Composing Measures for Computing Text Similarity

Technical Report TUD-CS-2015-0017
January, 2015

Daniel Bär SAP SE, Walldorf, Germany
Torsten Zesch Language Technology Lab, University of Duisburg-Essen
Iryna Gurevych UKP Lab, Technische Universität Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Abstract

We present a comprehensive study of computing similarity between texts. We start from the observation that while the concept of similarity is well grounded in psychology, text similarity is much less well-defined in the natural language processing community. We thus define the notion of text similarity and distinguish it from related tasks such as textual entailment and near-duplicate detection. We then identify multiple text dimensions, i.e. characteristics inherent to texts that can be used to judge text similarity, for which we provide empirical evidence.

We discuss state-of-the-art text similarity measures previously proposed in the literature, before continuing with a thorough discussion of common evaluation metrics and datasets. Based on the analysis, we devise an architecture which combines text similarity measures in a unified classification framework. We apply our system in two evaluation settings, for which it consistently outperforms prior work and competing systems: (a) an intrinsic evaluation in the context of the Semantic Textual Similarity Task as part of the Semantic Evaluation (SemEval) exercises, and (b) an extrinsic evaluation for the detection of text reuse.

As a basis for future work, we introduce DKPro Similarity, an open source software package which streamlines the development of text similarity measures and complete experimental setups.

1 Introduction

Within the natural language processing (NLP) community, similarity between texts (text similarity, henceforth) is a ubiquitous notion and utilized in a wide range of tasks such as question answering (Lin and Pantel, 2001), automatic essay grading (Attali and Burstein, 2006), or paraphrase recognition (Dolan et al., 2004). However, text similarity is often used as an umbrella term covering quite different phenomena—as opposed to the notion of similarity in psychology, which is well studied and captured in formal models such as the set-theoretic model (Tversky, 1977) or the geometric model (Widdows, 2004). We argue that the seemingly simple question "How similar are two texts?" cannot be answered independently from asking what properties make them similar. Goodman (1972) gives a good example for physical objects regarding the situation of a baggage check at the airport: While a spectator might compare bags by shape, size, or color, the pilot only focuses on a bag's weight, and a passenger compares bags by just destination and ownership. Similarly, texts also have particular inherent properties that need to be considered in any attempt to judge their similarity (Bär et al., 2011). Take for example two novels by the famous 19th century Russian writer Leo Tolstoy. A reader may readily argue that these novels are completely dissimilar due to different plots, people, or places. On the other hand, a second reader (e.g. a scholar overseeing texts of disputed authorship) may argue that both texts are indeed highly similar because of their stylistic similarity. In consequence, text similarity remains a loose notion unless we provide a frame of reference. We argue that text similarity cannot be seen as a fixed, axiomatic notion. Rather, we need to define in what way two texts are similar.

From a human-centered perspective, we say that text similarity is a function between two texts t_1 and t_2 which can be informally characterized by the readers' shared view on the text characteristics along which similarity is to be judged. However, to the best of our knowledge the definition of appropriate text characteristics for text similarity computation has not been tackled yet in any previous research. We thus further argue that text similarity can be judged along different text dimensions, i.e. groups of text characteristics which are perceived by humans and for which we provide empirical evidence. For example, a scholar in digital humanities may be less interested in texts that share similar contents—as opposed to e.g. near-duplicate detection (see Section 2)—but may rather be looking for text pairs which are similar with respect to their style and structure. Throughout this work and in particular in Section 3, we will elaborate on the idea of text dimensions and further discuss suitable dimensions for text similarity tasks.

2 Related Tasks

Text similarity has a close relation to a variety of other tasks in the field of natural language processing. These tasks inherently are highly similar to text similarity, but require additional processing steps, e.g. for recognizing textual entailment (Dagan et al., 2006) which implies further constraints for the similarity computation process.

Textual Entailment

A related task is textual entailment which is defined as the directional relationship between a text T (the text) and a second text H (the hypothesis) where T entails H ($T \Rightarrow H$) "if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people" (Dagan et al., 2006). Typically, if T entails H, T and H are often also highly similar. However, an entailment relationship also holds true if H is not similar to T, but can be logically inferred from T. Textual entailment further differs from text similarity in two significant ways: (a) it is defined as a unidirectional relationship, while text similarity requires a bidirectional similarity relationship to hold between a text pair, and (b) textual entailment operates on binary judgments while text similarity is defined as a continuous notion.

Paraphrase recognition

A task which is also closely related to text similarity is paraphrase recognition (Dolan et al., 2004). A paraphrase comprises two texts t_1, t_2 which are "more or less semantically equivalent"[1], but might differ in their syntactic structure and the degree of details. Chen and Dolan (2011) define an ideal paraphrase as "meaning-preserving" which "must also diverge as sharply as possible in form from the original, while still sounding natural and fluent." Under this definition, paraphrase recognition and text similarity closely resemble each other: Two texts for which a paraphrase relationship holds are—in most cases—naturally also highly similar. However, in cases where e.g. one text is a negation of the second one, the texts would still be highly similar, but would not be paraphrases any more. Additionally, paraphrases typically comprise text pairs up to a sentence length only (Dolan et al., 2004; Knight and Marcu, 2002; Cohn et al., 2008; Chen and Dolan, 2011) while text similarity applies to texts of any length. A key difference between paraphrase recognition and text similarity is that paraphrases are annotated with binary judgments rather than continuous similarity scores.

[1] According to the instructions which were given to the annotators in the study by Dolan et al. (2004)

Task	Content	Structure	Style
Authorship Attribution (Mosteller and Wallace, 1964)			✓
Automatic Essay Scoring (Attali and Burstein, 2006)	✓	✓	✓
Information Retrieval (Manning et al., 2008)	✓	✓	✓
Paraphrase Recognition (Dolan et al., 2004)	✓		
Plagiarism Detection (Potthast et al., 2012)	✓		✓
Question Answering (Lin and Pantel, 2001)	✓		
Short Answer Grading (Leacock and Chodorow, 2003)	✓	✓	✓
Text Categorization (Cavnar and Trenkle, 1994)	✓		
Text Segmentation (Hearst, 1997)	✓	✓	
Text Simplification (Chandrasekar et al., 1996)	✓	✓	
Text Summarization (Barzilay and Elhadad, 1997)	✓	✓	
Word Sense Alignment (Ponzetto and Navigli, 2010)	✓		

Table 1: Classification of common natural language processing tasks with respect to the relevant text dimensions suitable for text similarity computation: content, structure, and style

Near-duplicate detection

Another field of related work is near-duplicate detection. In the context of web search and crawling, text pairs (i.e. typically pairs of web pages) are to be detected which differ only slightly in some small portion of text, e.g. by advertisments or timestamps, or as the result of a revision step (Manku et al., 2007; Hoad and Zobel, 2003). While near-duplicate detection is similar to text similarity in that a bidirectional similarity relationship holds, it differs in what is considered a text: In the context of near-duplicate detection, a text refers to a sequence of arbitrary characters, e.g. HTML source code rather than a natural language text. Prior work thus mainly uses fingerprinting and hashing techniques (Charikar, 2002) rather than methods from natural language processing to find near-duplicates—typically in very large datasets, e.g. a corpus of 1.6 billion web pages (Henzinger, 2006).

Plagiarism detection

Manifold definitions for plagiarism have been proposed: the result of copying an original text and claiming its authorship (Potthast et al., 2012), the "unauthorised use or close imitation" of an original text and claiming its authorship (Hannabuss, 2001), or the "unacknowledged copying of documents" (Joy and Luck, 1999). By these definitions, plagiarism detection and text similarity are clearly similar to each other, as a copied text naturally exhibits a high degree of similarity with the original. However, a central aspect to all the definitions is the act of unacknowledged, unauthorized reuse or copy of an original text which may appear in different forms, e.g. direct word-by-word copies, the reuse of text with only slight changes (paraphrasing), or the omission of citations on referenced text parts (Clough, 2003). High similarity between two texts is thus only an indicator for a further plagiarism analysis (Potthast et al., 2012).

3 Text Dimensions

In this section, we introduce the notion of text dimensions. Text dimensions are characteristics inherent to texts that can be used to judge text similarity. In the first part of this section, we detail the proposal of a conceptual model of text dimensions and discuss its recent developments. In the second part, we give empirical evidence that humans perceive text similarity along the proposed dimensions.

3.1 Identification of Text Dimensions

In the beginning of this article, we argued that text similarity is much less well-defined than the notion of similarity in psychology. We proposed to define text similarity as a function between two texts t_1 and t_2 under given assumptions (see Section 1). A key aspect to our definition is that readers need to share a common view on the text characteristics along which similarity is to be judged. However, to the best of our knowledge no such characteristics have yet been identified in any previous work. In this section, we elaborate on how we can model suitable text characteristics in a framework which is applicable to any text similarity task.

We analyzed common NLP tasks with respect to the relevant text dimensions suitable for text similarity computation and present some examples in Table 1. We identified three major dimensions inherent to texts: content, structure, and style (Bär et al., 2011). Content addresses all topics and their relationships within a text. Structure refers to the internal developments of a given text, i.e. discourse structuring, such as the order of sections. Style refers to grammar, usage, mechanics, and lexical complexity, as proposed in the task of automatic essay scoring (Attali and Burstein,

2006). That task typically not only requires the essay to be about a certain topic (content dimension), but also an adequate style and a coherent structure are necessary. However, a scholar in digital humanities might be interested in texts that are similar to a reference document with respect to style and structure, while texts with similar content are of minor interest. It should be noted that the dimensions in conceptual spaces are not totally independent, but are correlated (Gärdenfors, 2000). For example, the structure dimension covaries with the style dimension, as a particular style inherently leads e.g. to a particular usage pattern of function words. The style dimension in turn covaries with the content dimension, as for example a particular content (such as a newspaper report about a car accident) requires a particular (factual) style. Changes in the content dimension hence inherently often come with changes in the style dimension.

3.2 Empirical Evidence

In Table 1, we identified a number of typical NLP tasks with respect to the relevant text dimensions suitable for text similarity computation. We now ground the proposed text dimensions empirically by reporting on the results of two annotation studies which we conducted in order to show that humans indeed judge similarity along the proposed text dimensions. In these studies, we asked human participants to give insights into the rationales behind their text similarity judgments (Bär et al., 2011). In the first one, we found empirical evidence for the content and structure dimensions, while the second study further grounds the style dimension. Overall, the results show that human annotators indeed distinguish between different text dimensions when they are asked to judge text similarity. In consequence, text similarity cannot be seen as a fixed, axiomatic notion, but rather needs to be put into context, i.e. a thorough definition of relevant text dimensions is fundamental to any text similarity computation.

Content vs. Structure Dimensions

In this study, we used the 50 Short Texts dataset (see Section 5.1.1) by Lee et al. (2005) that contains pairwise human similarity judgments for 1,225 text pairs. We selected a subset of 50 pairs with a uniform distribution of judgments across the whole similarity range. We asked three annotators (A_1-A_3): "How similar are the given texts?" We then computed the Spearman correlation ρ of each annotator's ratings with the gold standard: $\rho_{A_1} = 0.83$, $\rho_{A_2} = 0.65$, and $\rho_{A_3} = 0.85$. The much lower correlation of the annotator A_2 indicates that a different dimension was probably used to judge similarity.

To further investigate this issue, we asked the annotators about the reasons for their judgments. A_1 and A_3 reported consistently that they focused only on the content of the texts intuitively and completely disregarded other characteristics. However, A_2 was also taking structural similarities into account, e.g. two texts were rated highly similar because of the way they are organized: First, an introduction to the topic is given, then a quotation is stated, then the text concludes with a certain reaction of the acting subject. These results indicate that human annotators indeed judge similarity along different text dimensions, as introduced above.

Content vs. Style Dimensions

The annotators in the previous study distinguished between the text dimensions content and structure. The stylistic aspect of the texts was not addressed, as the text pairs were all of similar style, and hence that dimension was not perceived as salient.

In order to further investigate whether the style dimension can be grounded empirically, we conducted a second study: We selected 10 pairs of short texts from Wikipedia (WP) and Simple English Wikipedia[2] (SWP). We used the first paragraphs of Wikipedia articles and the full texts of articles in Simple English to obtain pairs of similar length. We then formed pairs in all combinations (WP-WP, SWP-WP, and SWP-SWP) to ensure that both text dimensions content and style are salient for some pairs. For example, an article from SWP and one from WP about the same topic share the same content, but are different in style, while two articles from SWP have a similar style, but different content.

We then asked three annotators to rate each pair according to the content and style dimensions. A qualitative analysis of the results shows that WP-WP and SWP-SWP pairs are perceived as stylistically similar, while WP-SWP pairs are seen similar with respect to their content. We conclude that human annotators indeed distinguish between the two text dimensions content and style for text pairs where they are discriminating, and thus perceived as salient text characteristics.

3.3 Discussion

In this section, we adopted the theory of conceptual spaces to texts. Based on an analysis of common NLP tasks where text similarity is fundamental, we proposed to focus on three major text dimensions: content, structure, and

[2] Articles written in Simple English use a limited vocabulary and easier grammar than the standard Wikipedia, `http://simple.wikipedia.org`

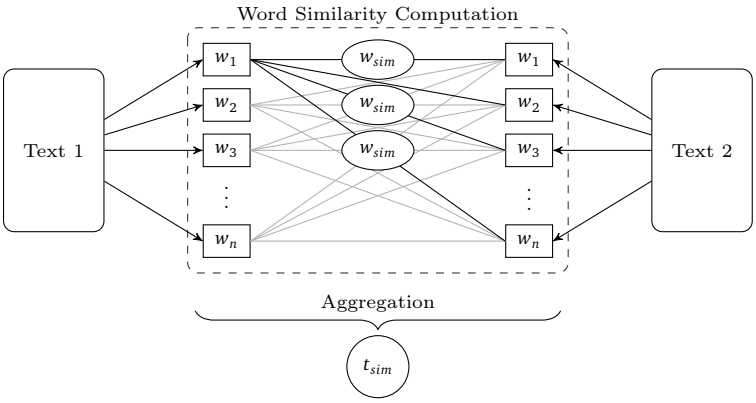

Figure 1: High-level depiction of the steps that are performed by compositional measures to compute text similarity between two given input texts: After tokenization, pairwise word similarity scores w_{sim} are computed between all words (i.e. similarity computation is limited to one word pair at a time), before then aggregating all pairwise scores to an overall score t_{sim} for the texts.

style. We then reported on the annotation studies which we conducted in order to empirically ground the proposed text dimensions. The results of these studies demonstrate that humans distinguish between the proposed dimensions when asked to rate the degree of text similarity. Also, they seem intuitively able to find an appropriate dimension of comparison for a given text collection. Smith and Heise (1992) refer to that as perceived similarity which "changes with changes in selective attention to specific perceptual properties." Selective attention can probably be modeled using dimension-specific similarity measures.

4 Text Similarity Measures

A large number of text similarity measures have been proposed in the literature. Making use of the text dimensions defined in the previous section, we propose to categorize text similarity approaches mainly into measures based on content, structure, and style.

4.1 Content Similarity Measures

We further sub-divide the quite large class of content similarity measures into compositional and non-compositional measures.

4.1.1 Compositional Measures

Compositional measures tokenize the input texts, compute pairwise word similarity between all words, and aggregate the resulting scores to an overall similarity score. Figure 1 shows a high-level depiction of the steps that are performed by compositional measures to compute text similarity between two given input texts. After tokenization, pairwise word similarity scores are computed between all words (i.e. similarity computation is limited to one word pair at a time), before then aggregating all pairwise scores to an overall score for the texts. As discussed above, the literature proposes different sets of instantiations for these steps. In Table 2, we summarize the proposals and list the underlying word similarity measures along with a very brief summary of the employed aggregation strategy. Popular word similarity measures have previously been surveyed by Zesch and Gurevych (2010) and comprise measures such as Wu and Palmer (1994), Hirst and St. Onge (1998), Leacock and Chodorow (1998), Jiang and Conrath (1997), or Lesk (1986). Basic strategies to aggregate the pairwise word similarity scores for a text pair are, for example, to compute the arithmetic mean of all scores or to take the maximum score, even though more sophisticated strategies exist.

4.1.2 Non-Compositional Measures

Non-compositional measures, on the other hand, first project the complete input texts onto a certain model, e.g. a high-dimensional vector space, before then comparing them based on the abstract representations. Figure 2 shows a high-level depiction of the steps that are performed by non-compositional measures to compute text similarity between two given input texts. After pre-processing, the texts are projected onto a certain model such as a high-dimensional vector space or a graph structure, before text similarity is then computed based on the abstract representations. For compositional measures which we discussed in the previous section, the two major steps (word similarity computation, aggregation) were modular and fully interchangeable. Any compositional measure thus is an instantiation of measures

Reference	Word Similarity Measures	Aggregation
Mihalcea et al. (2006)	PMI-IR (Turney, 2001), Latent Semantic Analysis (Landauer et al., 1998), Leacock and Chodorow (1998), Lesk (1986), Wu and Palmer (1994), Resnik (1995), Lin (1998a), Jiang and Conrath (1997), additional measures by Mohler and Mihalcea (2009): Shortest Path (Rada et al., 1989), Hirst and St. Onge (1998), Explicit Semantic Analysis Gabrilovich and Markovitch (2007)	Bidirectional aggreg.
Li et al. (2006)	Shortest Path (Rada et al., 1989), word order	Cosine
Islam and Inkpen (2008)	Longest Common Subsequence, SOC-PMI (Islam and Inkpen, 2006)	Unidirectional aggreg.
Ho et al. (2010)	Longest Common Subsequence, Yang and Powers (2005)	As above
Islam et al. (2012)	Word trigrams (Web1T)	As above, but modified maxSim function
Tsatsaronis et al. (2010)	Shortest Path (Rada et al., 1989), harmonic mean of tf-idf scores	Similar to Mihalcea et al. (2006)

Table 2: Compositional measures as sets of instantiations of the steps depicted in Figure 1. The measures differ in the proposal of underlying word similarity measures and aggregation strategy—which are both fully interchangeable.

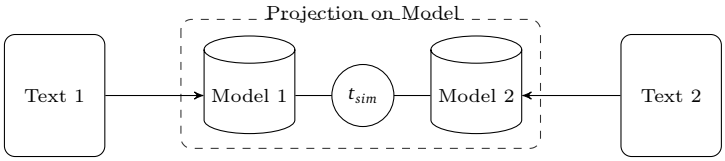

Figure 2: High-level depiction of the steps that are performed by non-compositional measures to compute text similarity between two given input texts: After pre-processing, the texts are projected onto a certain model such as a high-dimensional vector space or a graph structure, before the overall text similarity score t_{sim} is then computed based on the abstract representations.

and strategies for these steps. In order to form a new compositional measure, any combination of a new or existing word similarity measure and aggregation strategy may be chosen. In contrast, the two major steps for non-compositional measures (projection onto model, similarity computation) are an integral part of the text similarity measure and typically not interchangeable. A non-compositional measure is thus more than just a combination of a particular projection and similarity computation step. Due to the fact that the abstract models are very different between the proposed measures, a similarity computation step specifically tailored towards the employed model is required. In Table 3, we summarize major non-compositional measures.

From an algorithmic point of view, compositional measures are typically more expensive than non-compositional ones as they compute pairwise word similarity between all words in two texts. Depending on the nature of the underlying word similarity measure, this may require considerable processing time. Non-compositional measures, however, cannot be generally said to be superior in runtime complexity. While some of the measures such as as the string distance metrics require little to none effort to project the input texts onto the model, other measures such as Latent Semantic Analysis (Landauer et al., 1998) and Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) need to build up very large models first, before then being able to project texts onto them. Usually, however, the creation of the models is a one-time effort which casts a large document collection such as Wikipedia onto its model representation. The comparison of two texts can then be done rather efficiently.

Other Classification Schemes

In this article, we focus on compositional vs. non-compositional measures, but in the literature other classification schemes have been proposed. For example, measures can be classified by the type of resources used into corpus-based and knowledge-based measures (Mihalcea et al., 2006). Corpus-based measures operate on corpus statistics gathered from a usually large representative corpus. For example, the raw frequencies of all words may be computed along with statistics of which words appear in what documents in order to identify function words, a composition which is widely known as tf-idf. Knowledge-based measures operate on lexical-semantic resources that encode human knowledge about words. Such resources are, for example, dictionaries, thesauri, or wordnets. They encode knowledge about words and their definitions (dictionaries, wordnets), or the relations between them (thesauri, wordnets) in a machine-readable format. The most prominent example of a lexical-semantic resource probably is WordNet (Fellbaum, 1998). Ho et al.

Text Similarity Measure	Description
Levenshtein distance (Levenshtein, 1966)	Edit-distance metric (uniform)
Longest Common Subsequence (Allison and Dix, 1986)	Longest non-contiguous character sequence
Jaro distance (Jaro, 1989)	Short string matching (e.g. person or place names)
Jaro-Winkler distance (Winkler, 1990)	Short string and prefix matching
Greedy String Tiling (Wise, 1996)	Shared substrings of maximal length
Longest Common Substring (Gusfield, 1997)	Longest contiguous character sequence
Monge Elkan distance (Monge and Elkan, 1997)	Edit-distance metric (affice gap model)
Latent Semantic Analysis (Landauer et al., 1998)	Semantic space on corpus statistics
Word n-grams (Lyon et al., 2001)	Word n-gram set comparisons
Character n-gram profiles (Keselj et al., 2003)	Character n-gram set comparisons
Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007)	Vector space model on Wikipedia
Kennedy and Szpakowicz (2008)	Projection onto Roget's Thesaurus or WordNet
Vector Space Model (Salton and McGill, 1983)	Generalized vector space model on word weights
Random Walks (Ramage et al., 2009)	Modified PageRank on WordNet
WikiWalk (Yeh et al., 2009)	Modified PageRank on Wikipedia

Table 3: Overview of existing non-compositional text similarity measures

(2010) and Tsatsaronis et al. (2010) extend the above classification scheme by proposing a third class hybrid measures which employs resources of both types.

Islam and Inkpen (2008) propose an alternative classification scheme with four classes: vector-based document model measures, corpus-based measures, hybrid measures, and descriptive feature-based measures. The first class describes measures which use any type of vector representation to compare two texts. Corpus-based measures and hybrid measures correspond to the classes by Ho et al. (2010) and Tsatsaronis et al. (2010) described above. Descriptive feature-based measures refer to measures which use a machine learning classifier with any set of features derived from the given texts.

4.2 Structural Similarity

As discussed in Section 3, we assume that text similarity can be computed along multiple dimensions inherent to texts. We should thus also consider measures which compute similarity based on structural aspects inherent to the compared texts.

Stamatatos (2011) introduced Stopword n-grams that are based on the idea that similar texts may preserve syntactic similarity while exchanging content words. Thus, the measure removes all content words while preserving only stopwords. All n-grams of both texts are then compared using the containment measure (Broder, 1997). A possible extension are part-of-speech n-grams, where one disregards the actual words that appear in two given texts, while taking only the words' part-of-speech tags into account. Such part-of-speech sequences are indicators for the texts' shallow syntactic structures.

Hatzivassiloglou et al. (1999) employed two similarity measures between pairs of words in order to compare the texts' syntactic structures. They first construct the set of word pairs which appear in both texts. They then create one feature vector per text, where each vector element corresponds to the weight of a shared word pair. The weights are determined as follows: The word pair order measure assumes that a similar syntactic structure causes two words to occur in the same order in both texts (with any number of words in between). The word pairs hence receive binary weights whether they occur in the same order or not. The complementary word pair distance measure counts the number of words which lie between those of a shared word pair. The word pairs hence receive numeric weights which correspond to the number of words in between. Finally, the feature vectors are compared using Pearson correlation.

4.3 Stylistic Similarity

Measures of stylistic similarity adopt ideas from authorship attribution (Mosteller and Wallace, 1964) or use statistical properties of texts to compute text similarity. The type-token ratio (TTR) (Templin, 1957), for example, compares the vocabulary richness of two texts. However, it suffers from sensitivity to variations in text length and the assumption of textual homogeneity (McCarthy and Jarvis, 2010): As a text gets longer, the increase of the number of tokens is linear, while the increase of the number of types steadily slows down. In consequence, lexical repetition causes the TTR value to vary, while it does not necessarily entail that a reader perceives changes in the vocabulary usage. Secondly, textual homogeneity is the assumption of the existence of a single lexical diversity level across a whole text, which may be violated by different rhetorical strategies. Sequential TTR (McCarthy and Jarvis, 2010) alleviates these shortcomings. It iteratively computes a TTR score for a dynamically growing text segment until a point of saturation – i.e. a fixed TTR score of 0.72 – is reached, then it starts anew from that position in the text for a new segment. The final lexical diversity score is computed as the number of tokens divided by the number of segments.

Inspired by Yule (1939) who discussed sentence length as a characteristic of style, we define two simple measures, average sentence length and average token length. These measures compute the average number of tokens per sentence and the average number of characters per token, respectively. Additionally, we compared the average sentence and token lengths between the texts of a pair. We refer to these measures as sentence length ratio and token length ratio, respectively.

Finally, texts can also be compared by their function word frequencies (Dinu and Popescu, 2009) which have shown to be good style indicators in authorship attribution studies. They use a set of 70 function words identified by Mosteller and Wallace (1964) and compute feature vectors of their frequencies for each text pair. The vector similarity is determined using Pearson correlation.

4.4 Section Summary

In this section, we discussed a variety of text similarity measures that have been proposed in the literature and which we classify into compositional and non-compositional measures.

An important observation is that there exist virtually any number of compositional measures as they can be formed from arbitrary combinations of new or existing word similarity measures and a suitable aggregation strategy (see Table 2). Even though some combinations have been explicitly proposed in the literature, as discussed above, others may achieve good results as well. It is still unclear which combinations work well for what kind of data, and the literature on compositional measures typically gives no clue as to why particular measures and strategies have been preferred over others. In our opinion, all measures have their inherent strengths and weaknesses. We believe that all of them judge text similarity along particular text characteristics, and—as we will see later in this article—instead of creating more and more separate measures, it seems a promising research direction to harness the potential of the existing measures and combine them in a single model, in order to capture a wide variety of text characteristics.

5 Evaluation Datasets and Methodology

In this section, we discuss the methodology for evaluating text similarity measures. The performance of text similarity measures can be evaluated either in an intrinsic or extrinsic evaluation. In an intrinsic evaluation, the performance of text similarity measures is evaluated in an isolated setting. In an extrinsic evaluation, the performance of text similarity measures is evaluated with respect to a particular task at hand, where text similarity is a means for solving a concrete problem. In the following, we discuss both types of evaluation, and present datasets and evaluation metrics which have been widely used in the literature.

5.1 Intrinsic Evaluation

An intrinsic evaluation assesses the performance of text similarity measures in an isolated setting. Therefore, the datasets for an intrinsic evaluation contain text pairs along with human similarity judgments. The intuition is that machines should be able to judge text similarity for the given pairs in a similar manner as humans do. Systems are thereby expected to output continuous text similarity scores within a given interval, e.g. between 0 and 5 where 0 means not similar at all and 5 is perfectly similar. Evaluation is then carried out by comparing the system results with the human judgments. In the following, we introduce the datasets and the evaluation metrics which have been widely used in the literature. In Section 6, we present details and results on an intrinsic evaluation which we carried out in the context of the Semantic Textual Similarity (STS) Task (Agirre et al., 2012) at the Semantic Evaluation (SemEval) workshop.

5.1.1 Datasets

Popular datasets for an intrinsic evaluation of text similarity measures are the 30 Sentence Pairs dataset (Li et al., 2006), the 50 Short Texts collection (Lee et al., 2005), the Microsoft Paraphrase Corpus (Dolan et al., 2004), and the Microsoft Video Description Paraphrase Corpus (Chen and Dolan, 2011). Each dataset contains text pairs which are accompanied by human judgments about their perceived similarity.[3] In Table 4, we summarize the statistics for these datasets.[4]

[3] A subset of the Microsoft Paraphrase Corpus and the Microsoft Video Description Paraphrase Corpus have been re-annotated with human judgments in the context of the pilot Semantic Textual Similarity (STS) Task (Agirre et al., 2012) at the Semantic Evaluation (SemEval) workshop.

[4] Table 4 contains statistics for both intrinsic and extrinsic evaluation datasets. We discuss the latter in Section 5.2.1.

[5] We report the class distributions along with the dataset descriptions in Section 5.2.1.

Dataset Text Type / Domain	Length in Terms (⌀)	# Pairs	Rating Scale	# Judges per Pair
30 Sentence Pairs (Li et al., 2006) Concept Definitions	5–33 (14)	30	0–4	32
50 Short Texts (Lee et al., 2005) News (Politics)	45–126 (80)	1,225	1–5	8–12
MS Paraphrase Corpus (Dolan et al., 2004) News	5–31 (19)	5,801	binary[3]	2–3
MS Video. Paraphrase Corpus (Chen and Dolan, 2011) Video descriptions	1–50 (7)	120K	binary[3]	n/a
Wikipedia Rewrite Corpus (Clough and Stevenson, 2011) Computer Science	36–343 (208)	95	4-way[5]	1
METER Corpus (Gaizauskas et al., 2001) News (Law & Court, Show Business)	17–1K (205)	1,716	3-way[5]	1
Webis Crowd Paraphr. Corpus (Burrows et al., 2013) Book excerpts	28–954 (618)	7,859	binary[5]	1

Table 4: Statistics for seven datasets which have been used for the evaluation of text similarity measures. We classify them as datasets for an intrinsic (top) and extrinsic evaluation (bottom).

30 Sentence Pairs

Li et al. (2006) introduced a collection of 65 sentence pairs. The dataset is an extension of the collection of noun pairs by Rubenstein and Goodenough (1965), where each noun was replaced by its definition from Collins Cobuild English Dictionary (Sinclair, 2001). For example, the original noun pair gem/jewel was replaced by the sentence pair "A gem is a jewel or stone that is used in jewellery. / A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces." The dataset is accompanied by judgments from 32 subjects on how similar in meaning one sentence is to another. As the distribution of similarity scores was heavily skewed towards low scores, Li et al. (2006) selected 30 pairs to reduce the bias in the frequency distribution. We refer to this subset by the name 30 Sentence Pairs, henceforth. In the literature, this dataset has been used for evaluation, for example, by Islam and Inkpen (2008), Kennedy and Szpakowicz (2008), and Tsatsaronis et al. (2010).

50 Short Texts

The dataset by Lee et al. (2005) has been used for evaluation, for example, by Gabrilovich and Markovitch (2007) and Yeh et al. (2009). It comprises 50 short texts (45 to 126 words in length)[6] which contain newspaper articles from the political domain. For the annotation by human judges, each text was paired with every other one, resulting in 1,225 distinct text pairs. Human judgments were collected from 83 subjects who were paid on a per-100-ratings basis. The subjects were asked to rate "how similar they felt the documents were" on a discrete 1–5 scale (higher values indicating higher similarity). In the end, each pair received between 8 and 12 human scores, which were averaged to create the final scores.

An example text pair is shown in Figure 3. Here, the first text talks about a Chinese car registration system, while the second one elaborates on the topic of Chinese workers seeking employment in Russia. The average human similarity rating for this pair is in the medium range (2.80) on the 1–5 scale, which reflects the intuition that both texts share particular topics (e.g. China), but differ in various aspects (e.g. car registration system vs. work & employment).

Microsoft Paraphrase Corpus

Dolan et al. (2004) introduced a dataset of 5,801 sentence pairs (5 to 31 words in length) taken from news sources on the Web. The objective of this corpus is different from the corpora presented above: It originates in the field of paraphrase recognition, and hence is originally not accompanied by continuous human similarity scores. Rather, the text pairs are binary classified as paraphrase or no paraphrase. Despite the simpler annotation scheme, the text similarity community has widely adopted this dataset for evaluating text similarity measures (Mihalcea et al., 2006; Islam and Inkpen, 2008; Ramage et al., 2009; Tsatsaronis et al., 2010). Originally, binary judgments were collected from 2–3 subjects who indicated whether a pair captures a paraphrase relationship or not (83% inter-annotator agreement)[7]. In the end, 3,900 (67%) of all pairs were positive examples. According to the instructions which were given to the annotators, a paraphrase relationship holds if two sentences are "more or less semantically equivalent". The loose definition is explained in more detail by several positive and negative examples in the annotation guidelines, two of them are shown in Figure 4. For the positive example, the same content is expressed via similar lexical items

[6] Lee et al. (2005) report the shortest document having 51 words probably due to a different tokenization strategy.

[7] The reported agreement is the fraction of text pairs that the annotators agreed on (no chance correction).

(a) Beijing has abruptly withdrawn a new car registration system after drivers demonstrated "an unhealthy fixation" with symbols of Western military and industrial strength - such as FBI and 007. Senior officials have been infuriated by a popular demonstration of interest in American institutions such as the FBI. Particularly galling was one man's choice of TMD, which stands for Theatre Missile Defence, a US-designed missile system that is regularly vilified by Chinese propaganda channels.

(b) The Russian defense minister said residents shouldn't feel threatened by the growing number of Chinese workers seeking employment in the country's sparsely populated Far Eastern and Siberian regions. There are no exact figures for the number of Chinese working in Russia, but estimates range from 200,000 to as many as 5 million. Most are in the Russian Far East, where they arrive with legitimate work visas to do seasonal work on Russia's low-tech, labor-intensive farms.

Figure 3: Example sentence pair taken from the 50 Short Texts dataset by Lee et al. (2005). The average human similarity score of this pair is 2.80 (standard deviation 0.98) on a 1–5 scale.

Positive example: equivalent content

(a) The Senate Select Committee on Intelligence is preparing a blistering report on prewar intelligence on Iraq.

(b) American intelligence leading up to the war on Iraq will be criticised by a powerful US Congressional committee due to report soon, officials said today.

Negative example: shared content of the same event, but lacking details

(a) Researchers have identified a genetic pilot light for puberty in both mice and humans.

(b) The discovery of a gene that appears to be a key regulator of puberty in humans and mice could lead to new infertility treatments and contraceptives.

Figure 4: Positive and negative paraphrase examples taken from the annotation guidelines of the Microsoft Research Paraphrase Corpus (Dolan et al., 2004). Semantically overlapping and discriminating phrases are underlined.

(underlined). The negative example, on the other hand, does not capture a paraphrase relationship due to the lack of details (underlined), even though both sentences share a basic description of the same event.

Due to the nature of the original dataset described above, only an extrinsic evaluation could only be carried out, where the goal would be to identify whether a paraphrase relationship holds or not. However, in the context of the Semantic Textual Similarity (STS) Task (Agirre et al., 2012) at the Semantic Evaluation (SemEval) workshop, a subset of 1,500 text pairs from this dataset were re-annotated with human similarity judgments on a continuous $0-5$ scale. That way, the text pairs of the original dataset have been reused, but instead of the binary paraphrase annotations, the new dataset is accompanied by continuous human judgments on text similarity. Hence, the revised dataset is well suited for an intrinsic evaluation and may be highly beneficial for future work on text similarity. In Section 6, we conduct our intrinsic evaluation on this dataset in addition to four other datasets. The revised dataset is freely available.[8]

Microsoft Video Description Paraphrase Corpus

Chen and Dolan (2011) originally introduced this dataset in the field of paraphrase recognition to address the lack of a large-scale evaluation dataset. In their study, they presented short (i.e. typically less than 10 seconds) video clips to human subjects and asked each one to provide a single sentence description of the video's "main action or event". They carried out the study in a crowdsourcing setting using Amazon Mechanical Turk. In total, the collection contains about 120,000 multilingual parallel descriptions (approximately 85,000 in English) of more than 2,000 video clips. The parallel descriptions of a single video clip can then be considered paraphrases of each other. The authors argue that

[8] http://www.cs.york.ac.uk/semeval-2012/task6/index.php?id=data

gathering paraphrases in this particular way overcomes the limitations of traditional paraphrase acquisition: The subjects are not biased by any lexical or stylistic choices of an original sentence which is then to be paraphrased.

As with the aforementioned dataset, this dataset is originally suited only for an extrinsic evaluation where the goal would be to identify whether a paraphrase relationship holds or not. However, this dataset has also been adopted for the evaluation of text similarity measures in an intrinsic evaluation in the context of the Semantic Textual Similarity Task (Agirre et al., 2012) at SemEval-2012. Human subjects annotated the similarity of a subset of 1,500 text pairs from this dataset on a continuous $0-5$ scale. We report details on our intrinsic evaluation on this dataset in Section 6. The annotated subset is again freely available.[8]

5.1.2 Metrics

In an intrinsic evaluation, the system output is usually compared with human judgments by computing Pearson or Spearman correlation, which we discuss in the following.

Pearson's r

The correlation coefficient is typically denoted by r and measures the strength of linear dependence between two similarity score vectors, i.e. how well the system reflects human judgments. The value of r is in the interval $[-1;1]$ where $r=0$ can be interpreted as not similar at all, and $r=-1$ and $r=1$ are perfect linear relationships (completely similar). The relationship between $\vec{x}$ and $\vec{y}$ can be visualized in a scatter plot: For perfect linear relationships, all data points are on a regression line that has a positive ($r=1$) or negative ($r=-1$) slope. Pearson's r between two score vectors $\vec{x}$ and $\vec{y}$ is computed as follows, with $n=|\vec{x}|=|\vec{y}|$:

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

Linear transformations of the similarity scores do not affect the correlation. For example, a perfect correlation $r=1$ can be observed between $\vec{x}=[1,2,3,4]$ and $\vec{y}=[2,4,6,8]$. Pearson's r, however, cannot cope with any non-linear relationships such as $\vec{x}=[1,2,3,4]$ and $\vec{y}=[1,4,9,16]$ and is highly sensitive to outliers.

Spearman's ρ

Typically denoted by ρ, Spearman's rank correlation is not computed between the absolute values of the elements in two similarity score vectors, but between their ranks. In case of the absence of tied ranks, Spearmans's ρ can be calculated using a simplified procedure: The raw similarity scores are transformed into ranks, then the difference d_i between each of the ranks x_i and y_i is computed. Spearman's ρ is then computed as follows, with $n=|\vec{x}|=|\vec{y}|$:

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

If tied ranks are present, Spearman's ρ is computed as Pearson's r (see Equation 1) between the ranked similarity scores. For tied ranks, the arithmetic mean between all of their individual ranks is used.

Spearman's ρ overcomes the shortcomings of Pearson's r such as sensitivity to outliers and its limitation to measuring only linear relationships between the similarity score vectors. However, Spearman's ρ suffers from the drawback that a perfect correlation of all ranks does not entail a perfect prediction of continuous similarity scores: For example, if a measure were to predict $\vec{x}=[0.1,0.2,4.9,5.0]$, i.e. two very low and two very high similarity scores on a $0-5$ scale, a perfect Spearman's ρ would be computed with the gold standard $\vec{y}=[4.7,4.8,4.9,5.0]$, i.e. very high similarity scores for all four elements. Here, a perfect rank correlation $\rho=1$ would be observed as only the pairwise ranks are compared: $rank(\vec{x})=rank(\vec{y})=[4,3,2,1]$.

5.2 Extrinsic Evaluation

An extrinsic evaluation measures the performance of text similarity measures with respect to a particular task at hand, where text similarity is a means for solving a specific problem. In this article, and particularly in Section 7, we focus on the task of text reuse detection. Text reuse is thereby defined as "the reuse of existing written sources in the creation of new text" (Clough et al., 2002). A system for computing text similarity is expected to produce a classification output which assigns each text pair a class label such as similar/dissimilar for a binary classification, or highly similar/moderately similar/dissimilar for a multiclass setting. Evaluation is then carried out by comparing the system output with the human classifications. In the following, we introduce the datasets and the evaluation metrics which are suitable for an extrinsic evaluation. In Section 7, we present details and results of an extrinsic evaluation which we carried out for the task of text reuse detection.

5.2.1 Datasets

Popular datasets for this task include the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011), the METER Corpus (Gaizauskas et al., 2001), and the Webis Crowd Paraphrase Corpus (Burrows et al., 2013). The datasets contain text pairs along with human judgments on the degree of text reuse.

Wikipedia Rewrite Corpus

The dataset introduced by Clough and Stevenson (2011) contains 95 pairs of short texts (193 words on average). For each of 5 questions about topics of computer science (e.g. "What is dynamic programming?"), a reference answer (source text, henceforth) has been manually created by copying portions of text from a suitable Wikipedia article. Text similarity now occurs between a source text and the answers given by each of 19 human subjects. The subjects were asked to provide short answers, each of which should comply to one of four rewrite levels and hence reuse the source text to a varying extent. According to the degree of rewrite, the dataset is 4-way classified as cut & paste (38 texts; simple copy of text portions from the Wikipedia article), light revision (19; synonym substitutions and changes of grammatical structure allowed), heavy revision (19; rephrasing of Wikipedia excerpts using different words and structure), and no plagiarism (19; answer written independently from the Wikipedia article). We will further discuss this dataset in Section 7 and give an example of a heavy revision in Figure 5 on page 19.

METER Corpus

The dataset was created by Gaizauskas et al. (2001) and contains news sources from the UK Press Association (PA) and newspaper articles from 9 British newspapers that reused the PA source texts to generate their own texts. The complete dataset contains 1,716 texts from two domains: law & court and show business. All newspaper articles have been annotated whether they are wholly derived from the PA sources (i.e. the PA text has been used exclusively as text reuse source), partially derived (the PA text has been used in addition to other sources), or non-derived (the PA text has not been used at all).

Several newspaper texts, though, have more than a single PA source in the original dataset where it is unclear which (if not all) of the source stories have been used to generate the rewritten story. However, for text similarity computation it is important to have aligned pairs of reused texts and source texts. Therefore, Sánchez-Vega et al. (2010) proposed to select only a subset of text pairs where only a single source story is present in the dataset. This leaves 253 pairs of short texts (205 words on average) out of which 181 (72%) are classified as positive samples where text reuse occurs, and 72 (28%) as negative samples.

Webis Crowd Paraphrase Corpus

The dataset by Burrows et al. (2013) was originally introduced as part of the PAN 2010 international plagiarism detection competition (Potthast et al., 2010). It contains 7,859 pairs of original texts along with their paraphrases (28 to 954 words in length) with 4,067 (52%) positive and 3,792 (48%) negative samples. The original texts are book excerpts from free e-books of Project Gutenberg[9], and the corresponding paraphrases were acquired in a crowdsourcing process using Amazon Mechanical Turk (Callison-Burch and Dredze, 2010). In the manual filtering process[10] of all acquired paraphrases, Burrows et al. (2013) follow the paraphrase definition by Boonthum (2004): a good paraphrase exhibits patterns such as synonym use, changes between active and passive voice, or changing word forms and parts of speech, and a bad paraphrase is rather e.g. a (near-)duplicate or an automated one-for-one word substitution. This definition implies that a more sophisticated interpretation of text similarity scores needs to be learned, where e.g. (near-)duplicates with very high similarity scores are in fact negative samples.

5.2.2 Metrics

In an extrinsic evaluation, the system output is compared with human classifications. Besides accuracy, i.e. the number of correctly predicted text pairs divided by the total number of pairs, a popular evaluation metric is F_1 score, which we discuss in the following.

F_1 score

In a classification context, F_1 score evaluates system performance in terms of precision and recall. Contrary to Pearson's r and Spearman's ρ, F_1 score does not directly compare a system output of continuous similarity scores with human similarity judgments. It rather expects a system to produce a classification output which assigns each text pair a class label such as similar/dissimilar for a binary classification, or highly similar/moderately similar/dissimilar for a multiclass setting. These classifications are then compared with human judgments, and can be categorized

[9] http://www.gutenberg.org

[10] Burrows et al. (2013) do not report any inter-annotator agreement for the filtering process, as the task was split across two annotators and each text pair was labeled by only a single annotator.

according to the predictive value as true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). Precision and recall are defined as:

$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn} \tag{3}$$

$F_1 score$ is defined as the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

6 Intrinsic Evaluation

As we have shown in the previous section, humans indeed perceive text similarity along different text dimensions. We therefore proposed to focus on the three dimensions content, structure, and style. Depending on the concrete task at hand, we argue that a sophisticated system for the detection of text similarity may need to address more than a single text dimension when computing similarity, e.g. a combination of all three dimensions. We thus propose to combine a multitude of text similarity measures in a unified classification framework, where appropriate similarity measures are used for each of the text dimensions (Bär et al., 2012).

For an intrinsic evaluation, we apply our system to the pilot Semantic Textual Similarity (STS) Task at the Semantic Evaluation (SemEval-2012) exercises (Agirre et al., 2012). In the following, we first introduce the task in detail and report on the five evaluation datasets. We then give a detailed system description and an overview of our system along with its configuration parameters, and discuss the results obtained.

6.1 Task Description

Since 1998, SemEval is an ongoing series of semantic evaluation exercises, where systems compete on a number of shared tasks. The goal is to find computational means for assessing different aspects of meaning in language. While the exercise series primarily addressed word sense disambiguation (Yarowsky, 1995) in its early days (and was called Senseval back then)[11], it has evolved into a major venue for researchers working on a wide variety of topics such as text simplification (Chandrasekar et al., 1996), semantic role labeling (Palmer et al., 2010), or textual entailment (Dagan et al., 2006).

In 2012, a new task was introduced: The Semantic Textual Similarity (STS) Task (Agirre et al., 2012) is intended to unite multiple efforts across the applied semantics community. The goal is to design algorithms which measure the degree of semantic similarity between two sentences in a pair. These scores—a continuum from 0 (not similar at all) to 5 (completely similar)—are then compared to averaged human judgments. That way, it is evaluated how well the algorithmically produced scores resemble human judgments.

6.2 Employed Text Similarity Measures

In this section, we report on the text similarity measures which we used to compute similarity along the three characteristic text dimensions content, structure, and style.

6.2.1 Content Similarity

In Table 5, we list the compositional and non-compositional measures we used in the intrinsic evaluation and describe the configuration parameters as necessary.

For our experiments, we applied a novel measure based on a distributional thesaurus. The idea here is that similarity can possibly be effectively computed based on word co-occurrence statistics in a large corpus. Similar to Lin (1998b), we therefore implemented a word similarity measure based on pairwise similarity scores between all words from 10 million dependency-parsed sentences of English newswire. We then implemented a text similarity measure based on these pre-computed scores which follows the aggregation strategy by Mihalcea et al. (2006).

We used another novel approach, where we apply text expansion mechanisms which augment or replace (parts of) the original texts with synonyms, hence allowing for better comparisons in any consecutive step. We used two expansion mechanisms: lexical substitution and statistical machine translation, which we describe in the following. We then compared the modified texts using the strategy by Mihalcea et al. (2006) with the underlying measure by Resnik (1995) as described in Section 4.1.1.

[11] http://www.senseval.org

Text Similarity Measure	Configurations
Compositional Measures	
Mihalcea et al. (2006)	Resnik (1995) on WordNet
Non-compositional Measures	
Character n-gram profiles	$n = 2, \ldots, 15$
Explicit Semantic Analysis	Wikipedia, Wiktionary
Greedy String Tiling	
Longest common subsequence	2 normalizations
Longest common substring	
Word n-grams	Jaccard/Containment, $n = 1, \ldots, 15$

Table 5: Compositional and non-compositional text similarity measures introduced in Section 4, along with their configurations, which we used in the intrinsic evaluation

For lexical substitution, we used the system based on supervised word sense disambiguation (Biemann, 2012a,b) which automatically provides substitutions for a set of 1,012 frequent English nouns with high precision. For each covered noun, we added all the substitutions to the text and computed pairwise word similarity for the texts as described above. This feature alleviates the lexical gap for the covered subset of words. For statistical machine translation, we used the Moses SMT system (Koehn et al., 2007) to translate the original English texts via three bridge languages (Dutch, German, Spanish) back to English. In the translation process, additional lexemes are introduced which alleviate potential lexical gaps. The full augmented texts then comprise the concatenation of the original texts and all three back-translations. The system was trained on the Europarl corpus (Koehn, 2005), using the following configuration which was not optimized for this task: WMT11 baseline without tuning, with MGIZA alignment.

6.3 Experimental Setup

Our system builds upon the theoretical insights of Section 3: We combine a multitude of text similarity measures in a unified classification framework, where the measures are drawn from all three dimensions content, structure, and style. Our system is based on DKPro Core[12], a collection of software components for natural language processing built upon the Apache UIMA framework (Ferrucci and Lally, 2004). In Section 8, we will further detail the technical aspects of the framework.

The system works as follows: We first run each of the similarity measures introduced above separately on all text pairs. We then use the resulting similarity scores as features for a machine learning classifier in order to combine the measures. That way, we follow our motivation that the similarity computation process needs to address multiple text dimensions. In detail, we proceed as follows.

Pre-processing We tokenize the input texts and lemmatize using the TreeTagger (Schmid, 1994). Where applicable, we additionally apply a stopword filter.

Feature Generation We compute text similarity scores for the text pairs with all measures and for all configurations introduced above. This resulted in a vector of 300+ individual text similarity scores for each text pair.

Feature Combination We now use the pre-computed similarity scores and combine the log-transformed values using a linear regression classifier from the Weka toolkit (Hall et al., 2009). The classifier then predicts a numerical similarity score for all text pairs based on the given features.

Post-processing As the classifier determines a linear combination of the numerical features, the predicted text similarity scores may fall outside the required interval $[0, 5]$. We thus convert all scores outside this interval to the minimum/maximum value, respectively. For the configuration of Run 2—which we will introduce in the following—we further apply a post-processing filter which strips all characters off the texts which are not in the character range [a-zA-Z0-9]. If the texts are then equal, we set their similarity score to the maximum (5.0) regardless of the classifier's output. That way, we try to reduce the number of classification errors for texts which resemble each other under the assumption of a simplified vocabulary.

During the development cycle, we evaluated the performance of our system using 10-fold cross-validation on the training data. For the final system, we trained the classifier on the available training data and generated one model per dataset which we then applied to the test data. We report the following experimental configurations:

[12] `http://code.google.com/p/dkpro-core-asl`

Text Dimension	Text Similarity Features	MSRpar	MSRvid	SMT-eur
	Best Feature Set, Run 1	.711	.868	.735
	Best Feature Set, Run 2	.724	.868	.742
Content	Pairwise Word Similarity	.564	.835	.527
	Character *n*-gram profiles	.658	.771	.554
	Explicit Semantic Analysis	.427	.781	.619
	Word *n*-grams	.474	.782	.619
	String Similarity	.593	.677	.744
	Distributional Thesaurus	.494	.481	.365
	Lexical Substitution	.228	.554	.483
	Statistical Machine Translation	.287	.652	.516
Structure	Part-of-speech *n*-grams	.193	.265	.557
	Stopword *n*-grams	.211	.118	.379
	Word Pair Order	.104	.077	.295
Style	Statistical Properties[13]	.168	.225	.325
	Function Word Frequencies	.179	.142	.189

Table 6: Best results (Pearson correlation) for individual classes of measures, grouped by text dimension, on the training datasets MSRpar, MSRvid, and SMT-eur, using 10-fold cross-validation

Run 1 In a manual feature selection process, we identified the features which achieved the best performance on the training data (see Table 7). For each of the known datasets MSRpar, MSRvid, and SMT-eur, we trained a separate classifier and applied it to the test data. For the two surprise datasets On-WN and SMT-news, we trained the classifier on a joint dataset of all known training datasets.

Run 2 In the second configuration of our system we studied the effects of two additional features: lexical substitution and statistical machine translation. We added the corresponding measures (see Section 6.2.1) to the feature set of Run 1.

6.4 Results & Discussion

We now report and discuss the results obtained in the feature selection using the available training data, as well as the final results obtained on the test data.

6.4.1 Feature Selection

In order to find an optimal feature combination, we evaluated our system on the three available training datasets MSRpar, MSRvid, and SMT-eur by computing Pearson correlation of the system output with human judgments. In Table 6, we report the results achieved on each dataset using 10-fold cross-validation. The best results are based on the feature set of Run 2, with Pearson's $r = .724$, $r = .868$, and $r = .742$ for the datasets MSRpar, MSRvid, and SMT-eur, respectively. Run 2 performs at least as good as Run 1 across all three training datasets, even though the differences are not statistically significant.[14] Run 2 further outperforms (statistically significant) the best performing classes of content similarity measures for two of the three datasets.

Individual classes of content similarity measures achieved good results. A different class performed best for each dataset, e.g. character *n*-gram profiles for the MSRpar dataset, pairwise word similarity for the MSRvid dataset, and even simple string similarity measures for the SMT-eur dataset. We attribute the differences to the different nature of the data, as the original texts which were taken from completely different sources with a varying degree of lexical overlap.

Text similarity measures related to structure and style achieved only poor results on the training data. We attribute this fact to the following properties of the data: (i) The text length is only a single sentence. Measures designed to capture sophisticated aspects of structural similarity probably do not work well on these short texts. (ii) The texts of all pairs display similar style. This is due to the nature of the data and the pairing process: Pairs were only formed

[13] For brevity, we group the following measures (see Section 4.3): type-token ratio, sequential TTR, average sentence length, average token length, sentence length ratio and token length ratio.

[14] 95% confidence intervals (Fisher Z-value transformation): $0.688 \leq r \leq 0.756$ (MSRpar), $0.849 \leq r \leq 0.884$ (MSRvid), $0.707 \leq r \leq 0.772$ (SMT-eur)

Run	Text Similarity Measure	Configurations
1	Character 2-, 3-, and 4-gram profiles	
	Distributional Thesaurus	Cardinal numbers
	Explicit Semantic Analysis	Wikipedia, Wiktionary
	Greedy String Tiling	
	Longest common subsequence	2 normalizations
	Longest common substring	
	Mihalcea et al. (2006)	Resnik (1995) on WordNet
	Word 1- and 2-grams	Containment, w/o stopwords
	Word 1-, 3-, and 4-grams	Jaccard
	Word 2- and 4-grams	Jaccard, w/o stopwords
2	All Features of Run 1	
	Lexical Substitution for Word Similarity	
	Statistical Machine Transl. for Word Similarity	

Table 7: Feature sets of the two final system configurations for the intrinsic evaluation

between texts taken from the same source (e.g. news texts, video descriptions, etc.). By pairing texts that way, we expect stylistic measures to fail.

6.4.2 Final Results

In the final evaluation on the test data, three evaluation metrics were used (Agirre et al., 2012):[15] The ALL metric computes Pearson correlation of the union of the system outputs across all five datasets with human judgments. That is, given the five similarity score vectors $\vec{s_i} = \{s_{i_1}, \ldots, s_{i_n}\}$ for $i \in \{1, \ldots, 5\}$, a concatenated vector $\vec{s} = \{\vec{s_1}, \ldots, \vec{s_5}\}$ is constructed which is then compared with the analogous concatenation of the human judgments.

The ALLnrm metric first normalizes the system output per dataset, then applies the ALL metric.[16] The normalization reduces the squared error $\sum_i (y_i - x'_i)^2$, whereby $x'_i = x_i\beta_1 + \beta_2$ with $\beta_{1,2}$ determined analytically and $\vec{x}$ and $\vec{y}$ being the system output and the human judgments, respectively.

Mean refers to the weighted mean across the Pearson correlations r_i of all five datasets $i = \{1, \ldots, 5\}$, where the weight corresponds to the number n_i of text pairs in each dataset: $\sum_i (r_i n_i) / \sum_i n_i$

In Table 8, we report the offical results achieved on the test data. In total, 35 teams submitted 88 systems in addition to the provided cosine similarity baseline. Almost all systems outperformed the provided baseline for the ALL and ALLnrm metrics. For the Mean metric, the baseline was ranked #70 with an average Pearson correlation $r = 0.435$ across the five test datasets. Our first configuration, Run 1, was ranked #4 across all three evaluation metrics. It was outperformed by our best system configuration (Run 2) which was ranked #1 for the evaluation metrics ALL ($r = .823$)[17] and Mean ($r = .677$), and #2 for ALLnrm ($r = .857$). The results of both configurations are within the 95% confidence interval for the ALL metric.[17]

Error Analysis

In the following, we investigate the erroneous predictions by the classifier. In particular, we analyze the differences between the configurations of Run 1 and 2 in order to assess the effects of the two additional features lexical substitution and statistical machine translation. We therefore show an example text pair in the following, which is taken from the SMT-news dataset:

(i) Putin's Russia has already lost 12 leading journalists to murder in the past...

(ii) Putin's Russia has already lost 12 prominent journalists, murdered in the last...

[15] At system submission time, the ALL metric was the single offical evaluation metric. The task organizers later introduced two additional methods: ALLnrm and Mean.

[16] The ALL metric penalizes systems which do not perform equally well across all five datasets. The ALLnrm metric, on the other hand, assigns higher overall scores to systems which perform well only on some of the datasets. Besides the open question that remains which goal is more desirable, the ALLnrm metric was criticized for reducing the variance of similarity scores in the normalization step, which disproportionally favors poor predictions when computing Pearson correlation in the following step.

[17] 95% confidence interval (Fisher Z-value transformation): $0.811 \leq r \leq 0.834$

$\#_1$	$\#_2$	$\#_3$	System	r_1	r_2	r_3	MSR-par	MSR-vid	SMT-eur	On-WN	SMT-news
1	2	1	UKP (Run 2)	.823	.857	.677	.683	.873	.528	.664	.493
2	3	5	TakeLab	.813	.856	.660	.698	.862	.361	.704	.468
3	1	2	TakeLab	.813	.863	.675	.734	.880	.477	.679	.398
4	4	4	UKP (Run 1)	.811	.855	.670	.682	.870	.511	.664	.467
5	6	13	UNT	.784	.844	.616	.535	.875	.420	.671	.403
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
87	85	70	Baseline	.311	.673	.435	.433	.299	.454	.586	.390

Table 8: Official results on the test data for the top 5 participating runs out of 89 for the datasets MSRpar, MSRvid, and SMT-eur, as well as for the surprise datasets On-WN and SMT-news. We report the ranks ($\#_1$: ALL, $\#_2$: ALLnrm, $\#_3$: Mean) and the corresponding Pearson correlation r according to the three offical evaluation metrics (see Sec. 6.4.2). The provided cosine similarity baseline is shown at the bottom of this table. The 95% confidence interval[17] comprises the top four systems.

The above example received the maximum similarity score (5.0) by human annotators. In Run 1, our system predicted a similarity score of 4.37, while in Run 2 a score of 4.80 was predicted. We can clearly see that the improved score is due to the additional features of Run 2, as we show in the following. The first part of the texts are transformed into the following (not necessarily grammatically correct) fragments using statistical machine translation (as described in Section 6.2.1, using Spanish as a bridge language):

(i) The Russia of Putin has already lost 12 outstanding journalists of murder...

(ii) The Russia of Putin has already lost 12 outstanding journalists, murdered...

In the translation process, the mismatching words leading and prominent were transformed into outstanding in both texts. That way, the lexical gap was closed and in consequence an improved similarity score (4.80) was computed, which is very close the the human score (5.0). In the following, we give a second interesting example which also highlights the positive effects of the additional features of Run 2:

(i) Western Europeans, who have been spared this legacy, should heed our warnings.

(ii) The western Europeans, who have forgotten this history, should heed our warnings.

The above example received an average human similarity score of 5.0. The predictions by Runs 1 and 2 are 3.67 and 3.95, respectively. For this text pair, there is a lexical gap between the words legacy and history, which carry similar meanings in the two texts. In analogy to the first example, we see that the prediction of Run 2 shows an improvement over Run 1. In this case, we attribute the improved performance to the employed lexical substitution system: As discussed in Section 6.2.1, this system provides substitutions for a set of 1,012 frequent English nouns. The substitutions for the words legacy and history are listed below:

(i) legacy: –

(ii) history: record, story, background, past, annals, account, antiquity, historical record

For this example, no substitutions were found for the word legacy. For history in the second text, eight substitutions (see above) were found. As described in Section 6.2.1, in our experiments we add all substitutions to the original texts. In a subsequent step, text similarity measures are thus much better able to compute a proper similarity score between the two augmented texts, as additional synonyms are given.

6.5 Section Summary

In this section, we presented an intrinsic evaluation of our system in the context of the pilot Semantic Textual Similarity (STS) Task at SemEval-2012. We showed that our system performed best across the three official evaluation metrics.

While we did not reach the highest scores on any of the single datasets (see Table 8), our system was most robust across different data.

In future work, it would be interesting to inspect the performance of a combined model on all features of all participating systems. We already carried out first experiments in this direction: We combined the feature sets of the two top-ranked systems, i.e. our Run 2 and the system by TakeLab (Sarić et al., 2012), in a single log-linear regression model. Our assumption holds true and the combined model performs best across all three evaluation metrics with $r_1 = .835$, $r_2 = .872$, and $r_3 = .707$ for the metrics ALL, ALLnrm and Mean, respectively. The combined feature set also achieves the best performance on the MSRvid dataset, $r = .899$. The results show that the text similarity computation process can greatly benefit from a rich set of features covering a wide variety of text characteristics. In consequence, we believe this to be a highly promising direction for future research.

The STS Task is a supervised setting, i.e. text pairs along with human judgments are provided at development time in order to train the systems. This setting allows our system to built a machine learning classifier which learns very well combinations of the text similarity scores with respect to human judgments on the training data, and then successfully applies this model to unseen test data. However, we argue that its generalization is still limited, due to two major issues: (a) It is unclear how to judge similarity between pairs of texts which contain contextual (e.g. temporal) references such as on Monday vs. after the Thanksgiving weekend. (b) For several pairs, it is unclear what point of view to take, e.g. for the pair An animal is eating / The animal is hopping. Is the pair to be considered similar (an animal is doing something) or rather not (eating vs. hopping)? In both cases, a different set of annotators may have a different view on how similar such texts actually are. Even more, a new collection of text pairs may favor particular points of view, such as resolving temporal references with respect to a time frame that is shared by many text pairs. For future work, we believe that a more clear-cut definition of what constitutes text similarity and what point of view is to be taken is key to systems that will generalize even better.

7 Extrinsic Evaluation

In this section, we conduct an extrinisic evaluation of our system in the task of text reuse detection. We first describe the task in Section 7.1, and then report on our experimental setup and the features used in Sections 7.2 and 7.3. We then discuss the evaluation results on three standard datasets and conclude with an analysis of our findings.

7.1 Task Description

Text reuse is "the reuse of existing written sources in the creation of a new text" (Clough et al., 2002). Text reuse is a common phenomenon and arises, for example, on the Web from mirroring texts on different sites or reusing texts in public blogs. In other text collections such as content authoring systems of communities or enterprises, text reuse arises from keeping multiple versions, copies containing customizations or reformulations, or the use of template texts (Broder et al., 1997). Detecting text reuse has been studied in a variety of tasks and applications, e.g. the detection of journalistic text reuse (Clough et al., 2002), the identification of rewrite sources for ancient literary texts (Lee, 2007), or the analysis of text reuse in blogs and web pages (Abdel-Hamid et al., 2009).

A common approach to text reuse detection is to compute similarity between a source text and a possibly reused text. A multitude of text similarity measures have been proposed for computing similarity based on surface-level and semantic features (see Section 4). However, existing similarity measures typically exhibit a major limitation: They compute similarity only on the features which can be derived from the content of the given texts. Thus, they so far ignore any other text characteristics, e.g. the structural and stylistic text dimensions.

Figure 5 shows an example of text reuse taken from the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011) (see Section 5.2.1 for details on the dataset) where parts of a given source text have been reused either verbatim or by using similar words or phrases. As the example illustrates, the process of creating reused text includes a revision step in which the author has a certain degree of freedom on how to reuse the source text. Similarity between the topics and their relations in both revisions can then be detected by content-centric text similarity measures. However, the author has further split the source text into two individual sentences and changed the order of the reused parts. For detecting the degree of similarity of such a revision, structural similarity should be computed. Additionally, the given texts exhibit a high degree of similarity with respect to stylistic features, e.g. vocabulary richness.[18] In order to use such features as indicators of text reuse, we propose to further include measures of stylistic similarity.

7.2 Employed Text Similarity Measures

For this task, we employ a similar set of text similarity measures as in Section 6.2, with a minor set of modifications: While we used the set of structural and stylistic similarity measures as originally described, we slightly modified the

[18] The type-token ratio (Templin, 1957) of the texts is 0.79 and 0.71, respectively.

Source Text. PageRank is a link analysis algorithm used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set.

Text Reuse. The PageRank algorithm is used to designate every aspect of a set of hyperlinked documents with a numerical weighting. It is used by the Google search engine to estimate the relative importance of a web page according to this weighting.

Figure 5: Example of text reuse taken from the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011). Various parts of the source text have been reused, either verbatim (underlined) or using similar words or phrases (wavy underlined). However, the editor has split the source text into two individual sentences and changed the order of the reused parts.

set of content similarity measures to better fit the task of text reuse detection. In the following, we list all the differing measures and those with modified configuration parameters.

In the original system, we used the Resnik (1995) measure which is capable of measuring semantic similarity between words, and which we then aggregated according to the strategy proposed by Mihalcea et al. (2006). For this task, we used two additional word similarity measures with WordNet (Fellbaum, 1998): Jiang and Conrath (1997) and Lin (1998b).

For assessing word n-gram similarity, we compared the sets of n-grams using the Jaccard coefficient, following Lyon et al. (2001), as well as using the containment measure (Broder, 1997). We again tested n-gram sizes for $n = 1, 2, \ldots, 15$, and will use the original system name Ferret (Lyon et al., 2004) to refer to the variant with $n = 3$ using the Jaccard coefficient, henceforth.

Furthermore, we used Latent Semantic Analysis (LSA) (Landauer et al., 1998) for comparing texts, where the construction of the semantic space was done using the evaluation corpora (see Section 7.4).

For this task, we did not use the following measures: no text expansion mechanisms (see Section 6.2.1) such as lexical substitution or statistical machine translation, and no similarities from a Distributional Thesaurus (Lin, 1998b).

7.3 System Description

The system we used for the extrinsic evaluation in the context of text reuse detection is a modification of the system originally introduced in Section 6.3 . It again follows the idea that text similarity can be best detected if a composition of measures is used. As for the original system, we refer the reader to Section 8 for an elaborate discussion on the technical aspects of the system. The system we used for the experiments throughout this section modifies the original setup as follows:

Pre-processing no modifications; see Section 6.3

Feature Generation no modifications; see Section 6.3

Feature Combination We slightly modified the feature combination step of the original system to meet the requirements of this task. We still use the pre-computed similarity scores as input for the machine learning classifier. While we used a linear regression classifier in Section 6 which predicts numerical scores for each text pair, we now combine the features (using their raw values instead of applying a log-transformation) with a Naive Bayes and a C4.5 decision tree classifier (J48 implementation) from the Weka toolkit (Hall et al., 2009). These classifiers are able to predict nominal classes of text reuse, e.g. a heavy revision shown in Figure 5.

Post-processing We did not apply any post-processing steps for this task.

Using 10-fold cross-validation, we then ran three sets of experiments as follows:

Run 1 In the first set of experiments, we tested the text similarity scores of one single measure at a time as a single feature for the classifier. That way, we were able to determine those text similarity measures which perform best per text dimension.

Run 2 We then tested a combination of similarity measures per text dimension, i.e. we combined multiple similarity measures of the content, structure, and style dimensions separately. That way, we were able to determine the performance of multiple measures within a single text dimension.

Run 3 In the third configuration, we combined the measures across text dimensions to determine the best overall configuration, i.e. we combined multiple similarity measures regardless of their text dimension. That way, we were able to investigate the effects of computing text similarity along multiple text dimensions.

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.400	.143
Ferret Baseline	.642	.517
Chong et al. (2010)[21]	.705	.641
Clough and Stevenson (2011)		
- our re-implementation[20]	.726	.658
- as reported in their work	.800	.757
Our System	.842	.811

Table 9: Results for the best classification on the Wikipedia Rewrite Corpus for the original 4-way classification

7.4 Results & Discussion

In this section, we report the results obtained on the evaluation datasets, thereby demonstrating the effectiveness of the proposed compositional approach to computing text similarity across the three text dimensions content, structure, and style.

We utilized three datasets for the evaluation of our system which originate in the fields of plagiarism detection, journalistic text reuse detection, and paraphrase recognition: the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011), the METER Corpus (Gaizauskas et al., 2001), and the Webis Crowd Paraphrase Corpus (Burrows et al., 2013), described in detail in Section 5.2.1.

Evaluation was carried out in terms of accuracy and $\bar{F}_1$ score. By accuracy, we refer to the number of correctly predicted texts divided by the total number of texts. As the class distributions in the datasets are skewed, we report the overall $\bar{F}_1$ score as the arithmetic mean across the F_1 scores of all classes (which vary from dataset to dataset) in order to account for the class imbalance (Clough et al., 2002; Sánchez-Vega et al., 2010).

We compare our results with two baselines: the majority class baseline and the word trigram similarity measure Ferret (Lyon et al., 2004) (see Section 7.2). Additionally, we report the best results from the literature for comparison.

7.4.1 Wikipedia Rewrite Corpus

We summarize the results on this dataset in Table 9.[19] In the best configuration, when combining similarity measures across dimensions, our system achieves a performance of $\bar{F}_1 = .811$. It outperforms the best reference system by Clough and Stevenson (2011) by 5.4 points in terms of $\bar{F}_1$ score compared to their reported numbers, and by 15.3 points compared to our re-implementation of this system.[20] Their system uses a Naive Bayes classifier with only a very small feature set: word n-gram containment ($n = 1, 2, \ldots, 5$) and longest common subsequence. For comparison, we re-implemented their system and also applied it to the two datasets in the remainder of this section. We report our findings in Section 7.4.2 and 7.4.3.

In Table 10, we report the best results for the combinations of text similarity measures within and across dimensions. When we combine the measures within their respective dimensions, content outperforms structural and stylistic similarity. However, all combinations of measures across dimensions in addition to content similarity improve the results. The best performance is achieved by combining the three similarity measures longest common subsequence, stopword 10-grams, and character 5-gram profiles from the two dimensions content and structure. This supports our hypothesis that computing text similarity indeed benefits from dimensions other than content.

7.4.2 METER Corpus

For the evaluation of our system on the METER corpus, we followed Sánchez-Vega et al. (2010) and folded the annotations to a binary classification of 181 reused (wholly/partially derived) and 72 non-reused instances in order to carry out a comparable evaluation study.

We summarize the results on this dataset in Table 11. In the best configuration, our system achieves an overall performance of $\bar{F}_1 = .768$. It outperforms the best reference system by Sánchez-Vega et al. (2010) by 6.3 points in

[19] Figures in italics are taken from the literature, while we (re-)implemented the remaining systems. This applies to all result tables in this paper.

[20] While we were able to reproduce the results of the Ferret baseline as reported by Chong et al. (2010), our re-implementation of the system by Clough and Stevenson (2011) (Naive Bayes classifier, same feature set) resulted in a much lower overall performance. We observed the largest difference for the longest common subsequence measure, even though we used a standard implementation (Allison and Dix, 1986) and normalized as described by Clough and Stevenson (2011).

[21] Chong et al. (2010) report $\bar{F}_1 = .698$ in their original work. This figure, however, reflects the weighted arithmetic mean over all four classes of the dataset where one class is twice as prominent as each of the others. As dicussed in Section 7.4, we report all $\bar{F}_1$ scores as the unweighted arithmetic mean in order to account for the class imbalance.

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.747	.693
Structure	.716	.660
Style	.442	.398
Combinations across dimensions		
Content + Style	.800	.757
Content + Structure	.842	.811
Structure + Style	.632	.569
Content + Structure + Style	.832	.798

Table 10: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Wikipedia Rewrite Corpus

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.715	.417
Ferret Baseline	.684	.535
Clough and Stevenson (2011)[22]	.692	.680
Sánchez-Vega et al. (2010)	.783	.705
Our System	.802	.768

Table 11: Results for the best classification on the METER Corpus

terms of $\bar{F}_1$ score. Their system uses a Naive Bayes classifier with two custom features which compare texts based on the length and frequency of common word sequences and the relevance of individual words.

In Table 12, we show that the performance of text reuse detection always improves over individual measures when we combine the measures within their respective dimensions. An exception is the combination of structural similarity measures, which only performs on the same level as the best individual measure part-of-speech 3-grams containment. Combinations of content similarity measures show a better performance than combinations of structural or stylistic measures. Our system achieves its best performance on this dataset when text similarity measures are combined across all three dimensions content, structure, and style. The best configuration resulted from using a Naive Bayes classifier with the following measures: Greedy String Tiling, stopword 12-grams, and Sequential TTR. As for the previous dataset, the effects of dimension combination held true regardless of the classifier used.

The influence of the stylistic similarity measures is particularly interesting to note. In contrast to the Wikipedia Rewrite Corpus, including these measures in the composition improves the results on this dataset: Our classifier is able to detect similarity even for reused texts by expert journalists. This is due to the fact that a journalistic text which reuses the original press agency source most likely also shows stylistic similarity in terms of e.g. vocabulary richness.

7.4.3 Webis Crowd Paraphrase Corpus

We summarize the results on this dataset in Table 13. Even though the Ferret baseline is a strong competitor ($\bar{F}_1 = .789$), our system achieves the best results on this dataset with $\bar{F}_1 = .852$. The results reported by Burrows et al. (2013) are slightly worse ($\bar{F}_1 = .837$). Their best score was achieved by using a k-nearest neighbor classifier with a feature set of 10 similarity measures. They exclusively used similarity measures that operate on the texts' string sequences and thus capture the content dimension of text similarity only, e.g. the Levenshtein (1966) distance and a word n-gram similarity measure.

As for the previous datasets, our hypothesis holds true that the combination of similarity dimensions improves the results: When we combine the similarity features within each of the respective dimensions, the performance numbers increase (see Table 14. The combination of content similarity measures is stronger than the combination of structural and stylistic similarity measures, and performs on the same level as the original results reported by Burrows et al. (2013). This is to be expected, as their system uses a feature set which also addresses the content dimension exclusively.

[22] We report the results for our re-implementation of the system by Clough and Stevenson (2011). In their original work, they did not evaluate on this dataset.

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.759	.712
Structure	.731	.701
Style	.755	.672
Combinations across dimensions		
Content + Style	.779	.733
Content + Structure	.739	.713
Structure + Style	.767	.739
Content + Structure + Style	.802	.768

Table 12: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the METER Corpus

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.517	.341
Ferret Baseline	.794	.789
Clough and Stevenson (2011)[22]	.798	.795
Burrows et al. (2013)	.839	.837
Our System	.853	.852

Table 13: Results for the best classification on the Webis Crowd Paraphrase Corpus

When we combine measures across dimensions, the results improve even further. An exception is the combination of content and structural measures, which performs slightly worse than content measures alone due to the lower performance of structural measures on this dataset. The best configuration of our system resulted from combining all three dimensions content, structure, and style in a single classification model using the decision tree classifier, resulting in $\bar{F}_1 = .852$.

7.5 Section Summary

In this section, we conducted an extrinisic evaluation of our system in the task of text reuse detection. We evaluated our system on three standard datasets where text reuse is prevalent and which originate in the fields of plagiarism detection, journalistic text reuse detection, and paraphrase recognition: the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011), the METER Corpus (Gaizauskas et al., 2001), and the Webis Crowd Paraphrase Corpus (Burrows et al., 2013), which we described in detail in Section 5.2. We showed that the composition of measures consistently outperforms previous approaches across all three datasets.

As we showed, similarity computation works best if the text dimensions are chosen well with respect to the type of text reuse at hand. For the Wikipedia Rewrite Corpus, for example, the stylistic similarity features perform poorly, which is why the composition of all three dimensions performs slightly worse than the combination of only content

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.840	.839
Structure	.816	.814
Style	.819	.817
Combinations across dimensions		
Content + Style	.844	.843
Content + Structure	.838	.838
Structure + Style	.831	.830
Content + Structure + Style	.853	.852

Table 14: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Webis Crowd Paraphrase Corpus

and structural features. For the other two datasets, however, stylistic similarity is a strong dimension within the composition, and consequently the best performance is reached when combining all three text dimensions content, structure, and style.

8 Software Package DKPro Similarity

In the course of the experiments on text similarity presented in this article, we realized that the community could greatly benefit from a full-featured framework for similarity computation—as up to now, developing text similarity measures is a highly scattered effort. Only a few text similarity measures proposed in the literature (see Section 4) are released publicly, and those then typically do not comply with any standardization. This fact was also realized by the organizers of the SemEval-2012 Semantic Textual Similarity Task (see Section 6), as they argue for the creation of an open source framework for text similarity (Agirre et al., 2012).

In order to fill this gap, we developed DKPro Similarity (Bär et al., 2013), an open source framework for text similarity. Our goal was to provide a comprehensive repository of text similarity measures, which are implemented in a common framework using standardized interfaces. Besides the already available measures, DKPro Similarity is easily extensible and intended to allow for custom implementations, for which it offers various templates and examples. It is implemented in Java and released as public Maven[23] modules. The implementation is publicly available at Google Code[24] under the Apache Software License v2 and for some components under GNU GPL v3.

8.1 Architecture

DKPro Similarity is designed to operate in one of two modes: The stand-alone mode allows to use text similarity measures as independent components in any experimental setup, but does not offer means for further language processing, e.g. lemmatization. The UIMA-coupled mode tightly integrates similarity computation with full-fledged Apache UIMA-based language processing pipelines. That way, it allows performing any number of language processing steps, e.g. co-reference or named-entity resolution, along with the text similarity computation.

Stand-alone Mode

In this mode, text similarity measures can be used independently of any language processing pipeline just by passing them a pair of texts as (i) two strings, or (ii) two lists of strings (e.g. already lemmatized texts). We therefore provide an API module, which contains Java interfaces and abstract base classes for the measures. That way, DKPro Similarity allows for a maximum flexibility in experimental design, as the text similarity measures can easily be integrated with any existing experimental setup:

```
1 TextSimilarityMeasure measure = new GreedyStringTiling();
2 double similarity = measure.getSimilarity(text1, text2);
```

The above code snippet instantiates the Greedy String Tiling measure (Wise, 1996) and then computes the text similarity between the given pair of texts. The resulting similarity score is normalized into $[0,1]$ where 0 means not similar at all, and 1 corresponds to perfectly similar.[25] By using the common `TextSimilarityMeasure` interface, it is easy to replace Greedy String Tiling with any measure of choice, such as Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007). We give an overview of measures available in DKPro Similarity in the following section.

UIMA-coupled Mode

In this mode, DKPro Similarity allows text similarity computation to be directly integrated with any UIMA-based language processing pipeline. That way, it is easy to use text similarity components in addition to other UIMA-based components in the same pipeline. For example, an experimental setup may require to first compute text similarity scores and then to run a classification algorithm on the resulting scores.

In Figure 6, we show a graphical overview of the integration of text similarity measures (right) with a UIMA-based pipeline (left). The pipeline starts by reading a given dataset, then performs any number of pre-processing steps such as tokenization, sentence splitting, lemmatization, or stopword filtering, then runs the text similarity computation, before executing any subsequent post-processing steps and finally returning the processed texts in a suitable format for evaluation or manual inspection. As all text similarity measures in DKPro Similarity conform to standardized interfaces, they can be easily exchanged in the text similarity computation step.

With DKPro Similarity, we offer various subclasses of the generic UIMA components which are specifically tailored towards text similarity experiments, e.g. corpus readers for standard evaluation datasets as well as evaluation

[23] http://maven.apache.org

[24] http://code.google.com/p/dkpro-similarity-asl

[25] Some string distance measures such as the Levenshtein distance (Levenshtein, 1966) return a raw distance score where less distance corresponds to higher similarity. However, the score can easily be normalized, e.g. by text length.

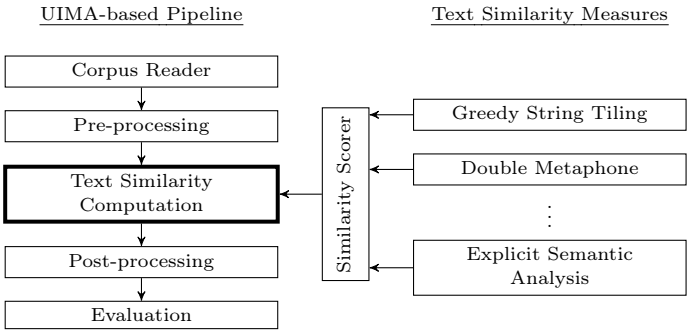

Figure 6: DKPro Similarity allows to integrate any text similarity measure (right) which conforms to standardized interfaces into a UIMA-based language processing pipeline (left) by means of a dedicated Similarity Scorer component (middle).

components for running typical evaluation metrics. By leveraging UIMA's architecture, we also define an additional interface to text similarity measures: The `JCasTextSimilarityMeasure` inherits from `TextSimilarityMeasure`, and adds a method for two `JCas` text representations:[26]

```
double getSimilarity(JCas text1, JCas text2);
```

The additional interface allows to implement measures which have full access to UIMA's document structure. That way, it is possible to create text similarity measures which can use any piece of information that has been annotated in the processed documents, such as dependency trees or morphological information.

A large number of measures presented in Section 4 are already available in the corresponding modules. As the project is actively under development, we do not provide a comprehensive list of measures here, as it would soon be outdated. Among many others, popular measures implemented in DKPro Similarity include compositional text similarity measures based on pairwise word similarity scores such as the one proposed by Mihalcea et al. (2006), and a set of non-compositional text similarity measures such Greedy String Tiling (Wise, 1996) and the Levenshtein distance (Levenshtein, 1966), as well as Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) and Latent Semantic Analysis (Landauer et al., 1998).

8.2 Experimental Setups

DKPro Similarity further encourages the creation and publication of complete experimental setups. That way, we promote the reproducibility of experimental results, and provide reliable, permanent experimental conditions which can benefit future studies and help to stimulate the reuse of particular experimental steps and software modules.

The experimental setups are instantiations of the generic UIMA-based language processing pipeline depicted in Figure 6 and are designed to precisely match the particular task at hand. They thus come pre-configured with corpus readers for the relevant input data, with a set of pre- and post-processing as well as evaluation components, and with a set of text similarity measures which are well suited for the particular task. The experimental setups are self-contained systems and can be run out-of-the-box without further configuration.[27]

DKPro Similarity contains ready-made setups for reproducing the results reported in this article. DKPro Similarity has also been used for other tasks such as question answering (Walter et al., 2012). We would like to encourage other researchers to participate in our efforts and invite them to explore our existing experimental setups as outlined above, run modified versions of our setups, and contribute own text similarity measures to the framework. For that, DKPro Similarity also comes with an example module for getting started, which guides first-time users through both the stand-alone and the UIMA-coupled modes.

9 Conclusions

In the natural language processing community, text similarity is fundamental to a variety of tasks and applications, e.g. automatic essay grading and question answering. Contrary to the notion of similarity in psychology, though, text similarity traditionally has been a loose notion and is much less well-defined than its psychological counterpart. We argued that a major shortcoming of previous text similarity research is the fact that no attempt has been made yet to formalize in what way text similarity between two texts can be computed. Still, text similarity is regarded as a

[26] The `JCas` is an object-oriented Java interface to the Common Analysis Structure (Ferrucci and Lally, 2004), Apache UIMA's internal document representation format.

[27] A one-time setup of local lexical-semantic resources such as WordNet may be necessary, though.

fixed, axiomatic notion in the community. In consequence, we proposed to define text similarity as a notion which can be judged along multiple text dimensions, i.e. characteristics inherent to texts. For the application to texts, we proposed to focus on three generic text dimensions for which we provided empirical evidence: content, structure, and style. Based on the analysis, we combined a multitude of text similarity measures along these dimensions in a unified classification model. As we showed, this system consistently outperformed prior work and competing systems in both an intrinsic and an extrinsic evaluation:

Intrinsic Evaluation

We first applied our system in an intrinsic evaluation, namely the Semantic Textual Similarity (STS) Task (Agirre et al., 2012) as part of the Semantic Evaluation (SemEval) exercises. Across all five evaluation datasets, our system performed best in two out of three official evaluation metrics and was ranked #2 for the third metric when comparing the system output to human judgments. While we did not reach the highest scores on any of the single datasets, our system was most robust across different data. In the experiments, we showed that text similarity can be best detected if a multitude of measures—each addressing different text characteristics—are combined in a single classification model. Due to the nature of the data in this task, a combination of measures along the content dimension showed the best results.

Extrinsic Evaluation

In an extrinsic evaluation, we applied our system to the task of text reuse detection. In this task, text pairs were classified by our system according to the degree of text reuse they exhibit and then compared with human classifications. In these experiments, we empirically showed that text reuse can be best detected if measures are combined across multiple text dimensions. For two out of three datasets, the best performance was reached when combining all three dimensions content, structure, and style in a single classification model, while for the third datasets stylistic measures showed a poor performance and in consequence the best performance was reached when combining only measures from the content and the structure dimensions. Across all three datasets, our system outperformed all previous work on this data.

Future Work

The greatest obstacle that hinders the broad adoption of our research in future work is the lack of suitable evaluation datasets. For example, in the data by Agirre et al. (2012) two major issues remain: (a) It is unclear how to judge similarity between pairs of texts which contain contextual references such as on Monday vs. after the Thanksgiving weekend. (b) For several pairs, it is unclear what point of view to take, e.g. for the pair An animal is eating. / The animal is hopping. Is the pair to be considered similar (an animal is doing something) or rather not (eating vs. hopping)? As long as there are no more specific instructions given on how to judge text similarity for such cases, there will inevitably be variations in text similarity judgments—both for judgments by human subjects as well as by text similarity measures.

Apart from that, the community also lacks a generalized framework which can be used to compute text similarity. While some frameworks for similarity computation already exist, e.g. the S-Space Package or the SimMetrics Library, none of them is specifically tailored towards text similarity. In this article, we thus introduced DKPro Similarity, an open source package for developing text similarity measures as well as complete experimental setups. Our goal here is to promote the reproducibility of experimental results, and to provide reliable, permanent experimental conditions which can benefit future studies and help foster the reuse of particular experimental steps and software modules. We hope to stimulate further research in this field and foster future developments of text similarity measures and experimental setups.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Klaus Tschira Foundation under project No. 00.133.2008. We thank Chris Biemann for his inspirations, as well as Carolin Deeg, Andriy Nadolskyy, and Artem Vovk for their participation in the annotation studies. We also thank the organizers of the STS Task at SemEval-2012 for sharing their annotation data with us.

References

Abdel-Hamid, O., Behzadi, B., Christoph, S., and Henzinger, M. (2009). Detecting the Origin of Text Segments Efficiently. In Proceedings of the 18th International Conference on World Wide Web, pages 61–70, Madrid, Spain.

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics, pages 385–393, Montreal, Canada.

Allison, L. and Dix, T. I. (1986). A Bit-String Longest-Common-Subsequence Algorithm. Information Processing Letters, 23:305–310.

Attali, Y. and Burstein, J. (2006). Automated Essay Scoring with e-rater v.2. Journal of Technology, Learning, and Assessment, 4(3).

Bär, D., Zesch, T., and Gurevych, I. (2011). A Reflective View on Text Similarity. In Proceedings of the International Conference on Recent Advances in Natural Language Processing, pages 515–520, Hissar, Bulgaria.

Bär, D., Zesch, T., and Gurevych, I. (2012). Text Reuse Detection Using a Composition of Text Similarity Measures. In Proceedings of the 24th International Conference on Computational Linguistics, pages 167–184, Mumbai, India.

Bär, D., Zesch, T., and Gurevych, I. (2013). DKPro Similarity: An Open Source Framework for Text Similarity. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations), pages 121–126.

Barzilay, R. and Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, pages 10–17, Madrid, Spain.

Biemann, C. (2012a). Creating a System for Lexical Substitutions from Scratch using Crowdsourcing. Language Resources and Evaluation: Special Issue on Collaboratively Constructed Language Resources, 47(1):97–122.

Biemann, C. (2012b). Turk Bootstrap Word Sense Inventory 2.0: A Large-Scale Resource for Lexical Substitution. In Proceedings of the 8th International Conference on Language Resources and Evaluation, pages 4038–4042, Istanbul, Turkey.

Boonthum, C. (2004). iSTART: Paraphrase Recognition. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 31–36, Barcelona, Spain.

Broder, A. Z. (1997). On the Resemblance and Containment of Documents. In Proceedings of Compression and Complexity of Sequences, pages 21–29, Salerno, Italy.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. (1997). Syntactic Clustering of the Web. In Proceedings of the 6th International World Wide Web Conference, pages 1157–1166, Santa Clara, CA, USA.

Burrows, S., Potthast, M., and Stein, B. (2013). Paraphrase Acquisition via Crowdsourcing and Machine Learning. Transactions on Intelligent Systems and Technology, 4(3):1–22.

Callison-Burch, C. and Dredze, M. (2010). Creating Speech and Language Data With Amazon's Mechanical Turk. In Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 1–12, Los Angeles, CA, USA.

Cavnar, W. B. and Trenkle, J. M. (1994). N-Gram-Based Text Categorization. In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, pages 161–175, Las Vegas, NV, USA.

Chandrasekar, R., Doran, C., and Srinivas, B. (1996). Motivations and Methods for Text Simplification. In Proceedings of the 16th International Conference on Computational Linguistics, pages 1041–1044, Copenhagen, Denmark.

Charikar, M. S. (2002). Similarity Estimation Techniques from Rounding Algorithms. In Proceedings of the 34th Annual Symposium on Theory of Computing, pages 380–388, Montreal, Canada.

Chen, D. L. and Dolan, W. B. (2011). Collecting Highly Parallel Data for Paraphrase Evaluation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 190–200, Portland, OR, USA.

Chong, M., Specia, L., and Mitkov, R. (2010). Using Natural Language Processing for Automatic Detection of Plagiarism. In Proceedings of the 4th International Plagiarism Conference, Newcastle upon Tyne, UK.

Clough, P. (2003). Old and New Challenges in Automatic Plagiarism Detection. Technical report, National UK Plagiarism Advisory Service.

Clough, P., Gaizauskas, R., Piao, S. S., and Wilks, Y. (2002). METER: MEasuring TExt Reuse. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pages 152–159, Philadelphia, PA, USA.

Clough, P. and Stevenson, M. (2011). Developing a Corpus of Plagiarised Short Answers. Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis, 45(1):5–24.

Cohn, T., Callison-Burch, C., and Lapata, M. (2008). Constructing Corpora for the Development and Evaluation of Paraphrase Systems. Computational Linguistics, 34(4):597–614.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In Machine Learning Challenges, Lecture Notes in Computer Science, pages 177–190. Springer.

Dinu, L. P. and Popescu, M. (2009). Ordinal Measures in Authorship Identification. In Proceedings of the 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse, pages 62–66, San Sebastian, Spain.

Dolan, W. B., Quirk, C., and Brockett, C. (2004). Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In Proceedings of the 20th International Conference on Computational Linguistics, pages 350–356, Geneva, Switzerland.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press.

Ferrucci, D. and Lally, A. (2004). UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering, 10(3-4):327–348.

Gabrilovich, E. and Markovitch, S. (2007). Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 1606–1611, Hyderabad, India.

Gaizauskas, R., Foster, J., Wilks, Y., Arundel, J., Clough, P., and Piao, S. (2001). The METER Corpus: A Corpus for Analysing Journalistic Text Reuse. In Proceedings of the Corpus Linguistics 2001 Conference, pages 214–223, Bailrigg, UK.

Gärdenfors, P. (2000). Conceptual Spaces: The Geometry of Thought. MIT Press.

Goodman, N. (1972). Seven Strictures on Similarity. In Problems and Projects, pages 437–446. Bobbs-Merrill.

Gusfield, D. (1997). Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, 11(1):10–18.

Hannabuss, S. (2001). Contested Texts: Issues of Plagiarism. Library Management, 22(6/7):311–318.

Hatzivassiloglou, V., Klavans, J. L., and Eskin, E. (1999). Detecting Text Similarity Over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pages 203–212, College Park, MD, USA.

Hearst, M. A. (1997). TextTiling: Segmenting Text Into Multi-Paragraph Subtopic Passages. Computational Linguistics, 23(1):33–64.

Henzinger, M. (2006). Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 284–291, Seattle, WA, USA.

Hirst, G. and St. Onge, D. (1998). Lexical Chains as Representations of Contexts for the Detection and Correction of Malapropisms. In Fellbaum, C., editor, WordNet: An Electronic Lexical Database. MIT Press.

Ho, C., Azrifah, M., Murad, A., Kadir, R. A., and Doraisamy, S. C. (2010). Word Sense Disambiguation-Based Sentence Similarity. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 418–426, Beijing, China.

Hoad, T. C. and Zobel, J. (2003). Methods for Identifying Versioned and Plagiarized Documents. Journal of the American Society of Information Science and Technology, 54(3):203–215.

Islam, A. and Inkpen, D. (2006). Second Order Co-Occurrence PMI for Determining the Semantic Similarity of Words. In Proceedings of the International Conference on Language Resources and Evaluation, pages 1033–1038, Genoa, Italy.

Islam, A. and Inkpen, D. (2008). Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. ACM Transactions on Knowledge Discovery from Data, 2(2):1–25.

Islam, A., Milios, E., and Keselj, V. (2012). Text Similarity Using Google Tri-Grams. In Proceedings of the 25th Canadian Conference on Artificial Intelligence, pages 312–317, Toronto, Canada.

Jaro, M. A. (1989). Advances in Record Linkage Methodology as Applied to the 1985 Census of Tampa Florida. Journal of the American Statistical Association, 84(406):414–420.

Jiang, J. J. and Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In Proceedings of the 10th International Conference on Research in Computational Linguistics, pages 19–33, Taiwan.

Joy, M. and Luck, M. (1999). Plagiarism in Programming Assignments. IEEE Transactions of Education, 42(2):129–133.

Kennedy, A. and Szpakowicz, S. (2008). Evaluating Roget's Thesauri. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 416–424, Columbus, OH, USA.

Keselj, V., Peng, F., Cercone, N., and Thomas, C. (2003). N-Gram-Based Author Profiles for Authorship Attribution. In Proceedings of the Conference of the Pacific Association for Computational Linguistics, pages 255–264, Halifax, Canada.

Knight, K. and Marcu, D. (2002). Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. Artificial Intelligence, 139(1):91–107.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In Proceedings of the 10th Machine Translation Summit, pages 79–86, Phuket Island, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Demo and Poster Sessions, pages 177–180, Prague, Czech Republic.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An Introduction to Latent Semantic Analysis. Discourse Processes, 25(2):259–284.

Leacock, C. and Chodorow, M. (1998). Combining Local Context and Wordnet Similarity for Word Sense Identification. In Fellbaum, C., editor, WordNet: An Electronic Lexical Database. MIT Press.

Leacock, C. and Chodorow, M. (2003). C-rater: Automated Scoring of Short-Answer Questions. Computers and the Humanities, 37(4):389–405.

Lee, J. (2007). A Computational Model of Text Reuse in Ancient Literary Texts. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 472–479, Prague, Czech Republic.

Lee, M. D., Pincombe, B., and Welsh, M. (2005). An Empirical Evaluation of Models of Text Document Similarity. In Proceedings of the 27th Annual Conference of the Cognitive Science Society, pages 1254–1259, Stresa, Italy.

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone From an Ice Cream Cone. In Proceedings of the 5th Annual International Conference on Systems Documentation, pages 24–26, New York, NY, USA.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady, 10(8):707–710.

Li, Y., McLean, D., Bandar, Z., O'Shea, J., and Crockett, K. (2006). Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Transactions on Knowledge and Data Engineering, 18(8):1138–1150.

Lin, D. (1998a). An Information-Theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning, pages 296–304, Madison, WI, USA.

Lin, D. (1998b). Automatic Retrieval and Clustering of Similar Words. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, pages 768–774, Montreal, Canada.

Lin, D. and Pantel, P. (2001). Discovery of Inference Rules for Question Answering. Natural Language Engineering, 7(4):343–360.

Lyon, C., Barrett, R., and Malcolm, J. (2004). A Theoretical Basis to the Automated Detection of Copying Between Texts, and its Practical Implementation in the Ferret Plagiarism and Collusion Detector. In Proceedings of the Conference on Plagiarism: Prevention, Practice and Policies, Newcastle upon Tyne, UK.

Lyon, C., Malcolm, J., and Dickerson, B. (2001). Detecting Short Passages of Similar Text in Large Document Collections. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 118–125, Pittsburgh, PA USA.

Manku, G. S., Jain, A., and Sarma, A. D. (2007). Detecting Near-Duplicates for Web Crawling. In Proceedings of the 16th International World Wide Web Conference, pages 141–149, Banff, Canada.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

McCarthy, P. M. and Jarvis, S. (2010). MTLD, vocd-D, and HD-D: A Validation Study of Sophisticated Approaches to Lexical Diversity Assessment. Behavior Research Methods, 42(2):381–392.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In Proceedings of the 21st National Conference on Artificial Intelligence, pages 775–780, Boston, MA, USA.

Mohler, M. and Mihalcea, R. (2009). Text-to-text Semantic Similarity for Automatic Short Answer Grading. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pages 567–575, Athens, Greece.

Monge, A. and Elkan, C. (1997). An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery, pages 23–29, Tucson, AZ, USA.

Mosteller, F. and Wallace, D. L. (1964). Inference and Disputed Authorship: The Federalist. Addison-Wesley.

Palmer, M., Gildea, D., and Xue, N. (2010). Semantic Role Labeling. Synthesis Lectures on Human Language Technologies, 3(1):1–103.

Ponzetto, S. P. and Navigli, R. (2010). Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1522–1531, Uppsala, Sweden.

Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., and Rosso, P. (2010). Overview of the 2nd International Competition on Plagiarism Detection. In Notebook Papers of CLEF 10 Labs and Workshops, Padua, Italy.

Potthast, M., Gollub, T., Hagen, M., Graß egger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., and Stein, B. (2012). Overview of the 4th International Competition on Plagiarism Detection. In CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, Rome, Italy.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and Application of a Metric on Semantic Nets. IEEE Transactions on Systems, Man, and Cybernetics, 19(1):17–30.

Ramage, D., Rafferty, A. N., and Manning, C. D. (2009). Random Walks for Text Semantic Similarity. In Proceedings of the Workshop on Graph-based Methods for Natural Language Processing, pages 23–31, Singapore.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pages 448–453, Montreal, Canada.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual Correlates of Synonymy. Communications of the ACM, 8(10):627–633.

Salton, G. and McGill, M. J. (1983). Introduction to Modern Information Retrieval. McGraw-Hill.

Sánchez-Vega, F., Villaseñor Pineda, L., Montes-y Gómez, M., and Rosso, P. (2010). Towards Document Plagiarism Detection Based on the Relevance and Fragmentation of the Reused Text. In Proceedings of the 9th Mexican International Conference on Artificial Intelligence, pages 24–31, Pachuca, Mexico.

Sarić, F., Glavas, G., Karan, M., Snajder, J., and Basić, B. D. (2012). TakeLab: Systems for Measuring Semantic Text Similarity. In Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics, pages 441–448, Montreal, Canada.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK.

Sinclair, J. (2001). Collins COBUILD Advanced Learner's English Dictionary. HarperCollins, 3rd edition.

Smith, L. B. and Heise, D. (1992). Perceptual Similarity and Conceptual Structure. In Burns, B., editor, Percepts, Concepts, and Categories, pages 233–272. Elsevier.

Stamatatos, E. (2011). Plagiarism Detection Using Stopword N-Grams. Journal of the American Society for Information Science and Technology, 62(12):2512–2527.

Templin, M. C. (1957). Certain Language Skills in Children. University of Minnesota Press.

Tsatsaronis, G., Varlamis, I., and Vazirgiannis, M. (2010). Text Relatedness Based on a Word Thesaurus. Journal of Artificial Intelligence Research, 37:1–39.

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of the 12th European Conference on Machine Learning, pages 491–502, Freiburg, Germany.

Tversky, A. (1977). Features of Similarity. Psychological Review, 84(4):327–352.

Walter, S., Unger, C., Cimiano, P., and Bär, D. (2012). Evaluation of a Layered Approach to Question Answering over Linked Data. In Proceedings of the 11th International Semantic Web Conference, pages 362–374, Boston, MA, USA.

Widdows, D. (2004). Geometry and Meaning. Center for the Study of Language and Information, Stanford, CA, USA.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In Proceedings of the Survey Research Methods Section, pages 354–359.

Wise, M. J. (1996). YAP3: Improved Detection of Similarities in Computer Program and Other Texts. In Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education, pages 130–134, Philadelphia, PA, USA.

Wu, Z. and Palmer, M. (1994). Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pages 133–138, Las Cruces, NM, USA.

Yang, D. and Powers, D. M. (2005). Measuring Semantic Similarity in the Taxonomy of WordNet. In Proceedings of the 28th Australasian Computer Science Conference, pages 315–332, Newcastle, Australia.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196, Cambridge, MA, USA.

Yeh, E., Ramage, D., Manning, C. D., Agirre, E., and Soroa, A. (2009). WikiWalk: Random Walks on Wikipedia for Semantic Relatedness. In Proceedings of the Workshop on Graph-based Methods for Natural Language Processing, pages 41–49, Singapore.

Yule, G. U. (1939). On Sentence-Length as a Statistical Characteristic of Style in Prose: With Application to Two Cases of Disputed Authorship. Biometrika, 30(3/4):363–390.

Zesch, T. and Gurevych, I. (2010). Wisdom of Crowds versus Wisdom of Linguists – Measuring the Semantic Relatedness of Words. Journal of Natural Language Engineering, 16(1):25–59.