of range 0.0 to 5.0, while the STS measures have a range of 0.0 to 1.0. To allow value based evaluation methods to work with the results of the STS measures, the results have been scaled accordingly. As a factor, the coefficient of linear regression through the origin was used.

The STS measures are evaluated according to the methods listed in section 3.1.2 as well as according to the suggested methods from the STS Evaluation Framework (section 3.2). Table 4 depicts the ranking of each STS measure according to the different evaluation methods and Table 5 depicts the scores for each STS measure.

	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean	PCC_nCG_hmean	SCC_nCG_hmean	PCC_F1_hmean	SCC_F1_hmean
GreedyStringTiling 3	4	4	10	14	14	6	10	3	2	3	3	4	8	4	4	6	5
LongestCommon SubsequenceComparator	6	5	11	9	9	4	13	5	4	5	5	10	13	7	5	13	13
LongestCommon SubsequenceNormComparator	10	8	9	12	12	5	14	8	7	7	7	11	14	11	10	14	14
LongestCommonSubstring Comparator	14	12	12	13	13	8	9	10	10	9	9	8	6	14	14	9	8
WordNGramContainment Measure 1 stopword.filtered	5	7	13	10	10	7	12	6	5	6	6	9	12	8	9	12	10
WordNGramContainment Measure 2 stopword filtered	8	9	5	4	4	10	4	9	9	10	10	5	4	5	6	4	4
WordNGramJaccardMeasure 1	7	6	14	11	11	9	11	7	8	8	8	7	9	9	7	7	7
WordNGramJaccardMeasure 3	11	11	3	1	1	12	6	12	12	12	12	12	7	10	11	8	9
WordNGramJaccardMeasure 4	12	13	1	1	1	13	7	13	13	13	13	13	10	12	12	10	11
WordNGramJaccardMeasure 2 stopword filtered	9	10	6	4	4	11	5	11	11	11	11	6	5	6	8	5	6
WordNGramJaccardMeasure 4 stopword filtered	12	13	1	1	1	13	7	13	13	13	13	13	10	12	12	10	11
CharacterNGramMeasure 2	1	1	7	8	8	1	2	1	1	1	1	2	2	3	3	2	1
CharacterNGramMeasure 3	2	2	8	7	7	2	1	2	3	2	2	1	1	1	1	1	2
CharacterNGramMeasure 4	3	3	4	6	6	3	3	4	6	4	4	3	3	2	2	3	3

Table 4: Ranking of the STS measures for the SemEval 2012 dataset using various evaluation methods

The goals of the experiments are to answer the following three questions:

1. How well does the PCC or SCC predict the performance of STS measures within applications (see research question 1 in section 2.4)?
2. How well does the evaluation method suggested by the STS Evaluation Framework predict the performance of STS measures within applications (see research question 2 in section 2.4)? And is the suggested evaluation method by the STS Evaluation Framework the best predicator?
3. And finally, does the STS Evaluation Framework help to identify the best performing STS measures? The last question is most relevant for practitioners, but not as decisive for the assessment of the STS Evaluation Framework as a whole.

	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean	PCC_nCG_hmean	SCC_nCG_hmean	PCC_F1_hmean	SCC_F1_hmean
GreedyStringTiling 3	.52	.52	.95	.55	.58	.88	.16	.66	.70	.77	.75	.43	.26	.55	.55	.34	.34
LongestCommon SubsequenceComparator	.35	.43	.95	.65	.65	.88	.01	.63	.67	.75	.73	.34	.01	.46	.51	.02	.02
LongestCommon SubsequenceNormComparator	.32	.36	.95	.63	.62	.88	.01	.59	.63	.73	.71	.32	.01	.42	.46	.02	.02
LongestCommonSubstring Comparator	.24	.25	.95	.59	.58	.84	.26	.53	.49	.68	.65	.37	.34	.34	.34	.28	.28
WordNGramContainment Measure 1 stopword.filtered	.36	.38	.95	.64	.63	.86	.10	.61	.65	.74	.72	.37	.18	.46	.48	.23	.24
WordNGramContainment Measure 2 stopword filtered	.35	.32	.96	.99	.98	.72	.32	.55	.52	.63	.62	.42	.40	.51	.49	.37	.36
WordNGramJaccardMeasure 1	.35	.39	.95	.64	.63	.83	.15	.59	.61	.71	.69	.38	.24	.45	.48	.29	.30
WordNGramJaccardMeasure 3	.27	.26	.97	1	1	.49	.29	.47	.32	.48	.48	.30	.30	.43	.41	.29	.28
WordNGramJaccardMeasure 4	.25	.23	.98	1	1	.37	.26	.44	.21	.40	.40	.24	.24	.40	.38	.24	.24
WordNGramJaccardMeasure 2 stopword filtered	.34	.32	.96	.99	.98	.66	.31	.53	.46	.59	.59	.38	.37	.50	.48	.35	.34
WordNGramJaccardMeasure 4 stopword filtered	.25	.23	.98	1	1	.37	.26	.44	.21	.40	.40	.24	.24	.40	.38	.24	.24
CharacterNGramMeasure 2	.66	.60	.96	.71	.71	.91	.62	.69	.73	.80	.79	.67	.67	.69	.65	.67	.63
CharacterNGramMeasure 3	.66	.58	.96	.80	.80	.91	.66	.67	.70	.79	.77	.68	.68	.72	.67	.67	.62
CharacterNGramMeasure 4	.61	.54	.96	.85	.85	.88	.62	.63	.64	.75	.73	.63	.63	.71	.66	.62	.58

Table 5: Scores of the STS measures for the SemEval 2012 dataset using various evaluation methods

To answer these questions, the experiments follow three steps: First, for each of the three tasks of the extrinsic evaluation, the STS measures will be applied. A ranking will be calculated on how well the STS measures have performed within in the STS based application (the extrinsic ranking). Second, the extrinsic rankings will be compared to the rankings of Table 4 (the intrinsic ranking) and the mean absolute difference (MAD), the mean square difference (MSD), as well as the SCC between the rankings will be calculated. And third, depending on the requirements of the STS based application, the STS Evaluation Framework proposes one, sometimes two, evaluation methods that should predict the outcome of the comparison in step two. Section 4.4 discusses the outcome of the three experiments.

4.2 Extrinsic Evaluation I: Text Reuse

Clough and Stevenson (2011) introduced the Wikipedia Rewrite Corpus which is used for the first experiment. The dataset consists of 95 documents, each containing an answer about one of five questions about computer science. The answers employ different levels of reuse of a Wikipedia article. The degree of reuse was split in one of four categories: *near copy,*

light revision, heavy revision, and *non-plagiarised*. The performance is evaluated by calculating the accuracy (the system that classifies the most texts correctly is the best performing system).

The task of the extrinsic evaluation is to classify each document correctly as a member of one of the four classes. The combination of requirements (see section 3.2.3) therefore is *(1:1, All, Classification)*. According to the STS Evaluation Framework the F1_hmean[6] of the intrinsic evaluation should resample the performance of this task the best.

Table 7 depicts the differences in ranking of the intrinsic evaluation of STS measures as shown in Table 4, with the ranking of the extrinsic task (Table 6). To assign the numeric values produced by the STS measures to the four classes of the Wikipedia Rewrite Corpus, the Weka toolkit was used (Hall et al., 2009). As classifiers the 1R classifier (Holte, 1993) with optimised bucket sizes and a logistic regression classifier (Le Cessie and Van Houwelingen, 1992) has been used, both with 10-fold cross-validation. The classifier with better results was chosen for each STS measure. The accuracy and rank for each STS measure are depicted in Table 6.

STS Measures	Accuracy	Rank
GreedyStringTiling 3	0.43	14
LongestCommon SubsequenceComparator	0.56	11
LongestCommon SubsequenceNormComparator	0.44	13
LongestCommonSubstring Comparator	0.68	2
WordNGramContainment Measure 1 stopword.filtered	0.59	10
WordNGramContainment Measure 2 stopword filtered	0.63	8
WordNGramJaccardMeasure 1	0.60	9
WordNGramJaccardMeasure 3	0.66	3
WordNGramJaccardMeasure 4	0.65	5
WordNGramJaccardMeasure 2 stopword filtered	0.64	7
WordNGramJaccardMeasure 4 stopword filtered	0.65	6
CharacterNGramMeasure 2	0.55	12
CharacterNGramMeasure 3	0.65	4
CharacterNGramMeasure 4	0.70	1

Table 6: The performance and ranking of the STS measures for Extrinsic Evaluation I: Text Reuse

Table 7 shows that F1_low, nDCG_Avg, and nCG_Avg are the evaluation methods for intrinsic evaluation, which resample the extrinsic evaluation the best. The proposed framework predicted that the F1_hmean measure would be a good indicator and indeed it is placed between four or five out of 17, similar to nDCG_All. Although the evaluation of the extrinsic task was done with an accuracy measure, all measures that employee accuracy for the

[6] The harmonic mean of F1 values for very similar and very dissimilar text pairs (see section 3.1.2).

intrinsic evaluation are placed in the lowest third of the ranking. PCC as well as SCC are also placed in the lowest third. A discussion of the result follows in section 4.4.1.

	MAD		MSD		Spearman	
	Score	**Rank**	**Score**	**Rank**	**Score**	**Rank**
PCC_Rank	5.36	11	42.07	12	-0.326	12
SCC_Rank	5.64	13	42.50	13	-0.343	13
nDCG_All_Rank	3.64	5	19.07	5	0.431	4
nDCG_Avg_Rank	3.29	2	17.86	2	0.504	1
nCG_Avg_Rank	3.29	2	17.86	2	0.504	1
Acc_low_Rank	5.50	12	40.36	11	-0.277	11
F1_low_Rank	3.21	1	16.07	1	0.497	3
Acc_high_Rank	5.79	14	43.64	16	-0.378	16
F1_high_Rank	6.07	17	48.36	17	-0.524	17
Acc_macro_Rank	5.79	14	43.36	14	-0.370	14
Acc_hmean_Rank	5.79	14	43.36	14	-0.370	14
F1_macro_Rank	4.64	8	33.07	8	-0.053	8
F1_hmean_Rank	3.50	4	17.93	4	0.427	5
PCC_nCG_hmean_Rank	4.79	9	35.93	9	-0.136	9
SCC_nCG_hmean_Rank	5.07	10	38.50	10	-0.216	10
PCC_F1_hmean_Rank	4.07	6	23.21	6	0.264	6
SCC_F1_hmean_Rank	4.07	6	26.36	7	0.163	7

Table 7: Comparison of the ranking of STS measures of the Text Reuse experiment (Table 6) and Intrinsic Ranking (Table 4) using mean absolute difference (MAD), mean square difference (MSD) and Spearman coefficient

4.3 Related Articles

The second and the third experiment uses a newly created corpus compiled from DIE ZEIT and ZEIT Online. The following section describes the construction of the dataset. The second experiment (section 4.3.2) tries to distinguish whether two articles are related or not and can be seen as a pretest to the third experiment. The third experiment (section 4.3.3) has a more realistic approach to the task; it tries to find the one related article in a set of 20 articles.

4.3.1 Construction of the ZEIT Dataset

For the second as well as for the third task a corpus compiled from DIE ZEIT and ZEIT ONLINE is used. “DIE ZEIT” (eng. “the time”) is a German weekly newspaper which also has an online representation (“ZEIT ONLINE”).[7] For most of the articles, the authors added two related articles that provide further information on the same topic. Selecting appropriate articles can be time consuming and an automation is desired. This experiment evaluates how STS measures can help to support the process. The basis is a dataset consisting of 4654 articles from the category Politics | Germany between January 1st 2012 and March 31st 2014.

[7] http://www.zeit.de/index

For these articles, the authors added a total of 10.111 related articles, typically two per article, at most three. The related articles are also part of the set of 4654 articles. We focus only on articles from the category Politics | Germany to ensure that all articles come from a similar domain. For our purpose the gold standard is binary (*related* or *not related*). The 4654 articles and the 10.111 links between them can also be interpreted as a graph, the articles being the nodes and the links being the edges.

The construction of the dataset has some limitations. The content of two articles could be related in reality but no direct link between them exists. To lessen this weakness, not only directly linked articles are considered related, but also articles which are linked indirectly with other articles in between.

Unlike the SemEval 2012 task 6 dataset and the Wikipedia Rewrite Corpus, the ZEIT Corpus is in German, which can have an effect on the performance and comparability of the individual STS measures.

4.3.2 Extrinsic Evaluation II a: Related Article Pairs

The goal of the second experiment is to distinguishing related from unrelated articles. This is a pretest to the final goal, a method to automatically propose related articles. For this experiment 1800 pairs of articles are used, 900 with related articles linking the articles and 900 without. The task is to classify whether an article pair is related or not and accuracy is used for evaluation. Following the STS Evaluation Framework (section 3.2.3), the requirements for this STS based application are *(1:1, All, Classification)*, the same as the first experiment and thus the STS Evaluation Framework recommends the same evaluation method, the F1_hmean. Table 8, similar to Table 7, depicts the comparison of intrinsic and extrinsic evaluation of STS measures.

The accuracy as well as the rank of each STS measure can be seen in Table 9. The Weka toolkit is used and as a classifier the R1 (Holte, 1993) with a bucket size of 50 and 10-fold cross-validation is employed.

	MAD		MSD		Spearman	
	Score	Rank	Score	Rank	Score	Rank
PCC_Rank	4.50	11	25.50	10	0.198	11
SCC_Rank	4.50	11	26.21	12	0.172	12
nDCG_All_Rank	4.36	10	25.79	11	0.238	10
nDCG_Avg_Rank	4.14	7	22.86	7	0.380	6
nCG_Avg_Rank	4.14	7	22.86	7	0.380	6
Acc_low_Rank	4.64	13	29.93	14	0.057	14
F1_low_Rank	2.36	1	9.36	2	0.717	2
Acc_high_Rank	4.79	14	29.79	13	0.062	13
F1_high_Rank	5.36	17	35.79	17	-0.123	17
Acc_macro_Rank	4.79	14	30.79	15	0.031	15
Acc_hmean_Rank	4.79	14	30.79	15	0.031	15
F1_macro_Rank	3.64	5	18.07	5	0.422	5
F1_hmean_Rank	2.36	1	8.79	1	0.722	1
PCC_nCG_hmean_Rank	4.07	6	20.93	6	0.339	8
SCC_nCG_hmean_Rank	4.21	9	22.93	9	0.277	9
PCC_F1_hmean_Rank	2.50	3	9.93	3	0.686	3
SCC_F1_hmean_Rank	2.79	4	12.79	4	0.594	4

Table 8: Comparison of the ranking of STS measures of the Related Article Pairs experiment (Table 6) and Intrinsic Ranking (Table 4) using mean absolute difference (MAD), mean square difference (MSD) and Spearman coefficient

According to Table 8, F1_hmean is the evaluation method that resembles the intrinsic evaluation the closest, which is in line with the prediction of the STS Evaluation Framework. F1_low and PCC_F1_hmean take second and third place. All accuracy measures, as well as PCC and SCC are placed within the lowest third of the ranking. A discussion of the results follows in section 4.4.2.

STS Measures	Accuracy	Rank
GreedyStringTiling 3	0.52	13
LongestCommon SubsequenceComparator	0.55	12
LongestCommon SubsequenceNormComparator	0.48	14
LongestCommonSubstring Comparator	0.64	7
WordNGramContainment Measure 1 stopword.filtered	0.60	11
WordNGramContainment Measure 2 stopword filtered	0.63	10
WordNGramJaccardMeasure 1	0.68	4
WordNGramJaccardMeasure 3	0.66	5
WordNGramJaccardMeasure 4	0.63	9
WordNGramJaccardMeasure 2 stopword filtered	0.69	3
WordNGramJaccardMeasure 4 stopword filtered	0.63	8
CharacterNGramMeasure 2	0.66	6
CharacterNGramMeasure 3	0.74	2
CharacterNGramMeasure 4	0.77	1

Table 9: The performance and ranking of the STS measures for Extrinsic Evaluation II a: Related Article Pairs

4.3.3 Extrinsic Evaluation II b: Related Article Sets

The goal of the third experiment is to select related articles from a set of articles, which resembles the task to recommend an article and, therefore, is much closer to an actual STS based application. For the third extrinsic task the ZEIT Corpus is used as well. As categories for the articles Politics | Germany is used again. Because the source article and the related article are supposed to be from the same category, such a preselection can be used. In the actual STS based application, the set of articles would be all articles in the same category.

100 articles are randomly chosen and for each article a set of 20 articles is added. Out of that set, one article has a link to the source article. The other 19 articles do not have a link, not even if intermediary articles are considered. That means if the ZEIT Corpus would be interpreted as a graph, no path of any length between the source article and the 19 not recommended articles exists. We limited the number of unrelated articles to 19 due to performance reasons. The task is to identify out of the set of 20 pairs the one pair which represents a pair of related articles. Therefore the requirement combination is *(1:n, K-best, Rank)* and the STS Evaluation Framework recommends STS measures with high scores for SCC_nCG_hmean or nDCG_Avg. SCC_nCG_hmean is the harmonic mean of the Spearman correlation and the normalized cumulative gain without a discount factor for ranks and nDCG_Avg is the mean of the normalized discounted cumulative gain with different sizes of the used set.

For the evaluation in each set the STS between the source article and the 20 added articles were calculated. The assumption is that the higher the STS score, the higher the similarity is. Therefore the article with the highest STS score when compared to the source article was selected as the related article. If the article with the highest STS score actually was the related article when compared to the gold standard, this outcome was considered a success. Otherwise it was considered a failure. That process was done for all of the 100 source articles and the accuracy was calculated. Table 10 shows the accuracy values as well as the rankings of the STS measures employed in the extrinsic task.

STS Measures	Accuracy	Rank
GreedyStringTiling 3	0.12	13
LongestCommon SubsequenceComparator	0.19	11
LongestCommon SubsequenceNormComparator	0.06	14
LongestCommonSubstring Comparator	0.46	3
WordNGramContainment Measure 1 stopword.filtered	0.19	11
WordNGramContainment Measure 2 stopword filtered	0.28	10
WordNGramJaccardMeasure 1	0.44	5
WordNGramJaccardMeasure 3	0.38	8
WordNGramJaccardMeasure 4	0.43	6
WordNGramJaccardMeasure 2 stopword filtered	0.46	3
WordNGramJaccardMeasure 4 stopword filtered	0.43	6
CharacterNGramMeasure 2	0.33	9
CharacterNGramMeasure 3	0.62	2
CharacterNGramMeasure 4	0.67	1

Table 10: The performance and ranking of the STS measures for Extrinsic Evaluation II b: Related Article Sets

Table 11 compares the ranking of the STS measures evaluated internally and externally. The highest resemblance of the ranking of intrinsic and extrinsic evaluation has F1_low, followed by F1_hmean and PCC_F1_hmean. SCC_nCG_hmean, one of the two proposed evaluation methods, ranks at place 10. The other proposed evaluation method, nDCG_Avg, ranks place five which is better, but still only in the upper end of the middle third of the ranking. PCC and SCC score even worse than SCC_nCG_hmean. A discussion of the results follows in section 4.4.3.

	MAD		MSD		Spearman	
	Score	Rank	Score	Rank	Score	Rank
PCC_Rank	5.00	11	33.71	11	-0.031	11
SCC_Rank	5.00	11	34.00	12	-0.040	12
nDCG_All_Rank	4.43	8	25.57	8	0.238	8
nDCG_Avg_Rank	4.07	5	23.79	6	0.338	5
nCG_Avg_Rank	4.07	5	23.79	6	0.338	5
Acc_low_Rank	5.14	13	34.43	13	-0.062	13
F1_low_Rank	2.86	1	13.29	2	0.611	2
Acc_high_Rank	5.29	14	35.86	14	-0.102	14
F1_high_Rank	5.86	17	41.57	17	-0.283	17
Acc_macro_Rank	5.29	14	36.14	15	-0.113	15
Acc_hmean_Rank	5.29	14	36.14	15	-0.113	15
F1_macro_Rank	4.14	7	23.29	5	0.289	7
F1_hmean_Rank	3.00	2	13.00	1	0.614	1
PCC_nCG_hmean_Rank	4.57	9	28.57	9	0.130	9
SCC_nCG_hmean_Rank	4.71	10	30.14	10	0.084	10
PCC_F1_hmean_Rank	3.14	3	15.71	3	0.536	3
SCC_F1_hmean_Rank	3.43	4	18.71	4	0.439	4

Table 11: Comparison of the ranking of STS measures of the Related Article Sets experiment (Table 6) and Intrinsic Ranking (Table 4) using mean absolute difference (MAD), mean square difference (MSD) and Spearman coefficient

4.4 Discussion

The goal of the three experiments (Text Reuse, Related Article Pairs, and Related Article Sets) was to test the STS Evaluation Framework. The rational of the STS Evaluation Framework is that it should be possible to predict the performants of STS measures when used within a STS based applications. The prediction is based on the performance of STS measures when evaluated intrinsically with an evaluation method that resembles the requirements of the STS based application. This section discusses the results of the three experiments.

4.4.1 Assessment of Extrinsic Evaluation I

For the Text Reuse experiment, the evaluation method that should resemble the performance of STS measures within an application best according to the STS Evaluation Framework was the F1_hmean. In the experiment three (according to MAD and MSD) or four (according to SCC) other rankings resemble the outcome of the extrinsic task better than the F1_hmean (F1_low, nDCG_All, nDCG_Avg, and nCG_Avg).

The actual differences in scores between the F1_low evaluation (the best predictor) and the F1_hmean (the predictor suggested by the STS Evaluation Framework) of MAD (0.29), MSD (1.87), and SCC (0.07) are quite small (see Table 7). It can therefore be argued, that the F1_hmean is a predictor which is about as good as the four evaluation methods that perform marginally better. The PCC, on the other hand, performs notably worse than the top predictor with differences in scores to the F1_low of MAD (2.15), MSD (26), and SCC (0.823). Values for the SCC are similar.

In addition, the top three performing STS measures according to the evaluation methods are compared to the top three performing STS measures according to the three extrinsic evaluation tasks. The rationale is that the goal of the STS Evaluation Framework is to enable practitioners to select the best STS measures for their task. Hence predicting the top performing STS measures correctly is more important for practitioners than ordering STS measures that do not perform well within the specific task. The F1_low as well as the F1_hmean both rank the CharacterNGramMeasure (2, 3, and 4) the highest, while the nDCG_All, nDCG_Avg, and nCG_Avg rank the WordNGramJaccardMesure (3, 4, and 4 with stopword filter) the highest (see Table 4). Within the extrinsic evaluation the CharacterNGramMeasure 4, the WordNGramJaccardMeasure 3 and the LongestCommonSubstringComperator are the top ranked STS measures (see Table 6). The F1_hmean (suggested by the STS Evaluation Framework) as well as the F1_low, nDCG_All,

nDCG_Avg, and nCG_Avg (the four measures that perform better than the F1_hmean) are only able to predict one out of the three best suited STS measures for the Text Reuse test correctly. It can again be argued that the F1_hmean is a similar good predictor as the other four evaluation methods, which supports the suggestion of STS Evaluation Framework. On the other hand was it only possible to predict one out of three STS measures with the best performance, which does not support the claim of the STS Evaluation Framework.

In section 4.1 three questions were stated that these experiments should answer. Are the PCC or SCC good predictor? Are the evaluation methods suggested by the STS Evaluation Framework good predictors? And can the evaluation method selected by the STS Evaluation Framework predict the best performing STS measures? For this experiment, the first two questions can be answered positively, while the last question needs to be denied.

4.4.2 Assessment of Extrinsic Evaluation II a

For the second experiment, the Related Article Pairs, the F1_hmean was suggested by the STS Evaluation Framework as the evaluation method that should predict the performance of STS measures within the application the best. Table 8 supports this with values for MAD of 2.36, for MSD of 8.79, and a SCC of 0.722, which are the best in this experiment. Similar to the previous experiment, the prediction for STS measures of the three best performing evaluation method will be compared. The three best evaluation methods are F1_hmean, F1_low, and PCC_F1_hmean. According to all three evaluation methods, the CharacterNGramMeasures (2, 3, and 4) are the best STS measures (see Table 4). The best STS measures according to the experiment (see Table 6) are CharacterNGramMeasures (3, and 4) as well as the WordNGramJaccardMeasure (2 with stopword filter). The F1_hmean could predict two out of the three best performing STS measures for this experiment correctly, which supports the claim of the STS Evaluation Framework. Therefore all three questions asked in section 4.1 can be answered positively.

4.4.3 Assessment of Extrinsic Evaluation II b

For the third experiment, the Related Article Sets, the STS Evaluation Framework suggests two evaluation methods, the SCC_nCG_hmean and the nDCG_Avg. Similar to the Related Article Pairs experiment, the actual best predictions for STS measures according to MAD, MSD, and SCC are F1_hmean, F1_low, and PCC_F1_hmean. SCC_nCG_hmean only ranks at place 10 out of 17 and nDCG_Avg at place 5 in Table 11 with differences in scores of MAD

(1.21), MSD (10.79), and SCC (0.276). This differences are much higher than the values in the Text Reuse experiment (section 4.4.1).

The PCC performs even worse than the nDCG_Avg with differences in scores to the best performing evaluation methods of MAD (2.14), MSD (20.71), and SCC (0.645). Values for the SCC are similar.

Again, the top three performing STS measures will be compared to the results of the evaluation methods. The three best STS measures for this extrinsic task (see Table 6) are the same as for the Related Article Pairs experiment: the CharacterNGramMeasures (3, and 4), and the WordNGramJaccardMeasure (2 with stopword filter). The the SCC_nCG_hmean predicts two of them correctly, CharacterNGramMeasures (3, and 4), the nDCG_Avg predicts none of them correctly.

Of the three questions asked in section 4.1, for the nDCG_Avg the first can be answered positively. The second question is controversial, because other evaluation methods performed much better. The last question must be denied. For the SCC_nCG_hmean, the first question needs to be denied, because the performance is only marginally better than the performance of PCC and SCC. The second question needs to be denied as well, because 10 out of 17 evaluation methods performed better. But the last question can be answered positively, because it was able to predict two out of three top performing measures correctly, although nine other evaluation methods in this experiment were able to do that as well.

4.4.4 Conclusion

In section 4.1 three questions are stated that the experiments should answer and which in turn would lead to a validation of the STS Evaluation Framework. For the first question, are the PCC or SCC good predictors, the framework was able to select an evaluation method for all three experiments that performed better than the PCC or SCC. For the second question, does the STS Evaluation Framework help to select a good predictor, for two experiments the framework was able to select a good predictor. For the Extrinsic Evaluation II b experiment (section 4.4.3), one evaluation method was mediocre and one was below average. The last question, can the evaluation method selected by the STS Evaluation Framework predict the best performing STS measures, is not as easy to answer. In the first experiment only one of the three top performing STS measures could be identified. In the second and third, two out of the three top performing STS measures could be identified, while one evaluation method used in the third experiment was not able to identify any of the top performing STS measures.

The evaluation methods proposed by the STS Evaluation Framework are in all cases more capable to predict the performance of STS measures than the PCC or SCC and were able to predict the performance of STS measures well compared to other evaluation methods. But the framework cannot consistently predict the top performing STS measures used within applications, based on the intrinsic performance of the STS measures. The experiments, therefore, only partly support the claim of the STS Evaluation Framework. The following section will discuss possible reasons for this result.

4.4.5 General Discussion of the Experiments

The three extrinsic experiments could not proof the claim of the STS Evaluation Framework completely and, thereby, could not falsify the second research question (see section 2.4), which is most relevant for practitioners. The second research question is based on the theory that different STS based applications have different requirements on STS measures and, therefore, different properties of STS measures are important to different STS based applications. Furthermore, that it is possible to align requirements of STS based applications to evaluation methods for STS measures. The possible explanations why the experiments do not support the claim of the framework can be split into two categories. Either the issue lies within the design of the experiments and its elements (the compilation of STS measures or the datasets) or with the framework itself (the selection of requirements, the alignment of requirements to evaluation methods, and the before mentioned theory).

Starting with the compilation of STS measures, in the best case, the performance of the STS measures should be on a similar level and differ according to different evaluation methods.

According to Table 5 the performance of the STS measures, however, differs highly according to the evaluation methods. The PCC values, for example, lies between 0.24 and 0.66, the value for F1_low is between 0.01 and 0.66, and the value for F1_hmean ranges from 0.01 and 0.68. The values for nDCG_All on the other hand have more similar values between 0.95 and 0.98. In addition, the CharacterNGramMeasures dominates the STS measures. With n=4 it performs the best according to all three experiments (Table 6) and for two of the experiments n=3 ranks second place. For 10 out of 17 evaluation methods, the first three ranks consist of CharacterNGramMeasures with n = {2, 3, 4}. For additional four measures, two out of the three highest ranks consists of these measures (Table 4). Only cumulative gain based measures don't value these measures as high. Such dominant measures within a small set of measures (3 out of 14) might have influenced the result of the experiments.

The next element of the experiment that will be discussed is the dataset. Table 7, Table 8, and Table 11 show a comparison of rankings of intrinsic evaluation to extrinsic evaluation using the MAD, MSD and SCC. Table 12 does the same comparison for rankings of STS measures between extrinsic evaluations (the three experiments). Although the three experiments have different combinations of requirements on the STS measures, the ranking of the STS measures is very similar. Actually the best predictors for the performance of STS measures are not the intrinsic evaluation methods (Table 4) but the extrinsic experiments (Table 12). Especially the Related Article Pairs and Related Article Sets, which use the same dataset but different tasks, are very similar in ranking. That indicates that the performance of STS measures is highly dependent on the dataset. The intrinsic ranking of STS measures was done using the SemEval 2012 task 6 dataset; using a different dataset for that task might lead to a different result of the experiments. Unfortunately, that limits the value of the framework.

	Text Reuse			**Related Article Pairs**			**Related Article Sets**		
	MAD	**MSD**	**SCC**	**MAD**	**MSD**	**SCC**	**MAD**	**MSD**	**SCC**
Text Reuse	0.00	0.00	1.00	2,57	9,86	.70	1,79	5,64	.82
Related Article Pairs	2,57	9,86	.70	0.00	0.00	1.00	1,21	3,50	.91
Related Article Sets	1,79	5,64	.82	1,21	3,50	.91	0.00	0.00	1.00

Table 12: Comparison of the ranking of STS measures of the three experiments using mean absolute difference (MAD), mean square difference (MSD) and Spearman coefficient (SCC)

The selection of requirements in the STS Evaluation Framework is not based on empirical data, but on logical distinction of known STS based applications. This distinction is supported by a short study in section 3.2.2. However, this study is neither exhaustive nor representative. It is therefore possible, that the selection of requirement is incorrect or at least not complete.

The alignment of requirements to evaluation methods that is done by the STS evaluation Framework is also not based on empirical data. In section 3.2.3 the arguments for selecting certain evaluation methods for certain combination of requirements are explained. Still, it is quite possible to come to a different and maybe better selection of evaluation methods. Interestingly, the F1_low measure, which is not directly part of the STS Evaluation Framework, ranks on first or second place as a predictor for the performance of STS measures for all three experiments. As explained in section 3.1.2, the F1_low evaluation method is meant to assess the ability of a STS measure to distinguish between very dissimilar text pairs and all other text pairs. The F1_high, on the other hand, which is meant to assess the ability to distinguish between very similar and all other text pairs, is not a good predictor. The combination of both, the F1_hmean, has proved to be a good predictor, independent of the task.

The intention of the three experiments was to support the theory that different STS based applications have different requirements on STS measures and, therefore, different properties of STS measures are important to different STS based applications. This was derived argumentatively in section 3.2. In addition, the theory was that it is possible to align requirements of STS based applications to evaluation methods for STS measures. The three experiments could support this claim partially, but failed to consistently predict the top performing STS measures. Within this section different weaknesses of the experiment setup that could have influenced that outcome were explained.

In the first research question in section 2.4 it was asked, whether the PCC or the SCC are the best indicators for the performance of STS measures. Referring to Table 7, Table 8, and Table 11, the PCC as well as the SCC always perform worse than other evaluation methods as predictors. For predicting the best three STS measures, the PCC and the SCC are average, mainly because of the dominance of the CharacterNGramMeasure. Without this measure, the PCC and SCC would perform below average.

Based on this work, it can be recommended to use an evaluation method for the selection of a STS measure, which resembles the requirements of the STS based application. The STS Evaluation method can be a starting point for this. It can also be advised against solely using the PCC or the SCC as evaluation methods for STS measures, because they performed below average in all three experiments. Which does not mean, that no use cases exist, where the PCC or the SCC are good predictors of the performance of STS measures when used within applications.

5 Conclusion

The intention of this work was to show that a STS system with a higher Pearson correlation value will not always outperform a STS system with a lower Pearson correlation when used in a STS based application. Additionally, it was asked whether a mapping of requirements of STS based applications on STS measures to evaluation methods was possible. Such a mapping would help to predict the performance of STS measures used in applications (see Figure 2).

To answer that question, the work was structured in three parts. In the first part, it was shown that different evaluation methods rank the performance of STS measures differently. As a database, the submissions to the SemEval 2012 task 6 dataset were used (chapter 3.1). For the evaluation common intrinsic evaluation methods like the Pearson correlation, the nDCG, and the F1 measure have been applied. Table 13 depicts the ranking of the submissions according to different evaluation methods and Table 1 shows the mean absolute difference (MAD) in rankings between the different evaluation methods. The task6-JU_CSE_NLPSemantic_Syntactic_Approach submission for example, ranks first place for nDCG and only 82nd out of 88 for the Pearson correlation.

The second part gives an overview on the various requirements STS based applications can have on the STS system. This was shown exemplarily in a study in chapter 3.2.2. Here, three dimensions of requirements were proposed: *Cardinality, Set of Interest,* and *Information*.

In the third part, the STS Evaluation Framework was developed. This framework systematically structures requirements of STS based applications on STS measures with the three before mentioned dimensions. Additionally combinations of these requirements are mapped to intrinsic evaluation methods. The intention of the STS Evaluation Framework is to be able to predict the relative performance of STS measures within applications, based on the requirements of the application. A STS measure that performs well in the intrinsic evaluation using the developed framework should perform well in a specific STS based application (section 3.2). The STS Evaluation Framework was subsequently tested in chapter 4.

To test the framework STS measures, different datasets were used. First all STS measures were evaluated using the SemEval 2012 task 6 dataset and were ranked according to evaluation methods proposed by the STS Evaluation Framework. This represents the intrinsic benchmark. For the three experiments, the STS measures were used in STS based applications with two other datasets. Finally the rankings of the intrinsic benchmark as well as the rankings of the applications were compared.

The framework leads to a better prediction of which STS measures perform well in the STS based applications than using the traditional approach of choosing the measures with the highest Pearson or Spearman correlation in the intrinsic evaluation benchmark.

The evaluation methods proposed by the STS Evaluation Framework could outperform the Pearson correlation as well as the Spearman correlation in all experiments. In two experiments the proposed evaluation method was among the best performing predictors. In the last experiment, two evaluation methods were proposed, one performed above average and one below average. The question most relevant for practitioners, if the evaluation method proposed by the STS Evaluation Framework predicts the top performing STS measures, remains inconclusive. In two experiments, two out of the three top performing STS measures could be identified, in one experiment, only one of the three top performing STS measures could be identified.

Subsequently, the setup of the experiments were discussed (section 4.4.5). According to the STS Evaluation Framework, the performance of STS measures within STS based applications should be dependent on the requirement of the application. Contradictory, the character-n-gram ($n = 4$) performed best for all three experiments, although, the requirements differed for the experiments. In addition, the ranking of the performance of STS measures for two experiments with *different* requirements but the *same* dataset resembled each other strongly.

That leads to the conclusion that a STS measure can perform well within STS based applications, independent from the requirements of the application. And that the dataset, which the application uses, can be very influential on the performance of STS measures.

Nevertheless, the Pearson correlation as well as the Spearman correlation proved to be bad indicators for the performance of STS measures within applications for all three experiments. They always performed worse than average as a predictor for the relative performance of STS measures. When used to identify the three best performing STS measures, they performed average. This is mainly because of the dominance of the character-n-gram, which the Pearson correlation and the Spearman correlation rank on the top three places (for three different values of n).

Therefore, it could be shown that a STS measure with a higher Pearson correlation value for the intrinsic evaluation will not always outperform a STS measure with a lower Pearson correlation when used in a STS based application. Hence, for practitioners it can be recommended, not to overestimate the significance of the Pearson correlation as well as the Spearman correlation when picking a STS measure for an application. The STS Evaluation

Framework for alternative evaluation methods proved to be consistently better than the Pearson correlation as well as the Spearman correlation.

The conclusion of this work leads to different questions that could be answered in future work. Concerning the STS Evaluation Framework, it is of interest, whether different mappings of requirements to evaluation methods come to better results. It might be necessary to include more or different evaluation methods in the framework. A more rigorous study about the requirements of STS based applications could help to develop a new framework. In addition, a comparative study of different datasets and STS measures could generate valuable insights about the impact datasets have on the performance of STS measures.

Appendix

	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean
task6-UKP-run2_plus_postprocessing_smt_twsi	1	2	18	16	16	2	2	3	3	2	2	2	2
task6-takelab-syntax	2	5	12	46	46	4	6	5	4	5	5	6	6
task6-takelab-simple	3	1	19	31	31	6	5	2	2	3	3	4	4
task6-UKP-run1	4	6	22	21	22	5	4	4	5	4	4	3	3
task6-UNT-IndividualRegression	5	10	29	61	58	7	8	25	31	18	19	12	12
task6-ETS-PERPphrases	6	4	6	55	54	15	27	8	14	10	9	17	20
task6-ETS-PERP	7	3	7	55	54	14	24	9	15	9	8	16	17
task6-UKP-run3_plus_random	8	16	2	16	16	1	1	1	1	1	1	1	1
task6-UNT-IndividualDecTree	9	11	27	44	44	7	9	21	25	13	15	11	10
task6-SRIUBC-SYSTEM2	10	9	23	49	45	17	14	6	8	6	6	7	7
task6-SRIUBC-SYSTEM1	11	8	24	45	43	20	17	7	10	7	7	9	11
task6-UNITOR-2_REGRESSION_ALL_FEATURES	12	13	53	78	78	20	18	23	32	20	21	19	18
task6-UNITOR-1_REGRESSION_BEST_FEATURES	13	12	51	78	78	23	19	24	35	21	23	20	19
task6-UNT-CombinedRegression	14	21	32	62	60	9	10	39	40	28	30	15	14
task6-SOFT-CARDINALITY	15	29	42	30	30	3	3	11	12	7	10	5	5
task6-UIUC-MLNLP-CCM	16	14	20	12	12	11	7	12	19	11	11	8	8
task6-University_Of_Sheffield-Machine_Learning	17	7	16	9	10	27	28	10	17	12	12	21	22
task6-UMCC_DLSI-MultiSemLex	18	17	31	34	34	12	12	28	30	19	24	13	13
task6-SOFT-CARDINALITY-ONE-FUNCTION	19	18	54	66	69	13	20	19	16	14	14	14	15
task6-weiwei-run1	20	33	45	59	61	26	15	84	85	79	80	79	82
task6-SRIUBC-SYSTEM3	21	42	28	32	32	10	11	22	23	16	17	10	9
task6-LIMSI-gradtree	22	27	8	28	27	20	37	16	21	17	16	23	24
task6-sbdlrhmn-Run1	23	30	21	15	15	18	13	47	50	31	34	22	21
task6-sranjans-2	24	19	44	51	50	37	38	35	45	34	32	32	32
task6-BUAP-RUN-3	25	28	25	38	38	25	49	14	9	15	13	25	36
task6-UMCC_DLSI-MultiLex	26	32	48	76	75	16	16	32	37	26	27	18	16
task6-Penn-ELReg	27	34	46	22	23	30	34	38	43	36	37	29	27
task6-Penn-ERReg	28	31	52	23	24	36	39	40	44	39	40	33	33
task6-UMCC_DLSI-MultiSem	29	43	50	54	53	24	22	52	46	42	48	24	23
task6-sranjans-1	30	24	40	67	65	39	35	51	56	45	50	39	37
task6-PolyUCOMP-RUN1	31	22	57	72	73	87	75	84	85	87	87	85	82
task6-FBK-run3	32	20	43	73	70	41	69	19	22	23	22	53	62
task6-Penn-LReg	33	35	47	20	21	40	44	36	41	37	36	35	39
task6-University_Of_Sheffield-Hybrid	34	15	55	8	9	55	42	40	52	48	47	40	40
task6-FBK-run2	35	23	49	74	74	43	70	27	34	29	28	59	64
task6-UOW-LEX_PARA	36	47	15	52	48	19	40	49	47	32	41	37	38
task6-LIMSI-cosprod	37	41	9	33	33	51	87	89	84	80	82	89	82
stanford_fsa	38	25	34	57	56	34	60	43	53	41	43	57	57
task6-SAGAN-RUN3	39	40	38	27	29	33	36	45	38	38	39	28	26
task6-UNITOR-3_REGRESSION_ALL_FEATURES_ALL_DOMAINS	40	49	62	75	77	28	29	45	39	35	38	26	25
task6-UNIBA-RI	41	45	60	70	68	54	85	17	6	27	26	71	80

task6-SAGAN-RUN2	42	37	39	24	25	42	25	56	57	53	53	30	29
task6-DeepPurple-DeepPurple_hierarchical	43	36	58	36	36	58	32	54	58	54	54	38	35
task6-UNIBA-LSARI	44	46	61	71	72	29	41	33	36	30	31	27	28
task6-LIMSI-sumdiff	45	38	11	37	37	73	66	79	77	77	78	77	68
task6-UNIBA-DEPRI	46	52	67	88	88	45	73	18	11	25	25	52	66
task6-yrkakde-JaccNERPenalty	47	26	10	1	1	47	65	13	13	22	18	47	59
task6-University_Of_Sheffield-Vector_Space	48	44	37	19	19	70	45	55	61	58	58	44	42
task6-UOW-LEX_PARA_SEM	49	51	13	39	39	30	62	57	64	49	52	64	60
task6-yrkakde-DiceWordnet	50	39	33	18	18	34	53	42	24	40	42	41	44
task6-aggarwal-run2	51	58	73	53	52	55	26	63	68	60	61	43	41
task6-aggarwal-run1	52	68	65	26	26	49	23	71	72	64	69	45	43
task6-ABBYY-General	53	71	41	29	28	38	43	76	78	72	75	65	58
task6-FBK-run1	54	50	36	63	62	51	87	15	7	24	20	74	82
stanford_rte	55	65	59	69	66	48	30	58	54	55	55	34	31
task6-DeepPurple-DeepPurple_sigmoid	56	59	80	87	87	46	50	68	69	63	64	51	49
task6-SAGAN-RUN1	57	48	72	89	89	58	79	29	28	32	29	63	71
task6-DSS-average	58	56	77	57	56	62	47	61	66	62	60	48	47
task6-UOW-SEM	59	53	4	14	14	44	59	48	48	47	49	54	54
task6-DSS-alignheuristic	60	54	66	48	49	74	58	52	51	57	57	55	55
task6-DSS-wordsim	61	76	69	35	35	60	33	75	75	74	74	50	48
task6-sranjans-3	62	61	89	81	84	66	55	72	71	70	72	61	53
task6-BUAP-RUN-1	63	62	70	64	67	65	63	70	73	69	71	70	63
task6-ATA-CHNK	64	80	26	10	11	57	31	64	60	61	63	36	34
task6-SAARLAND-ALIGN_VSSIM	65	77	56	1	1	87	75	84	85	87	87	85	82
task6-DeepPurple-DeepPurple_single	66	55	63	13	13	72	54	26	26	43	33	42	45
task6-aggarwal-run3	67	70	71	77	80	61	80	37	33	46	45	67	73
task6-IRIT-pg3	68	69	82	41	40	66	51	69	70	67	70	58	51
task6-UNED-SP_INIST	69	75	87	82	82	82	56	83	83	82	81	82	79
task6-UIUC-MLNLP-Blend	70	82	17	50	47	30	21	66	62	59	62	31	30
task6-SAARLAND-MIXT_VSSIM	71	64	83	65	64	86	84	84	85	86	86	88	82
task6-tiantianzhu7-1	72	57	74	85	85	75	82	30	27	52	44	69	75
task6-ETS-TERp	73	60	35	1	1	81	64	77	79	81	79	78	67
task6-tiantianzhu7-3	74	66	68	83	83	71	71	34	29	51	46	60	65
task6-UNED-H34measures	75	72	64	84	76	78	57	60	59	73	68	56	52
task6-IRIT-pg1	76	78	85	41	40	63	61	67	67	65	65	66	61
stanford_pdaAll	77	73	78	80	81	51	87	58	49	56	56	80	82
task6-sbdlrhmn-Run2	78	74	79	60	63	77	72	73	74	75	73	76	72
task6-ATA-STAT	79	81	14	1	1	76	48	62	55	71	67	46	46
task6-tiantianzhu7-2	80	63	75	85	85	68	83	31	20	44	35	73	78
task6-IRIT-wu	81	79	88	41	40	69	74	65	63	66	66	72	69
task6-JU_CSE_NLP-Semantic_Syntactic_Approach	82	67	1	11	7	79	78	44	42	67	59	68	70
task6-ATA-BASE	83	87	5	1	1	64	46	78	76	76	77	62	56
task6-janardhan-UNL_matching	84	83	3	1	1	85	67	80	80	85	85	81	74
task6-UIUC-MLNLP-Puzzle	85	88	30	25	20	80	52	74	65	78	76	49	50
task6-EHU-RUN1v2	86	85	86	40	59	83	68	81	82	83	83	83	76
task6-Baseline	87	84	84	47	51	87	75	84	85	87	87	85	82
task6-UNED-HallMeasures	88	86	76	68	71	50	86	50	18	50	51	75	81
task6-BUAP-RUN-2	89	89	81	7	8	84	81	82	81	84	84	84	77

Table 13: SemEval 2012 task 6 different evaluation methods ordered by PCC (88 runs of 35 teams plus baseline)

Mean Square Difference	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean
PCC	0.0	71.7	670.1	1318.2	1296.2	168.9	328.8	400.8	537.8	258.1	304.7	269.9	295.4
SCC	71.7	0.0	688.7	1340.0	1319.4	307.6	521.8	333.1	476.7	258.4	277.7	378.2	420.6
nDCG_All	670.1	688.7	0.0	629.2	586.9	650.2	757.8	797.8	843.1	745.1	763.0	698.8	694.3
nDCG_Avg	1318.2	1340.0	629.2	0.0	7.4	1324.2	1051.1	1398.8	1434.6	1398.5	1404.3	1133.4	1051.0
nCG_Avg	1296.2	1319.4	586.9	7.4	0.0	1302.9	1032.1	1373.9	1411.2	1375.0	1379.9	1103.7	1024.9
Acc_low	168.9	307.6	650.2	1324.2	1302.9	0.0	279.0	371.4	438.4	184.8	245.6	192.2	229.0
F1_low	328.8	521.8	757.8	1051.1	1032.1	279.0	0.0	727.7	848.3	545.0	604.2	142.9	96.6
Acc_high	400.8	333.1	797.8	1398.8	1373.9	371.4	727.7	0.0	38.2	48.5	21.4	348.8	474.3
F1_high	537.8	476.7	843.1	1434.6	1411.2	438.4	848.3	38.2	0.0	100.2	68.8	440.6	581.3
Acc_macro	258.1	258.4	745.1	1398.5	1375.0	184.8	545.0	48.5	100.2	0.0	8.5	226.5	329.5
Acc_hmean	304.7	277.7	763.0	1404.3	1379.9	245.6	604.2	21.4	68.8	8.5	0.0	261.3	373.8
F1_macro	269.9	378.2	698.8	1133.4	1103.7	192.2	142.9	348.8	440.6	226.5	261.3	0.0	21.0
F1_hmean	295.4	420.6	694.3	1051.0	1024.9	229.0	96.6	474.3	581.3	329.5	373.8	21.0	0.0

Table 14: Mean Square Difference between ranks of submissions to SemEval 2012 task 6

SCC	PCC	SCC	nDCG_All	nDCG_Avg	nCG_Avg	Acc_low	F1_low	Acc_high	F1_high	Acc_macro	Acc_hmean	F1_macro	F1_hmean
PCC	1.00	0.95	0.49	0.02	0.03	0.87	0.75	0.69	0.59	0.80	0.77	0.79	0.76
SCC	0.95	1.00	0.48	0.00	0.02	0.77	0.60	0.74	0.63	0.80	0.79	0.71	0.67
nDCG_All	0.49	0.48	1.00	0.53	0.56	0.51	0.42	0.39	0.36	0.43	0.42	0.47	0.46
nDCG_Avg	0.02	0.00	0.53	1.00	0.99	0.01	0.21	-0.05	-0.08	-0.05	-0.05	0.15	0.20
nCG_Avg	0.03	0.02	0.56	0.99	1.00	0.03	0.23	-0.03	-0.06	-0.03	-0.03	0.18	0.22
Acc_low	0.87	0.77	0.51	0.01	0.03	1.00	0.79	0.72	0.67	0.86	0.81	0.86	0.82
F1_low	0.75	0.60	0.42	0.21	0.23	0.79	1.00	0.44	0.35	0.59	0.54	0.89	0.92
Acc_high	0.69	0.74	0.39	-0.05	-0.03	0.72	0.44	1.00	0.97	0.96	0.98	0.74	0.64
F1_high	0.59	0.63	0.36	-0.08	-0.06	0.67	0.35	0.97	1.00	0.92	0.95	0.67	0.55
Acc_macro	0.80	0.80	0.43	-0.05	-0.03	0.86	0.59	0.96	0.92	1.00	0.99	0.83	0.75
Acc_hmean	0.77	0.79	0.42	-0.05	-0.03	0.81	0.54	0.98	0.95	0.99	1.00	0.80	0.71
F1_macro	0.79	0.71	0.47	0.15	0.18	0.86	0.89	0.74	0.67	0.83	0.80	1.00	0.98
F1_hmean	0.76	0.67	0.46	0.20	0.22	0.82	0.92	0.64	0.55	0.75	0.71	0.98	1.00

Table 15: Spearman correlation coefficient between ranks of submissions to SemEval 2012 task 6

Bibliography

Aggarwal, Charu C, and Zhai, ChengXiang. 2012. A survey of text clustering algorithms. In *Mining Text Data*, 77-128: Springer.

Agirre, Eneko, Diab, Mona, Cer, Daniel, and Gonzalez-Agirre, Aitor. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. Paper presented at *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*.

Agirre, Eneko, Cer, Daniel, Diab, Mona, Gonzalez-Agirre, Aitor, and Guo, Weiwei. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. Paper presented at *In* SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.

Allison, Lloyd, and Dix, Trevor I. 1986. A bit-string longest-common-subsequence algorithm. *Information Processing Letters* 23:305-310.

Anscombe, Francis J. 1973. Graphs in statistical analysis. *The American Statistician* 27:17-21.

Attali, Yigal, and Burstein, Jill. 2006. Automated essay scoring with e-rater® V. 2. *The Journal of Technology, Learning and Assessment* 4.

Bär, Daniel. 2013. A Composite Model for Computing Similarity Between Texts, Technische Universität Darmstadt: PhD Thesis.

Bär, Daniel, Zesch, Torsten, and Gurevych, Iryna. 2013. DKPro Similarity: An Open Source Framework for Text Similarity. Paper presented at *ACL (Conference System Demonstrations)*.

Barrón-Cedeño, Alberto, Vila, Marta, Martí, M Antònia, and Rosso, Paolo. 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics* 39:917-947.

Barzilay, Regina, and Elhadad, Michael. 1999. Using lexical chains for text summarization. *Advances in automatic text summarization*:111-121.

Ben-Simon, Anat, and Bennett, Randy Elliot. 2007. Toward More Substantively Meaningful Automated Essay Scoring. *The Journal of Technology, Learning and Assessment* 6.

Callison-Burch, Chris, Fordyce, Cameron, Koehn, Philipp, Monz, Christof, and Schroeder, Josh. 2007. (Meta-) evaluation of machine translation. Paper presented at *Proceedings of the Second Workshop on Statistical Machine Translation*.

Chen, David L, and Dolan, William B. 2011. Collecting highly parallel data for paraphrase evaluation. Paper presented at *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*.

Clough, Paul, and Stevenson, Mark. 2011. Developing a corpus of plagiarised short answers. *Language Resources and Evaluation* 45:5-24.

Dagan, Ido, Glickman, Oren, and Magnini, Bernardo. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, 177-190: Springer.

Davide, Magatti, and Fabio, Stella. 2012. Probabilistic Topic Discovery and Automatic Document Tagging. In *Quantitative Semantics and Soft Computing Methods for the Web: Perspectives and Applications*, eds. F. Brena Ramon and Guzman-Arenas Adolfo, 25-49. Hershey, PA, USA: IGI Global.

Dolan, Bill, Quirk, Chris, and Brockett, Chris. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. Paper presented at *Proceedings of the 20th international conference on Computational Linguistics*.

Eneko Agirre, Enrique Amigó. In prep. Exploring Evaluation Measures for Semantic Textual Similarity. *Unpublished manuscript*.

Foltz, Peter W, Laham, Darrell, and Landauer, Thomas K. 1999. Automated essay scoring: Applications to educational technology. Paper presented at *World Conference on Educational Multimedia, Hypermedia and Telecommunications*.

Galliers, Julia Rose, and Jones, K Sparck. 1993. Evaluating natural language processing systems.

Gupta, Vishal, and Lehal, Gurpreet Singh. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence* 2:258-268.

Gusfield, Dan. 1997. *Algorithms on strings, trees and sequences: computer science and computational biology*: Cambridge University Press.

Hall, Mark, Frank, Eibe, Holmes, Geoffrey, Pfahringer, Bernhard, Reutemann, Peter, and Witten, Ian H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11:10-18.

Hassanzadeh, Oktie, Kementsietsidis, Anastasios, Lim, Lipyeow, Miller, Renée J, and Wang, Min. 2009. A framework for semantic link discovery over relational data. Paper presented at *Proceedings of the 18th ACM conference on Information and knowledge management*.

Hauke, Jan, and Kossowski, Tomasz. 2011. Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *Quaestiones Geographicae* 30.

He, Jiyin. 2009. Link detection with wikipedia. In *Advances in Focused Retrieval*, 366-373: Springer.

Hliaoutakis, Angelos, Varelas, Giannis, Voutsakis, Epimenidis, Petrakis, Euripides GM, and Milios, Evangelos. 2006. Information retrieval by semantic similarity. *International journal on semantic Web and information systems (IJSWIS)* 2:55-73.

Holte, Robert C. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine learning* 11:63-90.

Hovy, Eduard, and Lin, Chin-Yew. 1998. Automated text summarization and the SUMMARIST system. Paper presented at *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*.

Hovy, Eduard, Marcus, Mitchell, Palmer, Martha, Ramshaw, Lance, and Weischedel, Ralph. 2006. OntoNotes: the 90% solution. Paper presented at *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*.

Huang, Anna. 2008. Similarity measures for text document clustering. Paper presented at *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*.

Huang, Darren Wei Che, Xu, Yue, Trotman, Andrew, and Geva, Shlomo. 2008. Overview of INEX 2007 link the wiki track. In *Focused Access to XML Documents*, 373-387: Springer.

Järvelin, Kalervo, and Kekäläinen, Jaana. 2000. IR evaluation methods for retrieving highly relevant documents. Paper presented at *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*.

Jin, Wei, Srihari, Rohini K, Ho, Hung Hay, and Wu, Xin. 2007. Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques. Paper presented at *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*.

Jones, K Spärck. 1999. Introduction to text summarization. *Advances in Automated Text Summarization*:1-12.

Jurgens, David, Pilehvar, Mohammad Taher, and Navigli, Roberto. 2014. SemEval-2014 Task 3: Cross-Level Semantic Similarity. Paper presented at *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014), Dublin, Ireland*.

Kekäläinen, Jaana. 2005. Binary and graded relevance in IR evaluations—comparison of the effects on ranking of IR systems. *Information processing & management* 41:1019-1033.

Kešelj, Vlado, Peng, Fuchun, Cercone, Nick, and Thomas, Calvin. 2003. N-gram-based author profiles for authorship attribution. Paper presented at *Proceedings of the conference pacific association for computational linguistics, PACLING*.

Knoth, Petr, Zilka, Lukas, and Zdrahal, Zdenek. 2011. Using explicit semantic analysis for cross-lingual link discovery.

Le Cessie, Saskia, and Van Houwelingen, JC. 1992. Ridge estimators in logistic regression. *Applied statistics*:191-201.

Lee, Michael David, Pincombe, BM, and Welsh, Matthew Brian. 2005. An empirical evaluation of models of text document similarity. *Cognitive Science*.

Li, Yuhua, McLean, David, Bandar, Zuhair A, O'shea, James D, and Crockett, Keeley. 2006. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on* 18:1138-1150.

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. Paper presented at *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Lu, Wei, Liu, Dan, and Fu, Zhenzhen. 2009. Csir at inex 2008 link-the-wiki track. In *Advances in Focused Retrieval*, 389-394: Springer.

Lyon, Caroline, Malcolm, James, and Dickerson, Bob. 2001. Detecting short passages of similar text in large document collections. Paper presented at *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Manning, Christopher D, Schütze, Hinrich, and Raghavan, Prabhakar. 2008. Introduction to information retrieval.

Mihalcea, Rada, Corley, Courtney, and Strapparava, Carlo. 2006. Corpus-based and knowledge-based measures of text semantic similarity. Paper presented at *AAAI*.

Miller, George, and Fellbaum, Christiane. 1998. Wordnet: An electronic lexical database: MIT Press Cambridge.

Miller, George A, and Charles, Walter G. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes* 6:1-28.

Milne, David, and Witten, Ian H. 2008. Learning to link with wikipedia. Paper presented at *Proceedings of the 17th ACM conference on Information and knowledge management*.

Mohler, Michael, and Mihalcea, Rada. 2009. Text-to-text semantic similarity for automatic short answer grading. Paper presented at *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Morris, Charles William. 1938. *Foundations of the Theory of Signs*.vol. 1: University of Chicago Press.

Nenkova, Ani, Maskey, Sameer, and Liu, Yang. 2011. Automatic summarization. Paper presented at *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*.

Potthast, Martin, Gollub, Tim, Hagen, Matthias, Kiesel, Johannes, Michel, Maximilian, Oberländer, Arnd, Tippmann, Martin, Barrón-Cedeno, Alberto, Gupta, Parth, and Rosso, Paolo. 2012. Overview of the 4th International Competition on Plagiarism Detection. Paper presented at *CLEF (Online Working Notes/Labs/Workshop)*.

Powers, David Martin. 2011. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies* 2:37-63.

Real, Raimundo, and Vargas, Juan M. 1996. The probabilistic basis of Jaccard's index of similarity. *Systematic biology*:380-385.

Rubenstein, Herbert, and Goodenough, John B. 1965. Contextual correlates of synonymy. *Communications of the ACM* 8:627-633.

Rubin, Timothy N, Chambers, America, Smyth, Padhraic, and Steyvers, Mark. 2012. Statistical topic models for multi-label document classification. *Machine Learning* 88:157-208.

Salton, Gerard, and McGill, Michael J. 1983. Introduction to modern information retrieval.

Song, Wei, Li, Cheng Hua, and Park, Soon Cheol. 2009. Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures. *Expert Systems with Applications* 36:9095-9104.

Stein, Benno, Eissen, Sven Meyer zu, and Potthast, Martin. 2007. Strategies for retrieving plagiarized documents. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 825-826. Amsterdam, The Netherlands: ACM.

Steinbach, Michael, Karypis, George, and Kumar, Vipin. 2000. A comparison of document clustering techniques. Paper presented at *KDD workshop on text mining*.

Strehl, Alexander, Ghosh, Joydeep, and Mooney, Raymond. 2000. Impact of similarity measures on web-page clustering. Paper presented at *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*.

Tatu, Marta, and Moldovan, Dan. 2005. A semantic approach to recognizing textual entailment. Paper presented at *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Valenti, Salvatore, Neri, Francesca, and Cucchiarelli, Alessandro. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research* 2:319-330.

van Rijsbergen, CJ. 1979. *Information Retrieval (2nd edit.)* London, UK: Butterworths.

Wise, Michael J. 1996. YAP3: Improved detection of similarities in computer program and other texts. Paper presented at *ACM SIGCSE Bulletin*.

Zesch, Torsten. 2010. Study of Semantic Relatedness of Words Using Collaboratively Constructed Semantic Resources, TU Darmstadt.

Zu Eissen, Sven Meyer, and Stein, Benno. 2006. Intrinsic plagiarism detection. In *Advances in Information Retrieval*, 565-569: Springer.