A Composite Model for Computing Similarity Between Texts

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr.-Ing.

vorgelegt von
Dipl.-Inform. Daniel Bär
geboren in Würzburg

Tag der Einreichung:	20. August 2013
Tag der Disputation:	11. Oktober 2013
Referenten:	Prof. Dr. Iryna Gurevych
	Prof. Ido Dagan, Ph.D.
	Dr. Torsten Zesch

Darmstadt 2013
D17

Ehrenwörtliche Erklärung[1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades Dr.-Ing. mit dem Titel "A Composite Model for Computing Similarity Between Texts" selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 20. August 2013

Dipl.-Inform. Daniel Bär

[1] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Wissenschaftlicher Werdegang des Verfassers[2]

10/02 – 11/08	Studium der Informatik mit Nebenfach Linguistik an der Julius-Maximilians-Universität Würzburg
06/08 – 11/08	Diplomarbeit am Lehrstuhl für Künstliche Intelligenz und Angewandte Informatik der Universität Würzburg in Kooperation mit SAP AG (SAP Research), Karlsruhe, mit dem Titel "User-centered Annotation Concepts for the Semantic Web"
01/09 – 05/09	Wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Graphische Datenverarbeitung, Darmstadt
06/09 – 08/13	Wissenschaftlicher Mitarbeiter am Fachgebiet Ubiquitous Knowledge Processing der Technischen Universität Darmstadt

[2]Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Abstract

Computing *text similarity* is a foundational technique for a wide range of tasks in natural language processing such as duplicate detection, question answering, or automatic essay grading. Just recently, text similarity received wide-spread attention in the research community by the establishment of the *Semantic Textual Similarity (STS) Task* at the *Semantic Evaluation (SemEval)* workshop in 2012—a fact that stresses the importance of text similarity research. The goal of the STS Task is to create automated measures which are able to compute the degree of similarity between two given texts in the same way that humans do. Measures are thereby expected to output continuous text similarity scores, which are then either compared with human judgments or used as a means for solving a particular problem.

We start this thesis with 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. 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 fixed, axiomatic notion in the community. To alleviate this shortcoming, we describe existing formal models of similarity and discuss how we can adapt them to texts. We propose to judge text similarity along multiple *text dimensions*, i.e. characteristics inherent to texts, and provide empirical evidence based on a set of annotation studies that the proposed dimensions are perceived by humans.

We continue with a comprehensive survey of state-of-the-art text similarity measures previously proposed in the literature. To the best of our knowledge, no such survey has been done yet. We propose a classification into *compositional* and *non-compositional* text similarity measures according to their inherent properties. Compositional measures compute text similarity based on pairwise word similarity scores between all words which are then aggregated to an overall similarity score, while non-compositional measures project the complete texts onto particular models and then compare the texts based on these models.

Based on our theoretical insights, we then present the implementation of a text similarity system which composes a multitude of text similarity measures along multiple text dimensions using a machine learning classifier. Depending on the concrete task at hand, we argue that such a system may need to address more than a single text dimension in order to best resemble human judgments. Our efforts culminate in the open source framework *DKPro Similarity*, which streamlines the development of text similarity measures and experimental setups.

We apply our system in two evaluations, for which it consistently outperforms prior work and competing systems: an *intrinsic* and an *extrinsic evaluation*. In the intrinsic evaluation, the performance of text similarity measures is evaluated in an isolated setting by comparing the algorithmically produced scores with human judgments. We conducted the intrinsic evaluation in the context of the STS Task as part of the SemEval workshop. In the 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 particular problem. We conducted the extrinsic evaluation in the text classification task of *text reuse detection*. The results of both evaluations support our hypothesis that a composition of text similarity measures highly benefits the similarity computation process.

Finally, we stress the importance of text similarity measures for real-world applications. We therefore introduce the application scenario *Self-Organizing Wikis*, where users of *wikis*, i.e. web-based collaborative content authoring systems, are supported in their everyday tasks by means of natural language processing techniques in general, and text similarity in particular. We elaborate on two use cases where text similarity computation is particularly beneficial: the detection of duplicates, and the semi-automatic insertion of hyperlinks. Moreover, we discuss two further applications where text similarity is a valuable tool: In both *question answering* and *textual entailment recognition*, text similarity has been used successfully in experiments and appears to be a promising means for further research in these fields.

We conclude this thesis with an analysis of shortcomings of current text similarity research and formulate challenges which should be tackled by future work. In particular, we believe that computing text similarity along multiple text dimensions—which depend on the specific task at hand—will benefit any other task where text similarity is fundamental, as a composition of text similarity measures has shown superior performance in both the intrinsic as well as the extrinsic evaluation.

Zusammenfassung

Die Berechnung von *Textähnlichkeit* ist eine grundlegende Technik für ein breites Anwendungsspektrum in der automatischen Sprachverarbeitung, wie etwa der Duplikatserkennung, der Beantwortung natürlich-sprachlicher Fragen, oder auch der automatisierten Bewertung von Essays. Durch die Einrichtung des *Semantic Textual Similarity* Wettbewerbs im Rahmen des *Semantic Evaluation (SemEval)* Workshops im Jahr 2012 kam dem Thema Textähnlichkeit große Aufmerksamkeit in der wissenschaftlichen Gemeinde zugute – ein deutlicher Beleg dafür, dass hier aktuell großer Forschungsbedarf besteht. Ziel dieses Wettbewerbs ist es, maschinelle Maße zu entwickeln, die fähig sind, Ähnlichkeit zwischen zwei gegebenen Texten auf die gleiche Weise zu ermitteln, wie es auch Menschen tun. Von diesen Maßen wird dabei erwartet, Ähnlichkeitswerte auf einer kontinuierlichen Skala zu produzieren, die im Anschluss entweder direkt mit menschlichen Referenzbewertungen verglichen werden, oder als Hilfsmittel zur Lösung eines konkreten Problems dienen.

Wir beginnen diese Arbeit mit der Feststellung, dass der Begriff der *Ähnlichkeit* in der Psychologie zwar wohldefiniert ist, im Gegensatz dazu aber dem Begriff der *Textähnlichkeit* in unserer wissenschaftlichen Gemeinde nur eine rudimentäre Definition zugrunde liegt. Bisher gab es unseres Wissens keinen konkreten Versuch, zu formalisieren, *auf welche Weise* Texte denn überhaupt ähnlich sein können. Noch bis heute wird Textähnlichkeit ausschließlich als pauschalisierter Begriff verwendet. Um diesen Missstand zu beheben, beschreiben wir existierende formale Ähnlichkeitsmodelle und diskutieren, wie wir diese für Texte zuschneiden können. Wir schlagen vor, Textähnlichkeit anhand mehrerer *Textdimensionen* zu bestimmen, d.h. anhand von Merkmalen, die Texten zueigen sind. Im Rahmen mehrerer Annotationsstudien zeigen wir, dass die vorgeschlagenen Dimensionen in der Tat von Menschen zur Ähnlichkeitsbewertung von Texten herangezogen werden.

Im Anschluss zeigen wir eine gründliche Analyse des aktuellen Forschungsstandes zu Textähnlichkeitsmaßen auf, die unseres Wissens die bisher erste umfassende Analyse in diesem Bereich darstellt. Wir schlagen vor, die bestehenden Maße in zwei Merkmalsklassen einzuteilen: *Aggregierende Maße* berechnen zunächst paarweise Wortähnlichkeiten zwischen allen Wörtern der gegebenen Texte und aggregieren diese im Anschluss, um einen finalen Textähnlichkeitswert zu erhalten. *Nicht-aggregierende Maße* hingegen bilden die gegebenen Texte auf bestimmte Modelle ab und vergleichen im Anschluss die Texte ausschließlich anhand dieser Modelle.

Vor dem Hintergrund unserer theoretischen Analysen, die wir zu Beginn dieser Arbeit aufzeigten, entwerfen wir nun die Implementierung eines Textähnlichkeitssystems, welches eine Vielzahl von Textähnlichkeitsmaßen anhand verschiedener Textdimensionen im Rahmen eines maschinellen Lernverfahrens vereint. Wir argumentieren, dass ein solches System – abhängig von der konkreten Aufgabe – mehr als eine Textdimension in Betracht ziehen sollte, um menschliche Ähnlichkeitsbewertungen bestmöglich nachzubilden. Unsere Arbeiten münden schließlich in der quelloffenen Softwarebibliothek *DKPro Similarity*, welche die Entwicklung von Textähnlichkeitsmaßen anhand standardisierter Schnittstellen erlaubt, sowie dazu anregen soll, in einfacher Weise Experimentaufbauten im Hinblick auf die Reproduzierbarkeit der Ergebnisse der wissenschaftlichen Gemeinde zur Verfügung zu stellen.

Wir evaluieren unser System anschließend sowohl *intrinsisch* als auch *extrinsisch*,

wobei es in beiden Fällen durchgängig besser abschneidet als alle früheren Arbeiten und konkurrierenden Systeme. In der intrinsischen Evaluation messen wir die Güte der Textähnlichkeitsmaße in einem isolierten Versuchsaufbau und vergleichen die maschinell erzeugten Ähnlichkeitswerte mit denen menschlicher Studienteilnehmer. Wir führten diese Evaluation im Rahmen des SemEval Workshops im *Semantic Textual Similarity* Wettbewerb durch. Im Gegensatz dazu messen wir in der extrinsischen Evaluation die Güte der Textähnlichkeitsmaße nicht direkt, sondern im Rahmen einer konkreten Problemstellung. Wir führten die extrinsische Evaluation für eine Textklassifizierungsaufgabe durch, in welcher der Grad von *Textwiederverwendung* zwischen zwei Texten ermittelt wird. Die Ergebnisse beider Evaluationen stützen unsere Annahme, dass ein System zur Berechnung von Textähnlichkeit deutlich davon profitiert, eine Kombination mehrerer Maße zu verwenden.

Im finalen Teil der Arbeit betonen wir die besondere Bedeutung von Textähnlichkeit für reale Problemstellungen. Wir gehen dazu zunächst auf das Anwendungsszenario der *Selbstorganisierenden Wikis* ein. In diesem Szenario werden Benutzer von *Wikis*, d.h. kollaborativen Werkzeugen für das Internet-basierte Wissensmanagement, bei ihren täglichen Aufgaben durch Methoden der automatischen Sprachverarbeitung unterstützt, insbesondere auch durch Zuhilfenahme von Textähnlichkeitsmaßen. Wir diskutieren zwei Einsatzfelder im Besonderen: Die Erkennung von Duplikaten sowie das halbautomatisierte Einfügen von Querbezügen. Darüber hinaus gehen wir auf zwei weitere Anwendungen ein, in denen Textähnlichkeitsmaße bereits sehr vielversprechend eingesetzt wurden: Die Beantwortung natürlich-sprachlicher Fragen und das Erkennen von logischen Schlussfolgerungen.

Wir schließen diese Arbeit mit einer Analyse aktuell offener Forschungsfragen ab und formulieren dabei Herausforderungen, denen in zukünftigen Arbeiten begegnet werden sollte. Gestützt auf die positiven Ergebnisse unserer beiden Evaluationen sind wir der festen Überzeugung, dass der vorgeschlagene Weg, Textähnlichkeit anhand verschiedener Textdimensionen zu berechnen, die jeweils abhängig von der konkreten Aufgabe sind, auch andere verwandte Problemstellungen in der automatischen Sprachverarbeitung nachhaltig positiv beeinflussen wird.

Acknowledgements

First and foremost, I would like to thank my principal supervisor Iryna Gurevych that she gave me the chance to pursue my doctoral studies. I am enormously grateful to her for all her guidance, patience and support especially at the beginning of this work. I also thank Ido Dagan for the time he spent with me discussing my work. In particular, I am thankful to him for pointing me to the ongoing development efforts of the novel platform for textual inference named *Excitement*, which lead to the integration of *DKPro Similarity* with this platform.

I wholeheartedly thank Torsten Zesch, who co-supervised this thesis during the last couple of years. Without his most generous support and endless motivations, this work would not have been possible. He is a wonderful person to work with, and I am truly and deeply indebted to him for his endless feedback on the experimental work and publications. It was invaluable. He is an exceptionally skilled mentor and always helped me find a path back when I felt totally lost—which happened more than just once. I truly wish that many, many more students will get the opportunity to work under his supervision in the future.

Thanks also to the scientific community, in particular the anonymous peers that reviewed our publications and provided valuable feedback. I also thank the organizers of the pilot *Semantic Textual Similarity Task* at the SemEval-2012 workshop for providing a wonderful setting for text similarity evaluation: Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. For financial support, I thank the Klaus Tschira Foundation for supporting this work under project No. 00.133.2008.

I am enormously grateful to all my friends who still make my time in Darmstadt so worthwhile. Thanks for everything. Thanks also to all colleagues at our group. In particular, I would like to thank Nicolai Erbs, Christian M. Meyer and Richard Eckart de Castilho for sharing an office with me for so long. A special thanks to Nicolai for finding ever new ways of making time in the office so entertaining, to Niklas Jakob for inviting me to the regular soccer matches on Tuesdays, to Michèle Spankus for continuously putting up motivation stickers on my desk, and to Torsten Zesch and Oliver Ferschke for the great night out in Seattle. A special thanks also goes out to Johannes Hoffart. Since the beginning of our diploma studies, he was and still is an invaluable companion on all matters. He is an exceptionally nice person to dicuss all sorts of things with, and always reminded me of this work's big picture when I was about to get stuck in details.

I also thank my family for giving me the chance to pursue my studies and for their continuous support and motivation. Above all, I thank Jenny. She always had an open ear for my worries and encouraged me everlastingly. Without her, I would have stopped working on this thesis (literally) a hundred times. Thank you so much for making the past four years the most wonderful time of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Text similarity is the task of determining the degree of similarity between two texts. While texts may vary in length, e.g. from single words to paragraphs to complete novels or even books, we regard a sentence as the minimum length for texts under analysis, henceforth. Single words constitute a special case of text similarity which is commonly referred to as the task of computing *word similarity* (Zesch and Gurevych, 2010) and is not the focus of this thesis. As an example for text similarity, consider the following text pair which is taken from the dataset by Li et al. (2006) which we will dicuss in detail in Section 4.1.1:

(a) *A gem is a jewel or stone that is used in jewellery.* (1.1)

(b) *A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces.*

A system which computes text similarity should be able to detect that a strong similarity relationship holds between the two sentences in Example 1.1, as both give a definition for a *jewel*.

Computing text similarity has become increasingly important in recent years as a fundamental means for many natural language processing (NLP) tasks and applications. In automatic essay grading (Attali and Burstein, 2006), for example, text similarity can be used to compare texts by students with a reference answer created by a teacher, in order to automatically assign grades to the student essays. In question answering (Lin and Pantel, 2001), text similarity can be used to compare potential answer candidates from a large pool of potential answers with the question, in order to automatically determine whether it is a suitable answer. For text summarization (Barzilay and Elhadad, 1997), text similarity can be employed to improve the quality of the automatic summaries by reducing redundancy which is determined by measuring text similarity.

Computing text similarity is a very difficult task for machines. This is mainly due to the enormous variability in natural language, in which texts can be constructed using different lexical and syntactic constructions. In Example 1.1, we showed two sentences which basically express both the same thing: a jewel is a precious stone which is used for decoration. While it is simple for a machine to determine that the sentences both mention a *jewel*, as there is an exact word match in both Examples 1.1a and 1.1b, it is rather difficult to infer that *jewellery* matches with the

expression *decorate valuable things that you wear*, as there is no overlap between their lexical representations.

Text similarity received wide-spread attention in recent years. In 2012, the pilot *Semantic Textual Similarity (STS) Task* (Agirre et al., 2012) was established at the *Semantic Evaluation (SemEval)* workshop[1]. The task is intended to unite multiple efforts across the applied semantics community in order to investigate novel approaches to text similarity computation. In Chapter 6, we will discuss this task in detail, and present results of our system which achieved the best overall performance among 35 participating teams.

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 *geometric model* (Shepard, 1962; Widdows, 2004) or the *set-theoretic model* (Tversky, 1977). 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 likely compares bags by 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., 2011b). 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 certain 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 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 1.3)—but may rather be looking for text pairs which are similar with respect to their style and structure. Gärdenfors (2000) introduced a conceptual framework called *conceptual spaces*, which allows to model such *perceived similarity* (Smith and Heise, 1992) which "changes with changes in selective attention to specific perceptual properties." Throughout this work and in particular in Chapter 7, we will elaborate on the idea of *text dimensions* and further discuss suitable dimensions for text similarity tasks.

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

1.1 Text Similarity Measures

A large number of text similarity measures have been proposed in the literature. For the categorization of these measures, we propose a classification scheme which groups the measures according to the type of similarity computation being used into *compositional measures* and *non-compositional measures*. In Chapter 3, we argue that this classification scheme has several advantages over other schemes.

Compositional measures tokenize the input texts, compute pairwise word similarities between all words (i.e. similarity computation is limited to one word pair at a time), and finally aggregate all pairwise scores to an overall score for the text pair. The literature on compositional measures often does not introduce novel measures itself, but rather describes proposals for combining one or more existing word similarity measures with an optional word weighting scheme and a particular aggregation strategy. The literature proposes different sets of instantiations for these steps. 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 take the maximum score, even though more sophisticated strategies exist.

Non-compositional measures, on the other hand, first project the complete input texts onto certain models, e.g. high-dimensional vector spaces or graph structures, before text similarity is then computed based on the abstract representations. For compositional measures discussed above, the two major steps (word similarity computation, aggregation) were modular and fully interchangable. 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 models is required. Existing non-compositional measures range from simple string-based measures which compare two texts without any semantic processing solely based on their string sequences over vector space models such as *Latent Semantic Analysis* (Landauer et al., 1998) and *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007) to measures which project texts onto graph-based representations (Ramage et al., 2009; Yeh et al., 2009).

1.2 Evaluation

The performance of text similarity measures can be evaluated either in an *intrinsic* or *extrinsic evaluation*.

An intrinsic evaluation evaluates the performance of text similarity measures in an isolated setting by comparing the system results with a human gold standard. 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. Popular datasets 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.[2] Evaluation is then typically carried out by computing Pearson or Spearman correlation between the system output and the human judgments. We will discuss details of the intrinsic evaluation in Chapter 6.

An extrinsic evaluation, on the contrary, evaluates the performance of text similarity measures with respect to a particular task at hand, where text similarity is a means for solving a concrete problem. In Chapter 7, we applied our system to the task of *text reuse detection* where text reuse is defined as "the reuse of existing written sources in the creation of new text" (Clough et al., 2002). 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. 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, for example in terms of *accuracy*, i.e. the number of correctly predicted texts divided by the total number of texts, or F_1 *score*. We will discuss details of the extrinsic evaluation in Chapter 7.

1.3 Related Tasks

Text similarity is closely related to a couple of other tasks in the area 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. We elaborate on these related tasks in the following.

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.

[2] 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 Task* (Agirre et al., 2012) at the *Semantic Evaluation (SemEval)* workshop. We will report details in Chapter 6.

Paraphrase recognition A task which is also closely related to text similarity is paraphrase recognition (Dolan et al., 2004). A paraphrase comprises a pair of texts which are "more or less semantically equivalent"[3], 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 for example 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.

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, i.e. typically 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, near-duplicates are sought in very large datasets, for example 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 thus only serves as an indicator for a further plagiarism analysis (Potthast et al., 2012) which specifically adresses external factors, e.g. whether the original text has indeed been reused without authorization or proper citations are absent.

[3]According to the instructions given to the annotators in the study by Dolan et al. (2004). The instructions are part of the archive which is available at http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042

1.4 Outline and Contributions

We now give an overview of the main contributions of this thesis.

- No attempt has been made yet to formalize *in what way* text similarity between two texts can be computed—as opposed to the notion of *similarity* in psychology for which formal models exist. We thus start this thesis with a discussion of the existing formal models of similarity and how we can adapt them to texts. We propose to define text similarity as a notion which can be judged along multiple *text dimensions*, i.e. characteristics inherent to texts, and provide empirical evidence based on a set of annotation studies that humans perceive these dimensions when asked to judge text similarity.

- In previous research, numerous text similarity measures have been presented. Each of these measures exhibits particular inherent properties. However, to the best of our knowledge, no comprehensive survey of the existing measures has been done yet. In Chapter 3, we thus present an overview of existing text similarity measures and propose a classification into *compositional* and *non-compositional* text similarity measures according to their inherent properties.

- Text similarity measures are typically evaluated either *intrinsically* by comparing system outputs with human judgments, or *extrinsically* by trying to solve a particular task such as text classification. In Chapter 4, we introduce the evaluation methodology and present and discuss popular evaluation metrics and datasets previously used in the literature for both types of evaluation.

- Based on the theoretical insights of Chapter 2, we adopt the idea of *conceptual spaces* and propose a composite model which computes text similarity along multiple text dimensions. While traditionally text similarity is computed by using a single text similarity measure at a time, we assume that no single text similarity measure is able to capture all necessary text characteristics.

- We then apply this model for an intrinsic evaluation in the *Semantic Textual Similarity Task* (Agirre et al., 2012) as part of the *Semantic Evaluation (SemEval)* workshop. Using a simple log-linear regression model, our system produces similarity scores which best resemble human judgments across the five given datasets out of 35 participating teams.

- In an extrinsic evaluation, we apply our model to the text classification task of *text reuse detection*. Across three standard evaluation datasets, our model consistently outperforms all prior work. Moreover, the results empirically demonstrate that text reuse can be best detected if text similarity measures are combined across multiple text dimensions.

- We then stress the importance of text similarity for real-world natural language processing applications. We describe our focus application scenario *Self-Organizing Wikis* where we support users of *wikis*, i.e. web-based collaborative content authoring systems (Leuf and Cunningham, 2001), by text similarity features. We thereby focus on two use cases: *duplicate detection* and

link discovery. We also discuss two further applications where text similarity has been found useful: *question answering* and *textual entailment recognition*.

- We describe how our system architecture culminated in a generalized framework for computing text similarity named *DKPro Similarity*.[4] The framework alleviates the traditional fact that text similarity computation is a highly scattered effort. Our open source software package is designed to complement DKPro Core[5], a collection of software components for natural language processing based on Apache UIMA (Ferrucci and Lally, 2004).

- In the final part, we conclude with an analysis of shortcomings of current text similarity research and formulate challenges which should be tackled by future work. In particular, we believe that computing text similarity along multiple text dimensions—which depend on the concrete task at hand—will benefit any other task where text similarity is fundamental.

1.5 Publication Record

The majority of this thesis' contents have previously appeared in peer-reviewed conference or workshop proceedings which are listed below. The chapters which build upon these publications are indicated accordingly.

- 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, Sofia, Bulgaria (Chapters 5, 6, 7, and 9)

- Bär, D., Zesch, T., and Gurevych, I. (2012b). 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 (Chapters 3, 4, 5, and 7)

- 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 (Chapter 8)

- Bär, D., Biemann, C., Gurevych, I., and Zesch, T. (2012a). UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada (Chapters 3, 4, 5, and 6)

- Bär, D., Zesch, T., and Gurevych, I. (2011b). A Reflective View on Text Similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 515–520, Hissar, Bulgaria (Chapters 2 and 4)

[4] http://code.google.com/p/dkpro-similarity-asl
[5] http://code.google.com/p/dkpro-core-asl

- Bär, D., Erbs, N., Zesch, T., and Gurevych, I. (2011a). Wikulu: An Extensible Architecture for Integrating Natural Language Processing Techniques with Wikis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations*, pages 74–79, Portland, OR, USA (Chapter 8)

- Hoffart, J., Bär, D., Zesch, T., and Gurevych, I. (2009a). Discovering Links Using Semantic Relatedness. In *Proceedings of the 8th Workshop of the Initiative of the Evaluation of XML Retrieval*, pages 314–325, Brisbane, Australia (Chapter 8)

Chapter 2

Background and Theory

"A pretender, an impostor, a quack." That's what Goodman (1972) called similarity. To him, similarity is a slippery and useless notion (Decock and Douven, 2010). However, formal models of similarity have been developed in recent years which refute the arguments of a slippery notion and make similarity a well-formalized notion that can be captured in formal models. In the first part of this chapter, we introduce three formal models: the *geometric model* (Shepard, 1962; Widdows, 2004), the *set-theoretic model* (Tversky, 1977), and *conceptual spaces* (Gärdenfors, 2000). In the second part, we discuss that contrary to the notion of *similarity* in cognitive sciences, *text similarity* still lacks a clear definition. We therefore adapt *conceptual spaces* to texts. We propose that *text dimensions*, i.e. characteristics inherent to texts, can be used to judge text similarity, and provide empirical evidence based on a set of annotation studies that humans perceive these dimensions when asked to judge text similarity.

2.1 Formal Models of Similarity

We now introduce three formal models of similarity which have been developed in the cognitive sciences: the *geometric model*, the s*et-theoretic model*, and *conceptual spaces*. For an exhaustive discussion of formal models of similarity, we refer the reader to the work by Decock and Douven (2010).

2.1.1 Geometric Model

The *geometric model* (Shepard, 1962; Widdows, 2004) represents objects in a metric space $\langle X, \delta \rangle$ where each object is represented by a point $x \in X$, and δ is a distance function on X. For two objects a and b, the distance function $\delta(a, b)$ returns a non-negative number which denotes the distance between a and b. Thereby, less distance corresponds to higher similarity. In the geometric space, for all points $a, b, c \in X$ the three metric axioms *minimality*, *symmetry*, and *triangle inequality* hold (Tversky, 1977; Widdows, 2004):

$$\text{Minimality:} \quad \delta(a, b) \geq \delta(a, a) = 0 \tag{2.1}$$

$$\text{Symmetry:} \quad \delta(a, b) = \delta(b, a) \tag{2.2}$$

$$\text{Triangle Inequality:} \quad \delta(a, b) + \delta(b, c) \geq \delta(a, c) \tag{2.3}$$

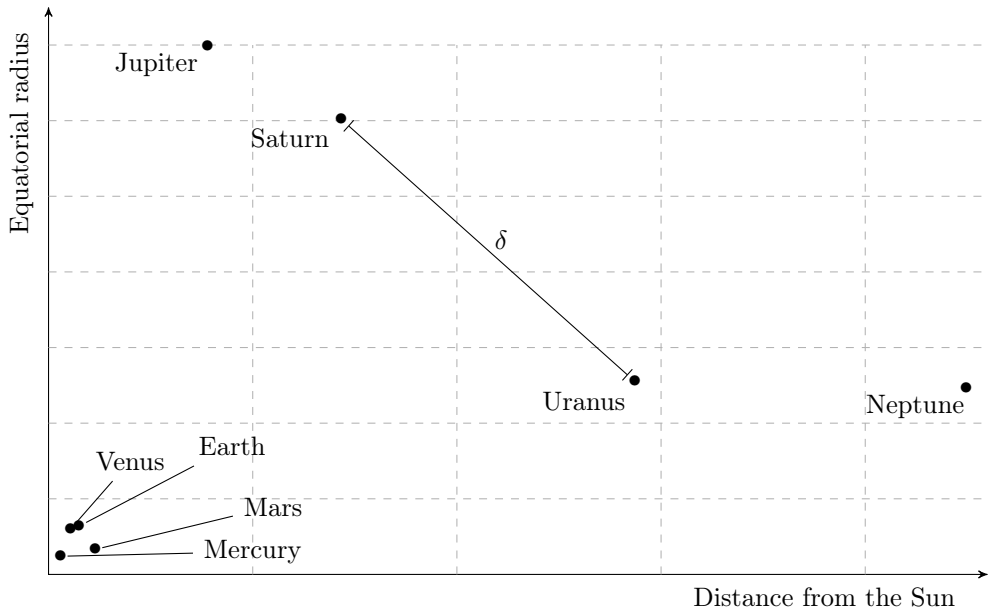


Figure 2.1: Graphical example of the geometric model: The eight planets of the solar system are represented in a two-dimensional space along two dimensions: distance from the Sun, and equatorial radius. The function $\delta(a, b)$ denotes the distance between a and b.

A graphical example is shown in Figure 2.1. Here, we depict the eight planets of the solar system (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune) in a two-dimensional space along two dimensions: distance from the Sun, and equatorial radius.[1] As described above, the distance function δ denotes the distance between two objects, where less distance corresponds to higher similarity. In this example, we can clearly see that the Earth is much more similar to Venus than to Saturn in the given two-dimensional space, as $\delta(\text{Earth}, \text{Venus}) < \delta(\text{Earth}, \text{Saturn})$.

2.1.2 Set-Theoretic Model

The geometric model, however, was critized for being inadequate (Goodman, 1972; Tversky, 1977). Besides the minimality and symmetry axioms, Tversky (1977) discusses the triangle inequality axiom and gives a contradicting example for the similarity of countries: While Jamaica is similar to Cuba (in terms of geographical proximity) and Cuba is similar to Russia (in terms of political affinity), Jamaica and Russia are not similar at all, neither geographically or politically. However, the triangle inequality axiom (see Equation 2.3) claims that $\delta(\text{Jamaica}, \text{Cuba}) + \delta(\text{Cuba}, \text{Russia}) \geq \delta(\text{Jamaica}, \text{Russia})$, i.e. that the sum of the distances between (Jamaica, Cuba) and (Cuba, Russia) is an upper bound for the distance between (Jamaica, Russia). This, however, is contradicted by the given example, as δ(Jamaica, Russia) is obviously much larger. Tversky (1977) hence concludes that "the triangle inequality is hardly compelling." However, we argue that this effect is due to the different assumptions under which the countries have been compared: While *geographical proximity* has been used to justify the high similarity for the pair (Jamaica,

[1] Data taken from http://nssdc.gsfc.nasa.gov/planetary/factsheet and Wikipedia

Table 2.1
Example of the set-theoretic model: We describe the eight planets of the solar system as sets of features rather than points in a metric space. While the values of nominal variables such as *type* or *core* can directly serve as features in the set-theoretic model, the values of cardinal variables such as *density*, *orbital period*, or *number of moons* need to be represented, for example, as a sequence of nested or overlapping sets (Tversky, 1977).

	Mercury	Venus	Earth	Mars
Type	terrestrial	terrestrial	terrestrial	terrestrial
Core	metallic	metallic	metallic	metallic
Density (kg/m^3)	5,427	5,243	5,515	3,933
Orbital Period (days)	88	224	365	686
# Moons	0	0	1	2
	Jupiter	Saturn	Uranus	Neptune
	gas giant	gas giant	gas giant	gas giant
	rocky	rocky	ice	ice
	1,326	687	1,270	1,638
	4,332	10,759	30,799	60,190
	67	62	27	13

Cuba), *political affinity* has been used for (Cuba, Russia). We will continue with the discussion of this effect in Section 2.2.

In consequence, Tversky (1977) proposes a set-theoretic similarity model which accounts for the fact that similarity may not be metric in nature. The proposed model is based on the *features* of the two objects under comparison, whereby a *feature* denotes the value of a binary, nominal, ordinal, or cardinal variable. While the values of binary and nominal variables can directly serve as features in the set-theoretic model, the values of ordinal and cardinal variables need to be represented, for example, as a sequence of nested or overlapping sets (Tversky, 1977). The similarity $sim(a, b)$ is then computed based on a *matching function* f between two feature sets A and B of two objects a and b, respectively:

$$sim(a, b) = f(A \cap B, A - B, B - A) \tag{2.4}$$

Thereby, $A \cap B$ denotes the common features shared by a and b, and $A - B$ and $B - A$ are the distinctive features which belong only to A or B, respectively. The more common features a and b share and the less distinctive features they have, the more similar a and b are.

To illustrate this model, we continue with the example of the solar system. In Table 2.1, we describe the eight planets as sets of features rather than points in a metric space. As example variables, we selected a planet's *type*, *core*, *density*, *orbital period*, and *number of moons*.[1] For example, the planet Venus is represented as Venus $= \{\text{terrestrial}, \text{metallic}, 5\,243, 224, 0\}$.[2] We can see that there is a high similarity between Venus and Earth, as they both are terrestrial planets with a metallic core, have a similar density, and differ only slightly in their orbital period

[2]In the example, we show the raw values of the cardinal variables for the sake of simplicity and for illustration purposes only. In a proper set-theoretic model, the raw values need to be represented, for example, as a sequence of nested or overlapping sets (Tversky, 1977).

and the number of moons. Venus and Jupiter, on the other hand, differ in all five features and are thus not similar at all in the given model.

Unlike the geometric model, the set-theoretic model has the advantage that it allows similarity to be judged in different contexts under different assumptions and objectives. That is, while in the geometric model the metric space is fixed prior to comparing two objects a and b, in the set-theoretic model a and b can be similar in a multitude of ways as only a subset of the features of a and b are considered *relevant* in a given context and hence selected in the comparison process (Decock and Douven, 2010).

2.1.3 Conceptual Spaces

We have learned that the set-theoretic model allows to judge similarity depending upon context, as a "limited list of relevant features" (Tversky, 1977) is compiled in the comparison process. However, Gärdenfors (2000) argues that similarity is indeed geometric in nature, but the traditional geometric model (see Section 2.1.1) needs a refinement. He proposes that its major limitation—the assumption of the existence of a *single* metric space—can be overcome by representing objects in a *multitude* of metric spaces.

Gärdenfors (2000) therefore introduces the idea of a *conceptual space*. A conceptual space is comprised of a number of *quality dimensions* with a geometric structure which are grouped into *domains*. Depending upon context, in a *conceptual space* one or more *domains* are then activated which are used to compare two objects.

A prominent example by Gärdenfors (2000) is that of color perception: He introduces the *color* domain which is comprised by the three quality dimensions *hue*, *chromaticness*, and *brightness*, which correspond to our cognitive representation of colors. For our solar system example, we can thus construct a conceptual space based on the potential domains *orbital characteristics* (potential quality dimensions: *orbital period*, *inclination*, *distance from the sun*), *physical characteristics* (e.g. *equatorial radius*, *mass*), and *atmosphere* (*surface pressure*, *composition*).[3]

The theory of conceptual spaces is a flexible representation framework which is considered to represent human cognition. Similar to the set-theoretic model (see Section 2.1.2), conceptual spaces allow for similarity judgments that depend on a given context by activating a subset of all available domains. In contrast to the set-theoretic model, though, the basic *quality dimensions* are considered geometric in nature. Decock and Douven (2010) thus describe conceptual spaces as a *refined* and *contextualized* geometrical model.

2.2 Adapting Conceptual Spaces to Texts

In the previous section, we learned that the criticism by Goodman (1972) doesn't hold and similarity is a well-formalized notion in the cognitive sciences. While the traditional geometric model (see Section 2.1.1) has its shortcomings, the set-theoretic model as well as conceptual spaces are very well suited to capture similarity.

[3]The proposed domains and corresponding quality dimensions are inspired by the classification of planets in Wikipedia.

Table 2.2
Classification of common natural language processing tasks with respect to the relevant text dimensions suitable for text similarity computation: *content*, *structure*, and *style*

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)	✓		

In this section, we now adapt the idea of conceptual spaces to texts. In the first part, we therefore introduce the notion of *text dimensions* which are inherent to texts. In the second part, we give empirical evidence that humans perceive text similarity along the proposed dimensions.

2.2.1 Dimension Identification

To the best of our knowledge, no attempt has been made yet to formalize *in what way* text similarity between two texts can be computed. We therefore adapt the idea of conceptual spaces, a framework that builds upon the idea of the existence of dimensions (see Section 2.1.3). In order to adapt this model to texts, we need to define explicit dimensions suitable for texts. These dimensions are characteristics inherent to texts that can be used to judge text similarity. We will refer to them as *text dimensions*, henceforth.

We analyzed common NLP tasks with respect to the relevant text dimensions suitable for text similarity computation and present some examples in Table 2.2. We identified three major dimensions inherent to texts: *content*, *structure*, and *style* (Bär et al., 2011b). *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 contents 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 is related to the *style* dimension, as a particular style inherently leads, for

example, to a particular usage pattern of structural elements. The *style* dimension in turn is related to the *content* dimension, as for example particular contents (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.

2.2.2 Empirical Evidence

In Table 2.2, we identified a number of typical NLP tasks with respect to the relevant text dimensions suitable for text similarity computation. We now give empirical evidence for the proposed text dimensions. We report 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., 2011b). 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. Rather, a thorough definition of relevant text dimensions is necessary.

Content vs. Structure Dimension

In this study, we used the *50 Short Texts* dataset by Lee et al. (2005) that contains pairwise human similarity judgments for 1,225 text pairs. We will discuss this dataset in detail in Section 4.1.1. For this study, 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?" Replicating the original annotation setting by Lee et al. (2005), we did not give further instructions to the annotators. The following text pair is an example from the subset that we selected:

(a) *Washington has sharply rebuked Russia over bombings of Georgian villages, warning the raids violated Georgian sovereignty and could worsen tensions between Moscow and Tbilisi. "The United States regrets the loss of life and deplores the violation of Georgia's sovereignty," White House spokesman Ari Fleischer said. Mr Fleischer said US Secretary of State Colin Powell had delivered the same message to his Russian counterpart but that the stern language did not reflect a sign of souring relations between Moscow and Washington.* (2.1)

(b) *Russia defended itself against U.S. criticism of its economic ties with countries like Iraq, saying attempts to mix business and ideology were misguided. "Mixing ideology with economic ties, which was characteristic of the Cold War that Russia and the United States worked to end, is a thing of the past," Russian Foreign Ministry spokesman Boris Malakhov said Saturday, reacting to U.S.*

Defense Secretary Donald Rumsfeld's statement that Moscow's economic relationships with such countries sends a negative signal.

The text pair in Example 2.1 is accompanied with a human similarity score of 3.8 in the original dataset. However, the three annotators who participated in our study rated this pair with scores 3, 5, and 2 on the $1-5$ scale, respectively. Obviously, A_2 had different assumptions than A_1 and A_3 on what makes these texts similar, as she assigned the highest possible similarity score (5) to this text pair.

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 (see Example 2.1). These results indicate that human annotators indeed judge text similarity along both the *content* and the *structure* text dimensions, as introduced above.

We further 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 most likely a different dimension was used to judge similarity, and this further supports our hypothesis that different text dimensions indeed influence the judgements of text similarity.

Content vs. Style Dimension

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 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[4] (SWP). We used the first paragraphs of Wikipedia articles and the full texts of articles in Simple English to obtain pairs of similar length. An example is shown in the following.

SWP. *The constitution of a country is a special type of law that tells how its government is supposed to work. It tells how the country's leaders are to be chosen and how long they get to stay in office, how new laws are made and old laws are to be changed or removed based on law, what kind of people are allowed to vote and what other rights they are guaranteed, and how the constitution can be changed. Limits are put on the Government in how much power they have within the Constitution. On the other hand, countries with repressive or corrupt governments frequently do not stick to their constitutions, or have bad constitutions without giving free-* (2.2)

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

dom to citizens & others. This can be known as dictatorship or simply "bending the rules". A Constitution is often a way of a uniting within a Federation.

WP. *A constitution is a set of laws that a set of people have made and agreed upon for government—often as a written document—that enumerates and limits the powers and functions of a political entity. These rules together make up, i.e. constitute, what the entity is. In the case of countries and autonomous regions of federal countries the term refers specifically to a constitution defining the fundamental political principles, and establishing the structure, procedures, powers and duties, of a government. By limiting the government's own reach, most constitutions guarantee certain rights to the people. The term constitution can be applied to any overall system of law that defines the functioning of a government, including several uncodified historical constitutions that existed before the development of modern codified constitutions.*

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.

2.3 Chapter Summary

In this chapter, we discussed formal models of similarity which have been developed in the cognitive sciences, and showed that the criticism by Goodman (1972) of similarity as a slippery and useless notion doesn't hold. We then adapted 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 *style*. We reported on a number of annotation studies which we conducted in order to empirically demonstrate that humans perceive text similarity along the proposed text dimensions. The results of these studies show that humans 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. We will refer to these theoretical insights in Chapter 5 where we present our composite model.

Chapter 3

Text Similarity Measures

A large number of text similarity measures have been proposed in the literature. For the categorization of these measures, we propose a classification scheme which clusters the measures according to the type of similarity computation being used into *compositional measures* and *non-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. Non-compositional measures, on the other hand, first project the complete input texts onto certain models, e.g. high-dimensional vector spaces, before then comparing them based on the abstract representations. We argue that the proposed classification scheme has several advantages over traditional classifications, which we review in the following.

In the literature, the measures are typically 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. (2010) and Tsatsaronis et al. (2010) extend the above classification scheme by proposing a third class *hybrid measures* which employ resources of both types. Such a classification scheme is well suited to describe the properties of word similarity measures which are applied to certain resources, e.g. the knowledge-based measures that are exhaustively discussed by Zesch and Gurevych (2010). However, we argue that it is not appropriate for text similarity measures, for two main reasons:

Multiple Resources Text similarity measures typically use more than a single resource, as we will see in Section 3.1. For example, many text similarity measures first compute raw similarity scores, e.g. using WordNet (*knowledge-based*), before then weighting these scores, e.g. with *tf-idf* weights (*corpus-based*). It is unclear how to classify such measures according to the *corpus-/ knowledge-based* scheme, as well as which part of the algorithm takes precedence and hence defines the overall class. Using the mixed class *hybrid mea-*

sures also is no feasible alternative, as almost all of the measures use more than a single type of resource, which would render the other two classes superfluous.

Interchangeability The literature on compositional text similarity measures often does not introduce novel measures itself, but rather describes proposals for combining one or more existing word similarity measures with an optional word weighting scheme and a particular aggregation strategy. That way, a text similarity measure could easily be modified by exchanging the underlying word similarity measures or adapting the aggregation strategy. For example, a word similarity measure based on WordNet can easily be replaced by a measure that operates on corpus statistics. We argue that again a classification into *corpus-/knowledge-based* measures is thus not appropriate: Such a class only describes the nature of one single constituent, i.e. the word similarity measure, rather than the whole text similarity measure.

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.

We argue, though, that this classification scheme mixes two orthogonal characteristics of text similarity measures: Corpus-based and hybrid measures describe the characteristics of the underlying word similarity measure, in particular which type of resource they use, as discussed above. Vector-based and descriptive feature-based measures, on the other hand, are instantiations of particular data models that can be used to compute similarity based on any kind of raw scores. For example, the raw pairwise similarity scores computed by a corpus-based measure could be used as the vector elements of a vector-based measure, which then computes the cosine to get the final similarity score. The same holds true for descriptive feature-based measures, where the raw pairwise similarity scores by any corpus-based or hybrid word similarity measure could be used as features in a machine learning classifier, as it was done, for example, by Bär et al. (2012a).

In the following, we elaborate on the proposed classes of *compositional* and *non-compositional measures*, and discuss the corresponding measures.

3.1 Compositional Measures

Figure 3.1 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 3.1, we

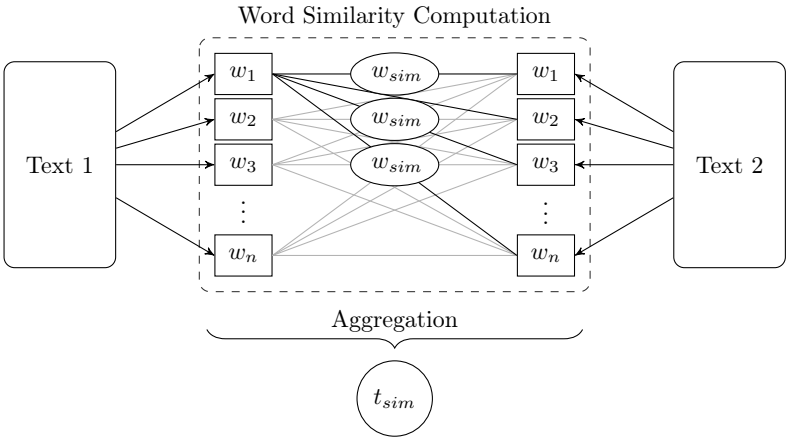


Figure 3.1: High-level depiction of the steps that are performed by compositional measures to compute text similarity between two given input texts: After tokenization, pairwise word similarity scores w_{sim} are computed between all words (i.e. similarity computation is limited to one word pair at a time), before then aggregating all pairwise scores to an overall score t_{sim} for the texts.

summarize the proposals and list the underlying word similarity measures along with a very brief summary of the employed aggregation strategy. Zesch and Gurevych (2010) previously summarized existing word similarity measures. These include, for example, the measures by *Wu and Palmer (1994)*, *Hirst and St. Onge (1998)*, *Leacock and Chodorow (1998)*, *Jiang and Conrath (1997)*, or *Lesk (1986)*. The pairwise word similarity scores can then be aggregated for a text pair by computing the arithmetic mean of all scores or taking the maximum score. However, more sophisticated aggregation strategies exist as well. In the following, we therefore discuss popular compositional text similarity measures with a focus on their underlying word similarity measures and the proposed aggregation strategies.

3.1.1 Mihalcea et al. (2006)

This work proposes to use a single word similarity measure at a time out of a rich set of measures in combination with a bidirectional aggregation strategy. The proposed measures are the *PMI-IR* metric (Turney, 2001), *Latent Semantic Analysis* (Landauer et al., 1998)[1], *Leacock and Chodorow (1998)*, *Lesk (1986)*, *Wu and Palmer (1994)*, *Resnik (1995)*, *Lin (1998a)*, and *Jiang and Conrath (1997)*. In a follow-up work, Mohler and Mihalcea (2009) used two additional measures: the *shortest path* in the WordNet taxonomy (Rada et al., 1989), the measure by *Hirst and St. Onge (1998)*, and *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007)[1].

The proposed aggregation strategy computes a directional similarity score from a text t_1 to a second text t_2 and vice-versa, whereas for each word a counterpart in the other text is sought which maximizes the pairwise similarity. The similarity scores are weighted by a word's *inverse document frequency (idf)* (Salton and McGill,

[1]Both *Latent Semantic Analysis* and *Explicit Semantic Analysis* can operate on single words as well as complete texts. We will discuss them in detail in Section 3.2.

Table 3.1
Compositional measures can be described as instantiations of the two main steps depicted in Figure 3.1: *word similarity computation* and *aggregation*. The specific proposals for compositional measures then differ in the measures and strategies chosen—which are both fully interchangeable.

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 aggregation
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 aggregation
Ho et al. (2010)	Longest Common Subsequence, Yang and Powers (2005)	As above
Islam et al. (2012)	Word trigrams (Web1T)	As above, with 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)

1983) on the British National Corpus (Leech, 1993)[2], then normalized. The final text similarity score is the average of applying this strategy in both directions, as depicted in Equation 3.1.

$$sim(t_1, t_2) = \frac{1}{2}\left(\frac{\sum_{w \in t_1} maxSim(w, t_2) \cdot idf(w)}{\sum_{w \in t_1} idf(w)} + \frac{\sum_{w \in t_2} maxSim(w, t_1) \cdot idf(w)}{\sum_{w \in t_2} idf(w)}\right) \tag{3.1}$$

Thereby, $maxSim(w, t_j) = \arg\max_{w_j \in t_j}(sim(w, w_j))$ denotes the maximum score that can be found for a word w compared with all other words $w_j \in t_j$, $j = 1, 2$.

3.1.2 Li et al. (2006)

The work by Li et al. (2006) uses the *shortest path* (Rada et al., 1989) word similarity measure in conjunction with a *word order* similarity measure. The latter is intended to avoid the shortcomings of traditional bag-of-words models which would compute a perfect similarity between the texts *A quick brown fox jumps over the lazy dog* and *A quick brown dog jumps over the lazy fox*. Two word order vectors are compared by their normalized difference, before the measure then weights all word pairs by their *information content* (Resnik, 1995), which is estimated as the probability of both words co-occurring in the Brown Corpus (Kucera and Francis, 1967). The weighted pairwise word similarity scores then serve as the vector elements of two text vectors, which are compared using cosine similarity to compute the final text similarity score.

[2] A 100 million word corpus containing various texts of current British English, http://www.natcorp.ox.ac.uk

3.1.3 Islam and Inkpen (2008)

This work uses a combination of two word similarity measures: *longest common subsequence* (Allison and Dix, 1986) and *Second Order Co-occurrence PMI (SOC-PMI)* (Islam and Inkpen, 2006) based on corpus statistics from the British National Corpus. The latter is a modification of the original PMI-IR measure (Turney, 2001) which has also been used by Mihalcea et al. (2006). A weighted sum of both measures is computed to determine the pairwise word similarity score. In order to aggregate these scores for the complete texts, the following strategy is applied:

$$sim(t_1, t_2) = \frac{(\delta + \sum_{i=1}^{|\rho|} \rho_i) \cdot (|t_1| + |t_2|)}{2 \cdot |t_1| \cdot |t_2|} \tag{3.2}$$

where t_1, t_2 are the two texts with $|t_1| \leq |t_2|$. The number of words which match perfectly in t_1 and t_2 according to their string sequences is denoted by δ, and ρ is the remaining list of words. For those, word similarity is computed, $\rho_i = maxSim(w_i, t_2)$ for $i = 1, \ldots, |\rho|$, using the *maxSim* notation introduced in Section 3.1.1.

In a follow-up work by Ho et al. (2010), the word similarity measure by Yang and Powers (2005) is used instead of the SOC-PMI metric, while the rest of the original proposal stays unmodified. In another follow-up work by Islam et al. (2012), both the word similarity measure as well as the aggregation strategy are modified. Word similarity is computed using only the *semantic similarity* component, now based on pairwise word similarity from 3-gram frequencies found in the Google Web1T corpus (Brants and Franz, 2006)[3]. In the aggregation step, the *maxSim* function now determines the best-matching word in t_2 by incorporating the mathematical *mean* and *standard deviation* functions across all words in t_2.

3.1.4 Tsatsaronis et al. (2010)

This work relies on a modified *shortest path distance* measure (Rada et al., 1989) that weights the shortest path length in WordNet by *compactness* and *path depth*. *Compactness* is the assessment of frequencies of edge types (e.g. hypernym/hyponym, nominalization, etc.) that constitute the path, and *path depth* is a function to tell whether a path passes through more general or more specific concepts. The aggregation strategy is a directional strategy similar to the one proposed by Mihalcea et al. (2006). It differs in that (i) it weights the pairwise word similarity scores based on the harmonic mean of the two words' *tf-idf* scores rather than only one word's *idf* score, and (ii) it does not normalize the similarity scores, e.g. by the *idf* score as done by Mihalcea et al. (2006).

3.2 Non-Compositional Measures

Figure 3.2 shows a high-level depiction of the steps that are performed by non-compositional measures to compute text similarity between two given input texts.

[3] *Google Web1T* (Brants and Franz, 2006) is a corpus comprising n-grams for $n = 1, \ldots, 5$ which have been created from approximately 1 trillion words found on public web pages.

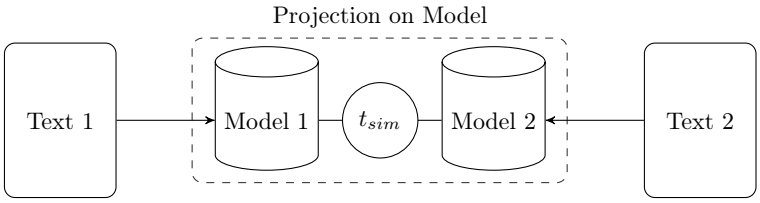


Figure 3.2: High-level depiction of the steps that are performed by non-compositional measures to compute text similarity between two given input texts: After pre-processing, the texts are projected onto certain models such as high-dimensional vector spaces or graph structures, before the overall text similarity score t_{sim} is then computed based on the abstract representations.

After pre-processing, the texts are projected onto certain models such as high-dimensional vector spaces or graph structures, 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 and strategies for these steps. In order to form a new compositional measure, any combination of a new or an existing word similarity measure and an 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 models is required. In Table 3.2, we summarize the existing non-compositional measures which we discuss in the following.

3.2.1 String distance metrics

A basic way to compare two texts is to take their representations at character level and compare them without any semantic processing solely based on their string sequences. One possibility is to compare the texts' *longest common substring* (Gusfield, 1997). Thereby, the length l of the longest contiguous character sequence lcs shared between the two texts t_1 and t_2 is compared with the text length:

$$sim(t_1, t_2) = 1 - \frac{l(t_1) + l(t_2) - 2 \cdot l(lcs(t_1, t_2))}{l(t_1) + l(t_2)} \tag{3.3}$$

However, this measure has limitations, e.g. in cases of word insertions/deletions or typographical errors which break the common substring. The *longest common subsequence* measure (Allison and Dix, 1986) overcomes this by dropping the contiguity requirement. Similarity is then computed according to Equation 3.3, whereby the lcs function now refers to the non-contiguous shared subsequence.

Greedy String Tiling (Wise, 1996) is a method which further allows to deal with shared substrings which do not appear in the same order in both texts. The measure determines the set of shared contiguous substrings, whereby each substring

Table 3.2
Overview of existing non-compositional text similarity measures

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

is a match of maximal length. Similarity is then computed as the number of *marked* characters (i.e. those participating in any shared substring) divided by the text length.

A number of other popular string distance metrics have also been proposed in the past where less distance corresponds to higher similarity (Cohen et al., 2003b). The popular *Jaro distance* (Jaro, 1989) is especially suitable for short strings such as person or place names:

$$sim_{jaro}(t_1, t_2) = \frac{1}{3}\left(\frac{m}{|t_1|} + \frac{m}{|t_2|} + \frac{m-t}{m}\right), \text{ if } m > 0, \text{ and 0 else} \tag{3.4}$$

where m refers to the number of matching characters between t_1 and t_2. A character $a_i \in t_1$ matches a character $b_j \in t_2$ if $a_i = b_j$ and $i - d \leq j \leq i + d$ where $d = \lfloor \frac{max(|t_1|,|t_2|)}{2} \rfloor - 1$. Matching characters which do not appear in the same order in both texts are called *transpositions*, and t is defined as a half of the number of transpositions.

The *Jaro-Winkler distance* (Winkler, 1990) is a variation of the original metric which assigns a higher similarity score to texts with a matching *prefix*, i.e. texts which match from the beginning rather than any position within the string sequence:

$$sim_{winkler}(t_1, t_2) = sim_{jaro}(t_1, t_2) + l \cdot p \cdot (1 - sim_{jaro}(t_1, t_2)) \tag{3.5}$$

where $0 \leq l \leq 4$ is the number of characters in a common prefix for t_1 and t_2. p is a scaling factor for assigning higher weights to a longer common prefix, and is originally set to 0.1 (Winkler, 1990).

The following measures are often referred to as *edit-distance metrics*. Here, the distance between two texts t_1 and t_2 is the minimum number of edit operations that transform t_1 into t_2. The *Levenshtein distance* (Levenshtein, 1966) is a simple metric that assigns uniform costs to all edit operations *insertion*, *deletion*, and *substitution*. The *Monge Elkan distance* (Monge and Elkan, 1997) is an edit-distance metric that uses an affine gap model. The intuition behind this model is that particular sequences of alignments and misalignments between character sequences are

more likely to occur than others. For this metric, edit operations have varying costs: exact matches +5, mismatches −5, and approximate matches 2, which occur if both characters are in one of the following sets: $\{dt\}$, $\{gj\}$, $\{lr\}$, $\{mn\}$, $\{bpv\}$, $\{aeiou\}$, or $\{,.\}$. The costs for starting and continuing a gap are +5 and +1, respectively.

3.2.2 n-gram models

Text similarity measures based on n-gram text representations exist for words and characters. *Character n-gram profiles* (Keselj et al., 2003), as implemented by Barrón-Cedeño et al. (2010), discard all characters (case insensitive) which are not in the alphabet $\Sigma = \{a, \ldots, z, 0, \ldots, 9\}$. All n-grams on character level are then generated and weighted by a *tf-idf* scheme. While in the original implementation only $n = 3$ is used, other values for n may also be considered. Finally, the feature vectors of both string sequences are compared by computing the cosine between them.

A method for comparing texts by means of *word n-grams* has been proposed by Lyon et al. (2001). Two sets of n-grams are generated for both texts, and may then be compared using the Jaccard coefficient, as originally done:

$$sim(t_1, t_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \tag{3.6}$$

or using the containment measure (Broder, 1997):

$$sim(t_1, t_2) = \frac{|T_1 \cap T_2|}{|T_1|} \tag{3.7}$$

where T_i is the set of n-grams for the text t_i, $i = 1, 2$. While in their original work $n = 3$ was proposed, other values for n may also be considered.

3.2.3 Vector Space Model

The vector space model (Salton and McGill, 1983) allows to project texts onto vectors which are used to compute text similarity. The basic model is very general: Within a set of *objects*, each object is defined by a number of *properties*. For two objects t_1, t_2, a shared set of properties is compiled, for which two vectors are constructed. The vector elements thereby are any kind of *weights* (e.g. binary weights indicating presence or absence of a property, or a property's strength of association with an object) that are associated with a particular property in the given text. Finally, the vectors are compared using a vector similarity function such as the cosine:

$$sim(t_1, t_2) = \frac{\sum_{i=1}^{n} t_{1,i} \cdot t_{2,i}}{\sqrt{\sum_{i=1}^{n} (t_{1,i})^2} \cdot \sqrt{\sum_{i=1}^{n} (t_{2,i})^2}} \tag{3.8}$$

For the application to texts, *objects* typically correspond to texts, and *properties* to words. A popular weighting scheme is *tf-idf* or variations thereof. It is often used as a scoring means in information retrieval to match a query with a set of documents (Manning et al., 2008). The vectors are compared by computing the cosine between them. This configuration of the vector space model (tf-idf weights, cosine) is often referred to as *cosine similarity* and is still a strong baseline in such applications.

3.2.4 Latent Semantic Analysis

Landauer et al. (1998) present a technique for representing a text T in a *semantic space* based on corpus statistics. It casts a large representative text corpus D into a term-document matrix. Each matrix element reflects the weight of a term w with respect to a document $d \in D$. The original matrix is then decomposed by Singular Value Decomposition (SVD) into a matrix of reduced rank. Text similarity is then computed by projecting both texts into a vector space (Salton and McGill, 1983) based on the reduced-rank matrix, and by comparing these vectors using cosine similarity.

3.2.5 Explicit Semantic Analysis

Gabrilovich and Markovitch (2007) introduced a method for representing and comparing texts of any length in a high-dimensional vector space. The vector space is constructed based on a given document collection D, where the documents are assumed to describe natural concepts such as *cat* or *dog* (*concept hypothesis*). While Wikipedia was proposed as the collection of choice in the original work, we follow Zesch et al. (2008) and also use two additional collections: Wiktionary[4] and WordNet (Fellbaum, 1998). In the construction phase, a term-document matrix is built using a *tf-idf* weighting scheme of terms with respect to the documents $d \in D$. Each word $w \in t$ is represented by the concept vector $\vec{c}(w)$ of the corresponding row in the term-document matrix, where each vector element denotes the strength of association with a particular document d. For $|t| > 1$ (i.e. texts rather than single words) the vector $\vec{c}(t)$ consists of the sum of the individual word vectors $\vec{c}(w)$ for all $w \in t$. Finally, two concept vectors are compared using cosine similarity.

Contrary to methods based on lexical-semantic resources such as WordNet, ESA hence does not need to operate on expensive expert-created knowledge resources. Virtually any document collection can be used for constructing the vector space, and as Anderka and Stein (2009) showed, not even the concept hypothesis needs to hold true. However, for document collections where the concept hypothesis does hold true, ESA allows to explain similarity between two texts "explicitly" by inspecting the top-ranked elements in the concept vectors. For the words *cat* and *dog*, for example, a strong association with the natural concepts *mammal* and *carnivore* is likely to be found.

3.2.6 Kennedy and Szpakowicz (2008)

This work proposes a text similarity measure based on either WordNet (Fellbaum, 1998) or *Roget's Thesaurus*. The latter is an English thesaurus, first introduced in 1852. Kennedy and Szpakowicz (2008) used the two versions released in 1911[5] and 1987, one version at a time. WordNet groups semantically similar words in *synsets*, and explicitly encodes the relations between them. Roget's Thesaurus, on the other hand, consists of eight *clusters* of thematically related words, where relations are

[4] *Wiktionary* is a collaboratively constructed, web-based multi-lingual dictionary, http://www.wiktionary.org

[5] The 1911 edition of Roget's Thesaurus is available at http://www.gutenberg.org/ebooks/22

only implicit. Each cluster contains a multi-level hierarchy of word sets, where each set further down in the hierarchy refines the parent set. In the 1911 edition, for example, the word *jewellery* is listed in the top level cluster *Words Relating to the Sentiment and Moral Powers*, then in *personal affections* (level 1), *discriminative affections* (level 2), *ornament/jewelry/blemish* (level 3), and so on (Kennedy and Szpakowicz, 2008).

The text similarity measure works as follows: First, all words are mapped onto word sets in the lexical-semantic resource, i.e. synsets in WordNet or the word sets in Roget's Thesaurus, by assigning equal weight $1/n$ to all n sets a word belongs to (for monosemous words, $n = 1$). Each word set then (i) receives the aggregated weights of all word sets further down in the hierarchy, and (ii) is weighted according to the depth in the hierarchy. For two texts, Kennedy and Szpakowicz (2008) transform the weighted concept hierarchies into flat vectors and compare them using cosine similarity.

3.2.7 Ramage et al. (2009); Yeh et al. (2009)

In the proposed graph-based models, texts are represented as graphs $G = (V, E)$ of weighted nodes V and edges E. Nodes correspond to words, edges to relations between the words, and node and edge weights to their importance. Depending on the concrete implementation, relations between two words hold e.g. in case of the existence of a lexical-semantic relation between them such as hypernymy, or if a word co-occurs with the other one above a given frequency threshold. Two measures have been proposed in recent years by Ramage et al. (2009) and Yeh et al. (2009) which employ a similar algorithm, but differ in the lexical-semantic resource they use to construct the graph, as well as in the assignment of the initial node weights. In both works, a modified PageRank algorithm (Brin and Page, 1998) is applied to the text graphs, and random walks are iterated on them until a convergence criterion is met. The final text similarity score is computed based on comparing the converged *semantic signatures* (Ramage et al., 2009), i.e. the stationary distributions, by means of a measure such as cosine similarity or the Jensen-Shannon divergence (Lin, 1991).

For the measure proposed by Ramage et al. (2009), the graph is constructed using WordNet. The nodes are all synsets (e.g. *foot#n#1*), all of the possibly ambiguous part-of-speech tagged words (e.g. *foot#n*), and all untagged words (e.g. *foot*). The connecting edges are all relations found in WordNet, as well as links from POS-tagged words to synsets and untagged words to their POS-tagged instances. The WordNet relations receive uniform weights, all other ones are derived from corpus statistics (Hughes and Ramage, 2007). An initial distribution of *tf-idf* weights (Salton and McGill, 1983) is assigned to each node.

Yeh et al. (2009) use Wikipedia as a lexical-semantic resource instead of WordNet, as they raise concerns about WordNet's limited coverage of world knowledge. In the Wikipedia graph, nodes represent Wikipedia articles, and edges are all links between these articles. For the initial distribution of node weights, the authors propose to use a direct mapping from individual words to Wikipedia articles, as well as Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007), which we described above in detail.

3.3 Chapter Summary

In this chapter, 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. We discussed that a number of other classification schemes have previously been proposed. However, we argued that those have serious limitations and shortcomings, and work well for *word* similarity measures, but fail when it comes to classifying *text* similarity measures.

From an algorithmic point of view, runtime complexity varies across both classes and the corresponding measures. Compositional measures are typically expensive 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 and Explicit Semantic Analysis 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.

Another 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 3.1). 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 Chapters 6 and 7—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.

Chapter 4

Evaluation Methodology

In this chapter, 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.

4.1 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 Chapter 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.

4.1.1 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

Table 4.1
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).

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
Microsoft Paraphrase Corpus (Dolan et al., 2004)	News	5–31 (19)	5,801	binary[1]	2–3
MS Video Desc. Paraphr. Corpus (Chen and Dolan, 2011)	Video descriptions	1–50 (7)	120K	binary[1]	n/a
Wikipedia Rewrite Corpus (Clough and Stevenson, 2011)	Computer Science	36–343 (208)	95	4-way[3]	1
METER Corpus (Gaizauskas et al., 2001)	News (Law & Court, Show Business)	17–1K (205)	1,716	3-way[3]	1
Webis Crowd Paraphrase Corpus (Burrows et al., 2013)	Book excerpts	28–954 (618)	7,859	binary[3]	1

similarity.[1] In Table 4.1, we summarize the statistics for these datasets.[2]

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). An example where the original noun pair *gem/jewel* was replaced by a sentence pair was already given in Example 1.1 on page 1. 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).

We further conducted a re-rating study in order to evaluate whether the human judgments in this dataset are stable across time and subjects. Therefore, we adapted the experiments by Miller and Charles (1991) who showed that human similarity judgments for pairs of words are stable over a long time span (more than 25 years in this case). We collected 10 judgments per pair asking: "How close do these sentences come to meaning the same thing?"[4] Spearman's rank correlation of the

[1] 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. We will report details in Section 6.2.

[2] Table 4.1 contains statistics for both *intrinsic* and *extrinsic* evaluation datasets. We discuss the latter in Section 4.2.1.

[3] We report the class distributions along with the dataset descriptions in Section 4.2.1.

[4] We asked the same question as in the original study by Li et al. (2006). We used Amazon Mechanical Turk (`http://www.mturk.com`) via CrowdFlower (`http://crowdflower.com`).

aggregated results with the original scores is $\rho = 0.91$. From the high correlation, we conclude that the human judgments are indeed stable across time and subjects, and it also indicates that humans indeed share a common understanding on what makes texts *similar*, otherwise such a high correlation could not have been observed. The detailed results of this study can be found in Appendix A.1.1.

In order to better understand the characteristics of this dataset, we performed a study which aimed at investigating human rationales behind similarity judgments. For each text pair we asked the annotators: "Why did people agree that these two sentences are (not) close in meaning?" We collected 10 judgments per pair in the same crowdsourcing setting as described above.

Interestingly, the annotators only used lexical-semantic relations between *words* to justify the similarity relation between *texts*. For example, the text pairs for the original word pairs `tool/implement` and `cemetery/graveyard` were consistently said to be *synonymous*. We conclude that—in this setting—humans rather reduce *text* similarity to *word* similarity. As the text pairs are originally based on term pairs, we further computed the Spearman correlation between the text pair scores and the original term pair scores. The very high correlation of $\rho = 0.94$ supports the results of our annotation study that the annotators indeed judged the similarity between *words* rather than *texts*.

In consequence, we believe that only very limited conclusions can be drawn when using this dataset for the evaluation of text similarity measures. We thus decided not to use this dataset in the intrinsic evaluation.

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)[5] 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 4.1. 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*).

In analogy to the study for the *30 Sentence Pairs* dataset, we performed an annotation study to show whether the encoded judgments are stable across time and subjects. We therefore selected a subset of 50 pairs which have a uniform distribution of judgments across the full similarity range. We asked three human annotators to rate *"How similar are the given texts?"*. The resulting Spearman

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

(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 4.1: 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.

correlation between the aggregated results of the annotators and the original scores is $\rho = 0.88$. This shows that again judgments are stable across time and subjects. The detailed results of this study can be found in Appendix A.1.2.

Even though this dataset has also been widely adopted for the evaluation of text similarity measures, we argue—similarly to the 30 Sentence Pairs dataset—that the evaluation on this dataset only allows to draw limited conlusions.

In this dataset, the distribution of similarity scores is heavily skewed towards low scores. 82% of all text pairs have a text similarity score between 1 and 2 on the 1–5 scale, i.e. are in the lowest similarity range. These text pairs are virtually not similar at all, while the remaining 18% of pairs are spread across the remaining similarity range. We depict the frequency distribution of text pairs across the full similarity range in Figure 4.2.

This limits the kind of conclusions that can be drawn as the number of the pairs in the most interesting class of highly similar pairs is actually very small. If a text similarity measure performs well on this dataset, this rather tells us that it is well able to determine text pairs which are not similar at all, but it does not allow for further conclusions on the most interesting class of highly similar pairs.

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

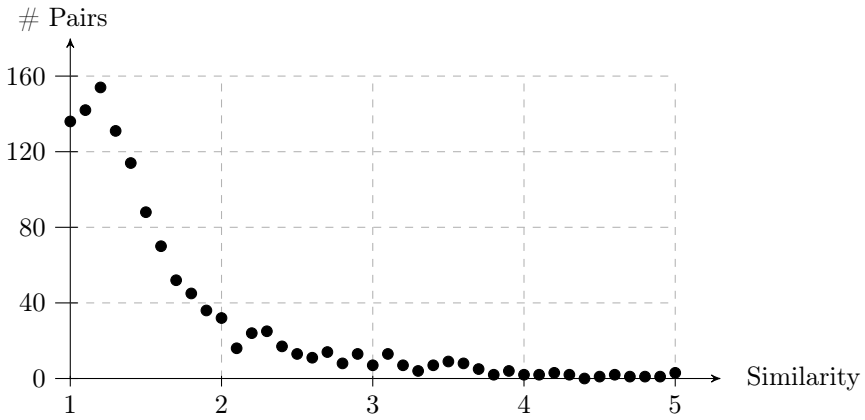


Figure 4.2: Distribution of similarity scores (0.1 intervals) for the *50 Short Texts* dataset.

paraphrase relationship or not (83% inter-annotator agreement)[6]. 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 4.3. For the positive example, the same content is expressed via similar lexical items (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 Chapter 6, we conduct our intrinsic evaluation on this dataset in addition to four other datasets. The revised dataset is freely available.[7]

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 multi-

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

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

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 4.3: 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.

lingual 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 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 Chapter 6. The annotated subset is again freely available.[7]

4.1.2 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*), i.e. in a plotted two-dimensional graph all data

points are on a line which either increases (+1) or decreases (−1). Pearson's r between two score vectors $\vec{x}$ and $\vec{y}$ is computed as follows, with $n = |\vec{x}| = |\vec{y}|$:

$$r = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 (\sum x_i)^2} \sqrt{n \sum y_i^2 (\sum y_i)^2}} \tag{4.1}$$

Linear transformations of the similarity scores do not affect the correlation. For example, a perfect correlation $r = 1$ can be computed 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 $n = |\vec{x}| = |\vec{y}|$:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{4.2}$$

If tied ranks are present, Spearman's ρ is computed as Pearson's r (see Equation 4.1) 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, as only the ranks of the pairs are compared.

4.2 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 thesis, and particularly in Chapter 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 Chapter 7, we present details and results of an extrinsic evaluation which we carried out for the task of *text reuse detection*.

4.2.1 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 Chapter 7 and give an example of a *heavy revision* in Figure 7.1 on page 68.

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[8], and the corresponding paraphrases were acquired in a crowdsourcing process using Amazon Mechanical Turk (Callison-Burch and Dredze, 2010). In the manual filtering process[9] 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.

4.2.2 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 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:

$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn} \tag{4.3}$$

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

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.4}$$

4.3 Chapter Summary

In this chapter, we discussed that the performance of text similarity measures can be evaluated either *intrinsically* or *extrinsically*. For each type of evaluation, we presented popular datasets and evaluation metrics.

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

[9] 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.

For an intrinsic evaluation, popular datasets 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). Evaluation is then carried out by comparing the system results with human judgments, typically by computing Pearson or Spearman correlation. We discussed the shortcomings of the former two datasets and reported on their limitations for the evaluation of text similarity measures. The latter two datsets, however, have been successfully employed in an intrinsic evaluation. We report details on this evaluation in Chapter 6.

For an extrinsic evaluation, popular datasets 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). Evaluation is then carried out by comparing the system output with human classifications, typically by computing *accuracy* and F_1 *score*. We report on our extrinsic evaluation in Chapter 7.

Chapter 5

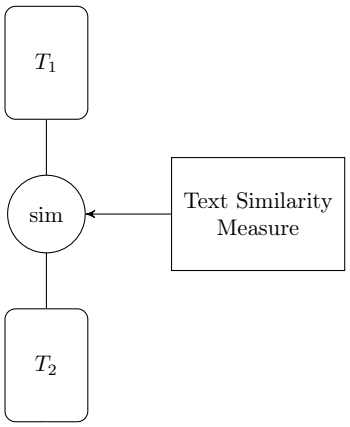
Composite Model

In Section 2.2, we provided empirical evidence that humans perceive text similarity along different dimensions inherent to texts. In particular, we identified three dimensions suitable for texts: *content*, *structure*, and *style*. In this chapter, we now take these theoretical insights as a basis for a composite model for computing text similarity. Depending on the concrete task at hand, we argue that such a model may need to address more than a single text dimension when computing similarity, for example a combination of all three dimensions. In Section 5.1, we discuss our motivation for the proposed composite model. In Section 5.2, we then describe how we combine a multitude of text similarity measures along the proposed text dimensions using machine learning. In Section 5.3, we briefly list the text similarity measures which we used for our experiments. In Section 5.4, we report details on our employed experimental setup. We will follow up on this setup in Sections 6 and 7 for an intrinsic and an extrinsic evaluation.

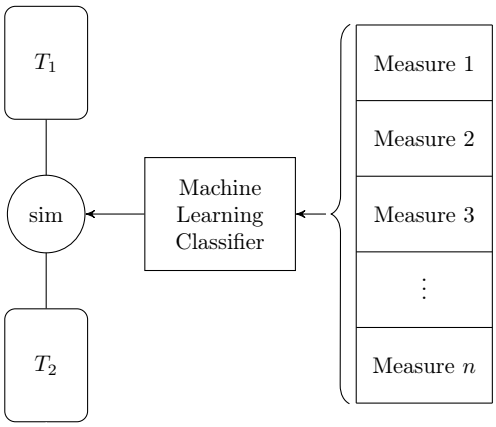
5.1 Motivation

In Chapter 2, we discussed formal models of similarity and how we can adapt them to texts. In particular, we proposed to follow the idea of *conceptual spaces* for text similarity (see Section 2.1.3). In this model, similarity is computed along multiple *dimensions*. In a set of annotation studies, we showed that humans indeed judge similarity along such dimensions when asked to rate the degree of similarity between two texts. In our studies, we identified three major dimensions inherent to texts: *content*, *structure*, and *style*.

Our goal in this chapter is to design a model which implements the idea of text similarity computation along the proposed text dimensions. Traditionally, however, text similarity between two texts is computed by using a single text similarity measure at a time, as depicted in Figure 5.1a, and most of the measures presented in Chapter 3 operate that way. A single measure is thereby assumed to capture all necessary text characteristics, and is assumed to be able to judge text similarity in the same way that humans do. Almost all measures presented in Chapter 3 have in common that they have been proposed in separation of other measures. However, we argue that these measures not necessarily compete with each other whether one is *more suitable* for text similarity computation than the other. Rather, we believe that each measure serves a certain purpose and makes different assumptions about



(a) Traditionally, text similarity between two texts T_1 and T_2 is computed by using a single text similarity measure.



(b) In contrast, we propose to use n similarity measure in parallel and combine them using a machine learning classifier.

Figure 5.1: Traditionally, text similarity (*sim*) between two texts T_1 and T_2 is computed using a single text similarity measure (Figure 5.1a). In contrast, our model is designed to employ multiple text similarity measures in parallel (Figure 5.1b), wich we combine using a machine learning classifier. Thereby, each measure captures particular text characteristics.

what makes texts similar. For example, while string distance measures (see Section 3.2.1) compare two texts based on their string sequences, other measures such as *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007) (see Section 3.2.5) are suitable for going beyond the lexical level and comparing texts based on their semantics. Which one is *more suitable* then depends on the specific task at hand.

However, drawing on the insights of Chapter 2, we argue that no single text similarity measure—if used in separation—is able to compute text similarity in the same way that humans do. As we reported in Chapter 2, humans seem intuitively able to find an appropriate dimension of comparison for given texts. In our opinion, a sophisticated model for computing text similarity thus needs to address a wide variety of text characteristics in order to best resemble human judgments. In the following section, we describe how we combine a multitude of text similarity measures in a single composite model.

5.2 Composing Measures

In the literature, a vast number of text similarity measures has been proposed to date. Our goal is to leverage their enormous potential and combine them in a single composite model. In that process, we do not attempt to judge the quality or suitability of any of the existing text similarity measures if used in separation. Rather, we argue that each measure has its justification and serves a certain purpose when it comes to similarity computation—such as addressing text similarity on a lexical or semantic level.

Instead of using a single measure at a time, we propose to compute text similarity by using multiple text similarity measures in parallel. The only measures presented in Chapter 3 which partly fulfill this requirement and combine a small

number of measures are the compositional measures by Li et al. (2006) (see Section 3.1.2) and Islam and Inkpen (2008) (see Section 3.1.3). In addition to measures which cover semantic aspects of the texts, they take into account *word order* or the *longest common subsequence*. However, we argue that we cannot built upon these approaches for combining a multitude of measures along different text dimensions, as the proposed aggregation strategies do not scale to a large number of individual measures: For example, Islam and Inkpen (2008) use a normalized, weighted sum of two measures. However, the weights need to be fixed prior to computing similarity. In consequence, the measure cannot easily adapt to different sets of text similarity measures (especially sets which contain more than just two individual measures) or different tasks at hand, as it is an open question which weights would be most suitable in that cases.

We thus propose to build a model which is flexible and easily adapts to different sets of text similarity measures. Drawing on the insights of Section 2.2, we propose to use similarity measures from multiple text dimensions—in particular, *content*, *structure*, and *style*. We will report further details on the employed text similarity measures in Section 5.3.

We show our composite model in Figure 5.1b. In contrast to the traditional measures, our model is designed to employ multiple text similarity measures in parallel, wich we combine using a machine learning classifier. The model is thereby assumed to operate in a supervised setting. That way, given training data—that is, text pairs of similar characteristics as the ones under study, accompanied by human similarity judgments—is used to train the classifier. The classifier then *learns* how to best combine the given text similarity measures along the different text dimensions in order to best reproduce the human similarity scores.

By following this approach, we take a different point of view than most previous research in this field (see Chapter 3). We do not propose a single novel text similarity measure in this thesis. Rather, we extensively build upon previous work and design a system architecture which integrates a multitude of existing text similarity measures in a single composite model. That way, we leverage the potential of previous work, but overcome the traditional limitation to a using the measures only in separation. This is in line with recent findings by Fokkens et al. (2013), who argue that the core value of research is to "build upon existing approaches."[1]

5.3 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*.

5.3.1 Content Similarity

The set of text similarity measures that we used for computing similarity along the *content* text dimension comprises measures which we already introduced in Chapter 3. In the following, we briefly list the measures that we used in both the intrinsic

[1] In Chapter 9, we will further elaborate on the the work by Fokkens et al. (2013), who attempt to reproduce previous experimental results and discuss various pitfalls.

evaluation (see Chapter 6) as well as the extrinsic evaluation (see Chapter 7). We will further report in the respective chapters the detailed configuration parameters for all measures that we used in our experiments, as well as additional measures which we used only in one of the evaluations.

From the class of compositional measures (see Section 3.1), we chose to follow the work by Mihalcea et al. (2006) and used the proposed aggregation strategy in combination with three word similarity measures that operate on a given lexical-semantic resource: *Jiang and Conrath (1997)*, *Lin (1998a)*, and *Resnik (1995)*. We used WordNet (Fellbaum, 1998) as the lexical-semantic resource of choice.

From the class of non-compositional measures (see Section 3.2), we first chose a set of string distance metrics, which we introduced in Section 3.2.1. We used the *longest common substring* measure (Gusfield, 1997), the *longest common subsequence* measure (Allison and Dix, 1986), *Greedy String Tiling* (Wise, 1996), and the metrics by Jaro (1989), Winkler (1990), Monge and Elkan (1997), and Levenshtein (1966).

We also compared sets of n-grams, as described in Section 3.2.2. In order to compare *character n-grams*, we followed the implementation by Barrón-Cedeño et al. (2010). We further compare *word n-grams* using the Jaccard coefficient, following Lyon et al. (2001), as well as using the containment measure (Broder, 1997).

We also used the vector space model *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007) as a text similarity measure in our system, which we described in Section 3.2.5. Besides WordNet, we used two additional lexical-semantic resources for the construction of the vector space: Wikipedia and Wiktionary[2].

5.3.2 Structural Similarity

As discussed above, we assume that text similarity can be best computed along multiple text dimensions. In the following, we introduce measures which allow to compute text similarity based on structural aspects inherent to the compared texts.

Stopword n-grams (Stamatatos, 2011) 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) which we described in Section 3.2.2.

For the same reason, we also included *part-of-speech n-grams* of all words in our feature set. We thereby disregard the actual words that appear in two given texts, while taking only the words' part-of-speech tags into account. We compile n-grams based on these tags and consider them indicators for the texts' shallow syntactic structures. We tested n-gram sizes for $n = 2, 3, \ldots, 15$, and compared the two sets using the containment measure (Broder, 1997).

We also employed two similarity measures between pairs of words (Hatzivassiloglou et al., 1999) in order to compare the texts' syntactic structures. We first constructed a set of all shared word pairs which appear in both texts. We then created one feature vector per text pair where each vector element corresponds to the weight of a shared word pair. The weights differ per measure and are determined as follows: The *word pair order* measure assumes that a similar syntactic structure

[2] http://www.wiktionary.org

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 in both texts 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, we compared the feature vectors using Pearson correlation.

5.3.3 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.

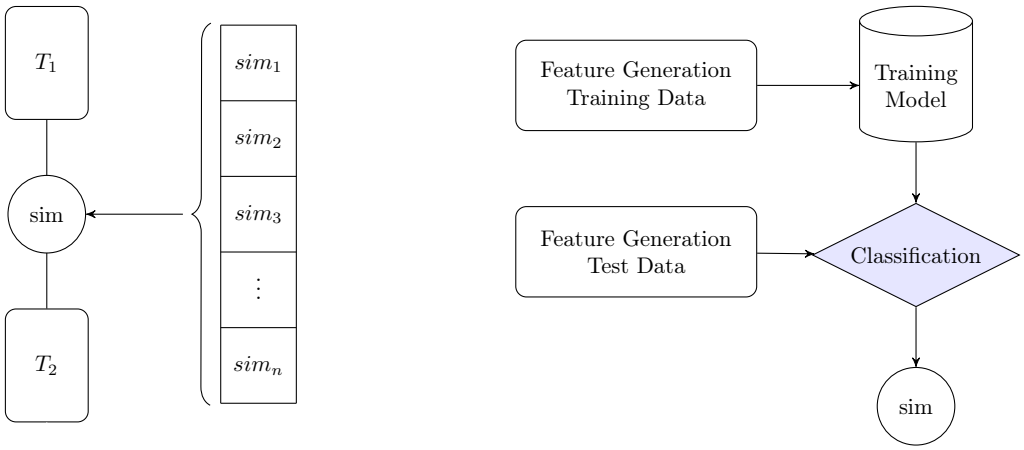
Inspired by Yule (1939) who discussed sentence length as a characteristic of style, we also used two simple measures, *average sentence length* and *average token length*, in our system. These measures compute the average number of tokens per sentence and the average number of characters per token, respectively. 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, we compare texts by their *function word frequencies* (Dinu and Popescu, 2009) which have shown to be good style indicators in authorship attribution studies. Following the original work, this measure uses a set of 70 function words identified by Mosteller and Wallace (1964) and computes feature vectors of their frequencies for each text pair. As originally proposed, the comparison of the vectors is then performed using Pearson correlation.

5.4 Experimental Setup

Our system is based on DKPro Core[3], a collection of software components for natural language processing built upon the Apache UIMA framework (Ferrucci and Lally, 2004). A high-level depiction of our experimental setup is shown in Figure 5.2. The system works as follows: We first run a multitude of text similarity measures separately on all text pairs. We then use the resulting similarity scores as features for

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



(a) In the *feature generation* step, text similarity is computed between two texts T_1 and T_2 with all n available measures, resulting in a vector with n individual text similarity scores $sim_1, sim_2, \ldots, sim_n$.

(b) In the *feature combination* step, a training model is built up from the generated features, which is then applied to the test data in order to predict the text similarity scores for all text pairs in the test data.

Figure 5.2: Our system performs two main steps: First, it pre-computes text similarity scores with all available measures (Figure 5.2a), before then using these scores as features for a machine learning classifier in order to combine them (Figure 5.2b).

a machine learning classifier in order to combine the measures. That way, we follow our assumption 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 then 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 available measures and for all configurations. This results in a large vector of individual text similarity scores for each text pair.

Feature Combination We now use the pre-computed similarity scores and combine them using a machine learning classifier from the Weka toolkit (Hall et al., 2009). Depending on the task at hand, the classifier then predicts either a numerical similarity score for all text pairs based on the given features, or a nominal similarity class. We discuss the effects of different classifiers in the remainder of this section.

Post-processing If necessary, we apply some additional post-processing steps, e.g. limiting the predicted scores to a fixed interval.

The proposed system has the advantage that it allows to combine multiple text similarity measures in a unified classification framework. We assume that each measure computes similarity along particular text characteristics, e.g. along the proposed dimensions *content*, *structure*, or *style*. In combination, our system allows

to take advantage of those manifold similarity scores and that way better learns how to predict text similarity scores similar to human judgments.

Our experimental setups use different machine learning classifiers. For an intrinsic evaluation, it is necessary that the classifier predicts continuous text similarity scores, which can then be compared with human judgments, for example, using Pearson correlation (see Section 4.1.2). In that case, we use a linear regression classifier from the Weka toolkit. An extrinsic evaluation, on the other hand, requires the classifier to predict nominal similarity classes, which can then be compared to human classifications by computing F_1 score (see Section 4.2.2). For example, in Chapter 7, we will report on the task of text reuse detection. In this task, text pairs are classified according to the degree of text reuse for example into *wholly derived*, *partially derived*, or *non-derived* classes. Suitable classifiers for predicting nominal similarity classes and which we used for our experiments are the Naive Bayes classifier or the C4.5 decision tree classifier. We will present details on the evaluation of our system in Chapters 6 and 7.

5.5 Chapter Summary

In this chapter, we presented a composite model for text similarity computation, which we devised based on our theoretical insights of Section 2.2. Our model follows the idea that text similarity can be best computed if multiple text similarity measures are used in parallel, whereby each measure is supposed to capture particular text characteristics. In our experiments, we included measures from the *content*, *structure*, and *style* text dimensions, which we listed in this chapter. Using machine learning techniques, we combined the measures and had the system output numeric similarity scores (suitable for an intrinsic evaluation, see Chapter 6) or nominal similarity classes (suitable for an extrinsic evaluation, see Chapter 7). The proposed system implementation is highly flexible and not limited to the application of the proposed text similarity measures or the employed machine learning classifiers. We will follow up on our system in Chapter 9, where we present *DKPro Similarity*, a generalized framework for text similarity, which resulted from the work presented in this chapter.

Chapter 6

Intrinsic Evaluation

For an intrinsic evaluation, we apply our system to the pilot *Semantic Textual Similarity (STS) Task* (Agirre et al., 2012) at the Semantic Evaluation (SemEval) workshop.[1] In the following, we first introduce the task in detail in Section 6.1 and report on the five evaluation datasets in Section 6.2. In Section 6.3, we then give an overview of the text similarity measures that we used in our system. We report details on the experimental setup along with its configuration parameters in Section 6.4. In Sections 6.5 and 6.6, we discuss the results obtained and present a thorough analysis of the most severe types of errors that occurred. In Section 6.7, we compare our system with the competing systems in this task. We conclude this chapter with a report on follow-up work in Section 6.8 and an overall summary.

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 workshop primarily addressed word sense disambiguation (Yarowsky, 1995) in its early days (and was called *Senseval* back then)[2], 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 texts 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. The text pairs in the employed evaluation datasets are rather short, i.e. typically sentence pairs which may or may not constitute grammatically correct texts. It remains an open question for future research how our results scale to even longer texts, as suitable datasets are currently not available.

[1] http://www.cs.york.ac.uk/semeval-2012/task6

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

Table 6.1
Statistics for the five datasets which have been used for the intrinsic evaluation of text similarity measures in the *Semantic Textual Similarity (STS) Task* (Agirre et al., 2012). All text pairs in these datasets are accompanied by human similarity judgments on a $0-5$ scale where 0 is *not similar at all* and 5 means *completely similar*. Additionally, we compared the ratings of each human annotator with the average ratings of all other annotators. We report it as the average weighted Pearson correlation r across all annotators.[3]

Dataset	Source	Text Type / Domain	Length in Terms (⌀)	# Pairs	Human ⌀r
MSRpar	Microsoft Paraphrase Corpus (Dolan et al., 2004)	News	6–38 (21)	1,500	0.707
MSRvid	Microsoft Video Descr. Paraph. Cor. (Chen and Dolan, 2011)	Video Descriptions	3–25 (8)	1,500	0.876
SMT-eur	ACL Workshops on SMT (Callison-Burch et al., 2007, 2008)	European Parliament Proceedings	3–78 (24)	1,193	0.547
SMT-news	WMT-2007 News Commentary (Callison-Burch et al., 2007)	News conversations	4–36 (13)	399	0.572
ON-WN	OntoNotes (Hovy et al., 2006), WordNet (Fellbaum, 1998)	Glosses	3–41 (9)	750	0.631

6.2 Datasets

The pilot STS task at SemEval-2012 employed five text similarity datasets. Each dataset contains between 399 and 1,500 sentence pairs drawn from publicly available corpora. We summarize the dataset statistics in Table 6.1 and describe the datasets in detail below.

Upon the construction of the datasets for the STS task, human subjects rated all sentence pairs on how similar they are. The original datasets did not contain continuous human judgments on text similarity, but were typically either binary rated or not yet rated at all. In a crowdsourcing setting, the subjects were asked to rate the similarity between the sentences on a discrete scale from 0 to 5. Thereby, explanations were given for each class (Agirre et al., 2012), e.g. "The two sentences are completely equivalent, as they mean the same thing." (5), "The two sentences are mostly equivalent, but some unimportant details differ." (4), or "The two sentences are on different topics." (0). In the end, approx. five annotations per sentence pair were gathered which were then averaged to create the final similarity score.

MSRpar[4] The first corpus is the Microsoft Paraphrase Corpus (Dolan et al., 2004). In Section 4.1.1, we already extensively discussed the nature of this corpus along with its data statistics. In Figure 4.3 on page 34, we already showed an example text pair which is taken from this datasets. For the STS task, a subset of 1,500 sentence pairs was chosen and split 50/50% for training and testing. The pairs were selected following a string-sampling approach: Based on the Levenshtein (1966) distance between the sentences, their similarity score was computed. Pairs were included in the final dataset based on the samples drawn from five equally distributed similarity bands in the range $[0.4, 0.8]$ in order to get a balanced dataset.[5]

[3] Originally, Agirre et al. (2012) did not report any insights on the human annotations.
[4] For consistency, we continue to use the abbreviations as introduced by Agirre et al. (2012).
[5] Randomly pairing texts would result in too many dissimilar pairs (Agirre et al., 2012).

MSRvid The second corpus is the Microsoft Video Description Paraphrase Corpus (Chen and Dolan, 2011), which we also already introduced in detail in Section 4.1.1. In the original dataset, more than 120,000 video descriptions were gathered for short video clips in several languages. However, the descriptions were only assigned to a particular video clip. No sentence pairs were constructed yet. For the STS task, Agirre et al. (2012) carried out the pairing process by selecting only English sentence pairs, and mixing sentences with others from the same video clip, as well as with ones from other clips. Thereby, they applied a similar string sampling technique as describe above, this time sampling pairs from four similarity bands in the range $[0.5, 0.8]$. In the end, 1,500 pairs were selected which were again split 50/50% for training and testing. We show an example below which is rated 1.2 ($0 - 5$ scale).

(a) *A man is playing a guitar.* (6.1)

(b) *A boy is playing a piano.*

SMT-eur The third collection of sentence pairs was drawn from data of the 2007/08 ACL Workshops on Statistical Machine Translation (WMT) (Callison-Burch et al., 2007, 2008). Here, English reference texts from the *Europarl corpus*[6] were paired with the automatic translations (i.e. the system submissions in this exercise) from French to English. In total, 1,193 pairs were selected, which were split into 734 pairs for training and 459 for testing. We show an example text pair below which is rated 5.0 on the $0 - 5$ scale.

(a) *Clearly, explanations are necessary for the increased incidence of BSE, not only in France but also in other Member States.* (6.2)

(b) *It is obviously necessary to determine the reasons for the increase in the incidence of BSE, not only in France but also in other Member States.*

SMT-news Finally, two *surprise* datasets were provided for testing the final system submissions, for which no training data had been provided. The first one is the *SMT-news* dataset. It contains 399 pairs of English reference texts along with their automatic translations (i.e. the system submissions in this exercise, similar to *SMT-eur*) from French to English, where the data was drawn from the WMT 2007 news commentary test set (Callison-Burch et al., 2007). We show a not necessarily grammatically correct example below which was rated 4.75 on the $0 - 5$ scale.

(a) *The old version of the European response–what psychologists might call "dollar envy"–will only become more acute.* (6.3)

(b) *The old version of the European response (what psychologists call "the envy of the dollar") only become more pervasive.*

ON-WN The second *surprise* dataset is the *ON-WN* dataset which contains 750 pairs of glosses from OntoNotes (Hovy et al., 2006) and WordNet (Fellbaum, 1998)

[6]The *Europarl corpus* comprises parallel text from the proceedings of the European Parliament (Koehn, 2005) and is available for download: `http://www.statmt.org/europarl`

senses. Here, the sampling procedure was to pair 50% "equivalent senses" and 50% unrelated senses using a binary-weighted cosine similarity metric on the tokenized sentences. We show an example below which was rated 4.75 on the $0 - 5$ scale.

(a) *bring back to life, return from the dead* (6.4)

(b) *cause to become alive again*

Unfortunately, Agirre et al. (2012) do not report any insights on the human annotations for any of the five datasets. In particular, they do not report how well the ratings of a single annotator correlate with the average ratings of all other annotators. Through personal communication, however, we were able to receive the raw crowdsourced annotation data and inspect it ourselves. We compared the ratings of each human annotator with the average ratings of all others by means of Pearson correlation, and in Table 6.1 report the final score as the weighted average of all correlations. Thereby, we weighted each annotator's correlation according to the number of pairs that were judged. This procedure is similar to the one used in follow-up work by Agirre et al. (2013), which we will dicuss later in Section 6.8. Other potential inter-rater agreement metrics such as Fleiss' (1971) κ are inappropriate, as the human similarity ratings are continuous scores rather than distinct annotation classes.

We observed the highest average correlation (0.876) for the MSRvid dataset. We argue that this is due to the fact that the sentences describe real-world situations which can be rather easily compared by humans, e.g. *"A man is playing a guitar."* (see Example 6.1). The lowest average correlation (0.547) was observed for the SMT-eur dataset. As described above, this dataset comprises texts that are taken from the proceedings of the European Parliament. We argue that these texts are potentially harder to read and understand than the very clear sentences of the MSRvid dataset. Hence, a lower agreement for the similarity scores between the different annotators is to be expected.

Nonetheless, we further argue that the reported overall scores of the human agreements are only to be considered a rough estimate. Even though we conducted the same procedure as Agirre et al. (2013) did in their follow-up work (see Section 6.8), we'd like to point out a major shortcoming of that process: The raw crowdsourced annotation data that was available to us contained the text pairs along with all human ratings that were collected for each of them. We argue that for a proper evaluation how well a single annotator performs compared to the rest of the annotators, *all* annotators need to rate *all* text pairs that are contained in the datasets. However, in the data that we had available, some annotators rated only very few text pairs, while others rated a large number of pairs. In consequence, the reported overall score is the result of computing the weighted arithmetic mean of the correlations which have been observed on different sets of text pairs.

6.3 Text Similarity Measures

For the intrinsic evaluation, we employ a set of text similarity measures which follows the description in Section 5.3 with a minor set of modifications: While we used the

set of *structural* and *stylistic* similarity measures as originally described, we slightly modified the set of *content* similarity measures to better fit this task.

In Table 6.2, we list the measures that we already described in Chapter 5.3 and that we used in the intrinsic evaluation. We also describe the configuration parameters as necessary. In the following, we report on all measures which differ from the basic setup described in Section 5.3.

In our experiments, we also used an additional measure based on a *distributional thesaurus*. The idea here is that similarity can 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 also tried a new approach which is inspired by *query expansion* techniques in information retrieval (Xu and Croft, 1996). Query expansion tries to overcome the problem of potential word mismatches (*lexical gaps*) between a *query* and the *documents* by *expanding* the original query words. Word mismatches in information retrieval typically occur when one user formulates a query using particular words, but a second user has authored a document which uses synonymous words rather than the query words. We believe that a similar technique can also benefit text similarity computation in the intrinsic evaluation: As the two texts within each pair in the evaluation datasets (see Section 6.2) are typically not authored by the same person, lexical gaps may easily occur. Thus, instead of applying even more text similarity measures, we decided to stick with the set of existing similarity measures, but modify the input texts, so that lexical gaps are closed *before* computing similarity. We therefore used two 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 3.1.

For lexical substitution, we used the system that operates on the *Turk Bootstrap Word Sense Inventory (TWSI) 2.0* (Biemann, 2012a,b). The TWSI resource is a word sense inventory which provides substitutions for a set of 1,012 frequent English nouns with high precision. The resource comes with a system for supervised word sense disambiguation which disambiguates polysemous nouns using a machine learning classifier from the Weka toolkit (Hall et al., 2009). For each noun for that substitutions exist[7], we added all the substitutions to the text—in addition to the original noun—and computed pairwise word similarity for the texts as described above. That way, we alleviate potential word mismatches 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

[7]The TWSI 2.0 resource is available for download at http://www.ukp.tu-darmstadt.de/data/lexical-resources/twsi-lexical-substitutions

Table 6.2
Overview of the text similarity measures that we already listed in Section 5.3 and which have been used in the intrinsic evaluation, along with their configurations.

Text Similarity Measure	Configurations
Compositional Measures	
Mihalcea et al. (2006)	Resnik (1995), Jiang and Conrath (1997), and Lin (1998a) 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, $n = 1, \ldots, 15$

comprise the concatenation of the original texts and all three round-trip 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.

We will discuss the effects of both text expansion mechanisms as part of Section 6.5.2, and provide examples and further insights there.

6.4 Experimental Setup

The system we used for the intrinsic evaluation is an adaptation of the system originally introduced in Chapter 5. If follows the idea that text similarity can be best detected if a composition of measures is used. In the following, we describe the system configuration that we used for the experiments throughout this chapter.

Pre-processing no modifications; see Section 5.4

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

Post-processing In the training phase, the linear regression classifier learns a linear combination of all given text similarity scores. Due to the learned coefficients of the linear combination, the predicted text similarity scores in the test phase may then fall outside the required interval $[0, 5]$. In order to avoid this, we convert all scores outside the interval to the minimum/maximum value, respectively. That way, we ensure that text pairs that receive an enormously

Table 6.3
Feature sets of the two final system configurations for the intrinsic evaluation

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 Translation for Word Similarity	

low/high similarity score outside the interval limits are classified as *not similar at all/perfectly similar* and hence receive the minimum/maximum score.

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 then equal each other on the character level, 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 equal each other under the assumption of a simplified vocabulary. For example, text pairs which equal each other in all characters but some additional whitespaces or punctuation marks will still receive the maximum similarity score.

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:

Run 1 In a manual feature selection process (which we detail in Section 6.5.1), we identified the features which achieved the best performance on the training data. We list these features in Table 6.3. 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 6.3) to the feature set of Run 1.

6.5 Results & Discussion

We now report and discuss the results obtained in the Semantic Textual Similarity Task. In Section 6.5.1, we first report on the feature selection process, where we identified an optimal combination of text similarity measures using the available training data. In Section 6.5.2, we then report and discuss the final results obtained on the test data.

6.5.1 Feature Selection

In order to find an optimal feature combination of text similarity measures, 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 6.4, 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 well as Run 1 across all three training datasets, even though the differences are not statistically significant.[8] Run 2 further outperforms (statistically significant)[8] 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 (see Section 6.2), as the original texts 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 how the data was originally paired: Pairs were only formed 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.

6.5.2 Final Results

In the final evaluation on the test data, three evaluation metrics were used (Agirre et al., 2012):[9] The *ALL* metric computes Pearson correlation of the union of the system outputs across all five datasets with human judgments. The *ALLnrm* metric first fits the system output to the gold standard using least squares, then applies the

[8] 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)

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

Table 6.4
Best results in terms of 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

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[11]	.564	.835	.527
	Character n-gram profiles	.658	.771	.554
	Explicit Semantic Analysis	.427	.781	.619
	Word n-grams	.474	.782	.619
	String Similarity[11]	.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[11]	.168	.225	.325
	Function Word Frequencies	.179	.142	.189

ALL metric.[10] *Mean* refers to the weighted mean across the Pearson correlations of all datasets, where the weight corresponds to the number of text pairs in each dataset.

In Table 6.5, we report the offical results achieved on the test data, and visualize the results in Figure 6.1. 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, while it was ranked #87 and #85 for the *ALL* and *ALLnrm* metric, respectively. 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* ($r = .823$)[12] and *Mean*

[10]The *ALL* metric penalizes systems which do not perform equally well across all five datasets. The *ALLnrm* metric, on the other hand, can assign higher overall scores to systems which perform well on some of the datasets, but not on others. 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.

[11]In order to better demonstrate the performance of different classes of text similarity measures, we used a combination of measures with a single machine learning classifier and report the combined results. We grouped the measures as follows: *Pairwise Word Similarity*: aggregation by Mihalcea et al. (2006) in combination with the word similarity measures by Resnik (1995), Jiang and Conrath (1997), and Lin (1998a) on WordNet (see Section 3.1.1); *String Similarity*: all measures described in Section 3.2.1; *Statistical Properties*: *type-token ratio*, *sequential TTR*, *average sentence length*, *average token length*, *sentence length ratio* and *token length ratio* (see Section 5.3.3).

[12]95% confidence interval (Fisher Z-value transformation): $0.811 \leq r \leq 0.834$

Table 6.5
Official results on the test data for the top 6 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. 6.5.2). The provided cosine similarity baseline is shown at the bottom of this table. The 95% confidence interval[12] for the ALL metric comprises the top four systems.

$\#_1$	$\#_2$	$\#_3$	System	r_1	r_2	r_3	MSR-par	MSR-vid	SMT-eur	On-WN	SMT-news
1	2	**1**	Bär et al. (2012a) (Run 2)	**.823**	.857	**.677**	.683	.873	.528	.664	.493
2	3	5	Sarić et al. (2012)	.813	.856	.660	.698	.862	.361	.704	.468
3	**1**	2	Sarić et al. (2012)	.813	**.863**	.675	**.734**	**.880**	.477	.679	.398
4	4	4	Bär et al. (2012a) (Run 1)	.811	.855	.670	.682	.870	.511	.664	.467
5	6	13	Banea et al. (2012)	.784	.844	.616	.535	.875	.420	.671	.403
6	27	7	Heilman and Madnani (2012)	.783	.808	.639	.639	.720	.485	.712	.531
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
87	85	70	Baseline	.311	.673	.435	.433	.299	.454	.586	.390

($r = .677$), and #2 for *ALLnrm* ($r = .857$). The results of both configurations are within the 95% confidence interval for the *ALL* metric.[12]

Comparison of Runs 1 and 2

In the following, we analyze the differences between the configurations of Runs 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:

(6.5)
(a) *Putin's Russia has already lost 12 leading journalists to murder in the past six years.*
(b) *Putin's Russia has already lost 12 prominent journalists, murdered in the last six years.*

The above Example 6.5 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 texts are transformed into the following (not necessarily grammatically correct) fragments using statistical machine translation (as described in Section 6.3, using Spanish as a bridge language):

(6.6)
(a) *The Russia of Putin has already lost 12 outstanding journalists of murder in the last six years.*
(b) *The Russia of Putin has already lost 12 outstanding journalists, murdered in the last six years.*

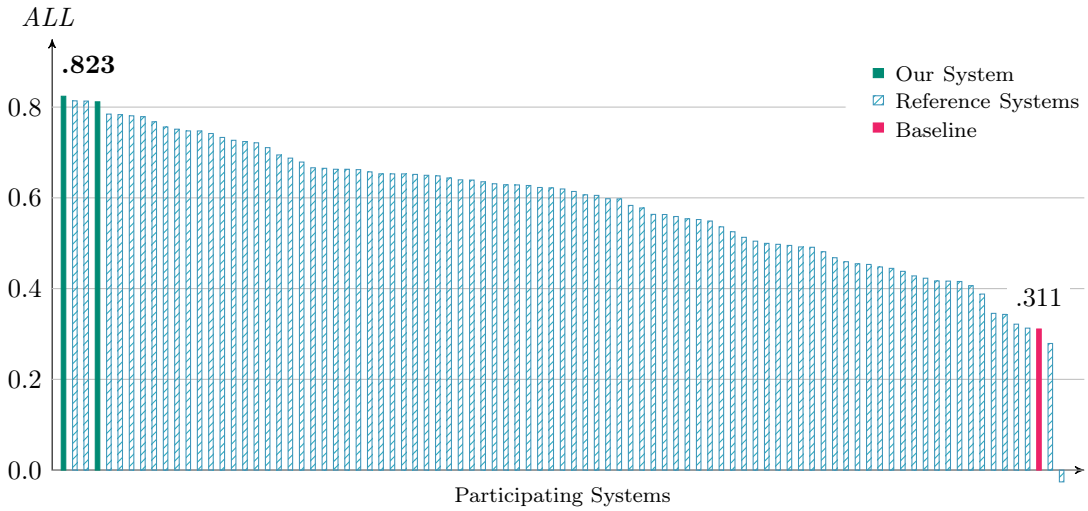


Figure 6.1: Official results on the test data for all 89 participating systems in terms of the *ALL* metric. Our two system configurations are ranked #1 and #4 (show in opaque green), while the baseline is ranked #87 (opaque red).

In the translation process, the mismatching words *leading* and *prominent* were transformed into *outstanding* in both texts (Example 6.6). That way, the lexical gap was closed and in consequence an improved similarity score (4.80) was computed, which is very close to 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:

(a) *Western Europeans, who have been spared this legacy, should heed our warnings.* (6.7)

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

The above Example 6.7 also 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 6.3, this system provides substitutions for a set of 1,012 frequent English nouns. The substitutions for the words *legacy* and *history* are listed below:

(a) *legacy:* – (6.8)

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

For the Example 6.8, no substitutions were found for the word *legacy*. For *history* in the second text, eight substitutions (see above) were found. As described in Section 6.3, 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.

6.6 Error Analysis

We now present insights on the most severe types of errors that occurred across the five evaluation datasets. In order to classify the errors, we manually inspected the system output of our best system configuration (Run 2) and compared it with the given gold standard scores assigned by human subjects (for details on the annotation procedure, see Section 6.2). In the following, we discuss the error classes that we identified and present examples from the datasets along with the similarity scores produced by our system as well as those assigned by human subjects.[13]

Spelling errors/nonsensical input texts

Several cases exist where the input texts are either nonsensical, i.e. the meaning of the texts is fully unclear, or contain spelling errors. This is particularly true for the datasets MSRvid, SMT-eur, and SMT-news, while the spelling errors most prominently occur for the MSRvid dataset. We attribute the errors to the nature of the datasets: The MSRvid dataset contains descriptions of short video clips that were gathered using crowdsourcing (see Section 6.2). In that process, spelling errors could easily have happened. The text pairs in the SMT-eur and SMT-news datasets, on the other hand, resulted from automatic translations for given input texts (see Section 6.2). That way, nonsensical input texts provided to the original SMT system naturally resulted in nonsensical paired texts as a result of the automatic translation process. In contrast, such effects cannot be found for the other two datasets MSRpar and ON-WN. Those datasets have been created based on (presumably spelling-corrected) online news sources (MSRpar), or glosses in expert-created wordnets (ON-WN) (for details, see Section 6.2).

We illustrate some texts with spelling errors in Examples 6.9 to 6.11 and highlight the errors in boldface. In these cases, the text similarity score computed by our system is substantially lower than the human assigned score. Even though it is easy for humans to determine the correct meaning of *potatos* (correct plural form is *potatoes*), *kangroo* (should be *kangaroo*), or *muybarak* (should be *Mubarak*), machines cannot easily do so. This is due to the fact that erroneously spelled words either cannot be found in the employed lexical-semantic resource such as WordNet in the similarity computation process, or—in cases of named entities—just cannot be matched with the second text using e.g. string distance metrics (see Section 3.2.1).

(6.9)
(a) *A man cuts up* ***potatos****.*
(b) *A man slices potatoes.*

(source dataset: MSRvid; system output: 1.54; human score: 4.80)

[13]The texts are taken from the original datasets; they may be grammatically incorrect and may contain erroneous punctuation.

(a) *A **kangroo** is eating something.* (6.10)

(b) *A kangaroo is eating.*

(source dataset: MSRvid; system output: 2.53; human score: 4.80)

(a) *Only a month ago, Mubarak dismissed demands for constitutional reform as "futile."* (6.11)

(b) *Only a month before, **muybarak** refused the demands of constitutional reform by taxing them with "futile."*

(source dataset: SMT-news; system output: 3.11; human score: 5.00)

In the following Examples 6.12 to 6.14, we illustrate nonsensical text pairs. For these cases, it is unclear what constitutes text similarity for the very short texts such as *Tunisia/Tunisia was*. While our system produced reasonable similarity estimates for these pairs, e.g. the maximum score 5.0 for this pair, the human scores are not clear. Example 6.14 also illustrates a case where two headlines of a report were to be compared. Similarly, we argue that this is an invalid text pair for text similarity evaluation, as it is unclear in what way these texts are to be compared.

(a) *A Europe for All* (6.12)

(b) *All Europe*

(source dataset: SMT-news; system output: 2.79; human score: 4.66)

(a) *Tunisia* (6.13)

(b) *Tunisia was*

(source dataset: SMT-eur; system output: 5.00; human score: 3.75)

(a) *Van Orden Report (A5-0241/2000)* (6.14)

(b) *Relation Horse-Trailer Orden (A5-0241/2000)*

(source dataset: SMT-eur; system output: 4.63; human score: 3.00)

Missing/non-resolvable contextual references

This error class comprises all errors that presumably happened due to contextual references, e.g. person or date/time references, that were either missing in one text, or not resolvable from the given text fragments. These errors exclusively occurred for the MSRpar dataset. We list three such erroneously classified text pairs in Examples 6.15 to 6.17 and highlight the contextual references in boldface.

In Example 6.15, it is unclear whether the expression *afternoon trading* can be matched with *trading Wednesday* in the second text, as no temporal point of reference is given. In Example 6.16, the phrase *believes the latter is true* cannot be resolved in the given context: It is unclear what the acting subject *Marcel Desailly* actually refers to. In Example 6.17, one may assume that the pronoun *he* in the

first sentence refers to the person named *Young* in the second text. However, this is not clear from the given context.

We argue that for a proper evaluation dataset, these artifacts should be resolved. An improved dataset would leave less room for speculation and subjective interpretations, and hence allow for a more precise evaluation.

(6.15)
(a) *Gillette shares rose $1.45, or 4.5 percent, to $33.95 in* ***afternoon*** *New York Stock Exchange* ***trading****.*
(b) *Shares of Gillette closed down 45 cents at $33.70 in* ***trading Wednesday*** *on the New York Stock Exchange.*

(source dataset: MSRpar; system output: 3.49; human score: 1.50)

(6.16)
(a) *The Chelsea defender Marcel Desailly has been the latest to speak out.*
(b) *Marcel Desailly, the France captain and Chelsea defender,* ***believes the latter is true****.*

(source dataset: MSRpar; system output: 4.36; human score: 2.40)

(6.17)
(a) ***He*** *also could be barred permanently from the securities industry.*
(b) ***Young*** *faces a fine, suspension or being permanently barred from the securities industry.*

(source dataset: MSRpar; system output: 3.63; human score: 1.80)

Unclear objectives

In the MSRpar as well as the MSRvid datasets, we found several cases where it is unclear how to judge similarity, as it is not well defined what makes the given texts *similar* or not. We list three cases in Examples 6.18 to 6.20.

In Example 6.18, it is unclear why the average human rating is very low (1.30). In our opinion, the given texts are rather similar, as both texts discuss the rise or fall of a given stock market index. However, the addressed indexes and the amount of increase or decrease vary. In Example 6.19, two texts are given which describe that *a person* is doing something. However, in the first sentence the person is *cutting [a] meat*, while in the second sentence the person is *riding a mechanical bull*. The average human similarity rating for this text pair is the minimum score of 0 on the $0 - 5$ scale, i.e. all humans unanimously assigned the lowest possible score. However, other interpretations seem also possible, e.g. by stressing the fact that both sentences mention *a person* who is doing something. Our system estimated a medium similarity score of 2.26, which seems—under this assumption—perfectly reasonable. In Example 6.20, both texts are equal except the fact that in one text *three* men are playing chess, while in the second text only *two* people are doing so. Intuitively, we believe that our system output (4.63) is in the correct order of magnitude, i.e. in the very high similarity range between 4 and 5, compared to the average human score which is in the medium range only (2.60).

In all these cases, more clearly defined guidelines should be provided which need to state explicitly what makes two texts similar or not. In consequence, we believe that both human annotators as well as text similarity measures will greatly benefit from such a more clear-cut definition of text similarity.

(a) *The technology-laced Nasdaq Composite Index .IXIC inched down 1 point, or 0.11 percent, to 1,650.* (6.18)

(b) *The broad Standard & Poor's 500 Index .SPX inched up 3 points, or 0.32 percent, to 970.*

(source dataset: MSRpar; system output: 3.43; human score: 1.30)

(a) *A person is cutting a meat.* (6.19)

(b) *A person riding a mechanical bull*

(source dataset: MSRvid; system output: 2.26; human score: 0.00)

(a) *Three men are playing chess.* (6.20)

(b) *Two men are playing chess.*

(source dataset: MSRvid; system output: 4.63; human score: 2.60)

Multiword expressions

A fourth class of errors comprises multiword expressions, which we illustrate in Examples 6.21 to 6.26. In all of these cases, humans rated the similarity of the text pairs very high (between 4 and 5 on the $0-5$ scale). However, our system was not able to capture the meaning of such expressions appropriately, and hence assigned substantially lower similarity scores to the given text pairs.

In Example 6.21, both words *restrict* and *confine* should presumably match with the expression *place limits on* in the second text. However, our system failed to do so and assigned only a very low similarity score of 0.78. In Example 6.22, the expressions *nonacceptance* and *refusing to accept* are expected to match. Again, our system failed and assigned only a medium similarity score, as some of the remaining text is similar. Examples 6.23 to 6.26 follow the same pattern: Humans assigned very high scores, while our system failed to determine the correct meaning of the multiword expressions and hence assigned substantially lower scores.

In our opinion, being able to compute similarity between multiword expressions is crucial for a robust text similarity system. While multiword expressions typically appear as isolated texts in the ON-WN dataset, they appear as part of longer texts in the SMT-news dataset. We strongly believe that many real-world texts exhibit similar characteristics, as such texts may be authored by different people or at different points in time, which in consequence may lead to variability in vocabulary usage. A robust text similarity system needs to be able to infer the correct meaning

of particular expressions even if the meaning is spread across multiple words.

(a) ***restrict*** *or* ***confine*** (6.21)

(b) ***place limits on*** *(extent or access).*

(source dataset: ON-WN; system output: 0.78; human score: 4.75)

(a) *a written message of* ***nonacceptance*** (6.22)

(b) *a message* ***refusing to accept*** *something that is offered.*

(source dataset: ON-WN; system output: 2.55; human score: 5.00)

(a) *the act of officially* ***gaining entrance*** *to somewhere* (6.23)

(b) *the act of* ***admitting someone to enter****.*

(source dataset: ON-WN; system output: 1.93; human score: 4.00)

(a) *Being a Muslim and being an Islamist are* ***not the same thing****.* (6.24)

(b) *To be Moslem and be Islamiste are* ***two different things****.*

(source dataset: SMT-news; system output: 2.84; human score: 5.00)

(a) *This tendency* ***extends deeper*** *than headscarves.* (6.25)

(b) *This trend* ***goes well beyond*** *simple headscarves.*

(source dataset: SMT-news; system output: 2.61; human score: 4.50)

(a) *None of this* ***absolves*** *rich countries* ***of their responsibility*** *to help.* (6.26)

(b) *But that does not* ***detract*** *rich countries* ***from their obligation*** *to help.*

(source dataset: SMT-news; system output: 3.15; human score: 5.00)

Discussion

We demonstrated that the most severe errors in the intrinsic evaluation resulted from one of four classes: *spelling errors/nonsensical input texts*, *missing/non-resolvable contextual references*, *unclear objectives*, and *multiword expressions*. Overall, however, as we have shown with the discussion of the annotation quality in Table 6.1 on page 48, not even humans perfectly agree on *how similar* two texts in a pair are. To date, there is still much speculation and subjective interpretation involved in the process of judging text similarity.

We argue that for future studies it needs to be better definied *in what way* texts are to be considered *similar*. In particular, it needs to be clearly stated what both human annotators and automated measures should focus on when judging similarity.

That way, (a) human subjects can be better trained and instructed on this task, (b) text similarity measures can be tailored more precisely to the type of similarity at hand. Additionally, if multiword expressions are then captured more precisely in a future system, we hypothesize that text similarity can be even better computed with automated measures, and in consequence the overall system performance will most likely also improve.

6.7 Competing Systems

It is particularly interesting to note that all of the top 3 teams (Bär et al., 2012a; Sarić et al., 2012; Banea et al., 2012)—which submitted the systems that performed best in the exercise and were ranked #1 to #5 in the final results according to the *ALL* metric (see Table 6.5)—used a machine learning classifier to combine different sets of text similarity features. While we used a linear regression classifier from the Weka toolkit (Hall et al., 2009) as described earlier (see Section 6.4), both Sarić et al. (2012) and Banea et al. (2012) combined the features using a support vector regression model from either the LIBSVM package (Chang and Lin, 2011) or Weka, respectively. In our opinion, this again supports our hypothesis that text similarity can best computed if multiple measures are used in parallel, i.e. similarity is computed along multiple text dimensions.

The features that were used by the system by Sarić et al. (2012) comprise different variations of n-gram overlap measures, Latent Semantic Analysis (Landauer et al., 1998) (see Section 3.2.4), as well as syntactic features based on dependency trees. Banea et al. (2012) use the aggregation strategy by Mihalcea et al. (2006) in combination with a set of word similarity measures (see Section 3.1.1), non-compositional measures such as Latent Semantic Analysis and Explicit Semantic Analysis (see Section 3.2.5), and also syntactic features based on dependendy graphs. However, contrary to these two systems, we not only evaluated the overall performance of our system, but precisely evaluated the performance of the individual classes of measures on the three given training datasets (see Table 6.4). That way, we were able to determine a set of text similarity measures which contains only a rather small number of general measures (see Table 6.3).[14] In consequence, we avoided overfitting the features of the machine learning classifier on the training data, which made our system able to perform most robust across the five datasets.

Besides the top systems, the next best system by Heilman and Madnani (2012) uses a text similarity measure which is based on a machine translation evaluation technique. Similar to edit-distance measures (see Section 3.2.1), the proposed measure is based on the following idea: the less operations it takes to transform one text into the other, the higher their similarity. Operations thereby are not only based on the character level of the texts (e.g. shift, insertion, deletion), but also comprise semantic operations such as synonymy replacements or paraphrasing text fragments. As we are not able to summarize all remaining systems in detail, we refer the interested reader to the task overview by Agirre et al. (2012). There, all tools and resources used by the different systems are briefly summarized. Further details

[14] Sarić et al. (2012), for example, use *number* and *stock index symbol* overlap measures which are highly tailored towards the particular data at hand and will most likely not generalize well.

Table 6.6
Official results on the 2013 test data for the STS Task. In total, there were 88 participating runs in addition to the cosine baseline and the two best-performing systems from 2012. We report the Pearson correlation on the four test datasets *HDL*, *OnWN*, *FNWN*, and *SMT* (Agirre et al., 2013) along the single offical evaluation metric *Mean* (see Section 6.5.2).

Rank	System	Mean	HDL	OnWN	FNWN	SMT
1	Han et al. (2013)	**.618**	.764	.752	**.581**	.380
2	Han et al. (2013)	.592	.742	.705	.544	.370
3	*deft-baseline*[15]	.579	.653	**.843**	.508	.326
⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	Bär et al. (2012a)	.565	.734	.734	.340	.325
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	Sarić et al. (2012)	.522	.655	.633	.405	.338
⋮	⋮	⋮	⋮	⋮	⋮	⋮
76	Baseline	.363	.539	.282	.214	.286

are reported in the respective system descriptions.

Following the idea of the top-performing systems, we assume it would be interesting in future work to inspect the performance of a combined regression model on all features of the top systems. We assume that the better these features capture different text characteristics, the better the combined model performs. 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 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 text similarity computation 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. Follow-up research in this direction has already partly been carried out in the context of the second *Semantic Textual Similarity Task* in 2013, which we discuss in the following section.

6.8 Follow-up Work

In 2013, the second *Semantic Textual Similarity (STS) Task* (Agirre et al., 2013) was held as a follow-up exercise to the pilot task that we reported on throughout this chapter. This year, the STS Task was held as a shared task at the *Second Joint Conference on Lexical and Computational Semantics*.[16] This time, we did not participate directly in the exercises, but provided our experimental setup (see Section 6.4) as a strong open source baseline which was also officially recommended

[15] Unfortunately, it is unclear who are the authors of this system submission.
[16] http://ixa2.si.ehu.es/sts

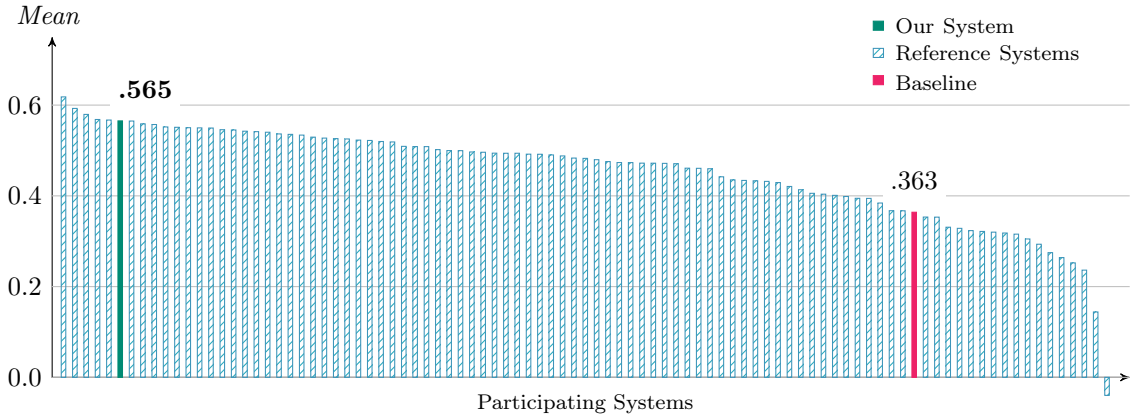


Figure 6.2: Official results on the 2013 test data for all participating systems in terms of the *Mean* metric. Our system from last year is reported as a strong baseline and ranked #6 (show in opaque green). The cosine similarity baseline is ranked #76 (opaque red).[18]

to all task participants.[17] The intention thereby was that we wanted to support participants of this year's exercises in exploring our system, in particular with respect to the features and the combination strategy used, and invite them to built upon our last year's best performing system, e.g. by introducing novel features or using a different machine learning classifier. That way, we also hope to lower to entrance barrier for new participants to this task by providing them an out-of-the-box solution for getting started with developing systems for text similarity computation.

In this year's task, a number of participants reported that they were inspired by our experimental setup and developed a system which reuses several of our proposed text similarity features (Buscaldi et al., 2013; Severyn et al., 2013; Zhu and Lan, 2013). Two participants further reported that they directly built upon our proposed experimental setup, either as a first step for the generation of text similarity features (Marsi et al., 2013) or as a basis for the whole text similarity system including the *feature combination* and *evaluation* components (Wu et al., 2013). This demonstrates that our previously proposed experimental setup is both highly competitive with respect to the text similarity features used, as well as thoroughly engineered so that new users can quickly adopt our system and customize it in various ways.

While the 2012 data (see Section 6.2) was provided to train the systems, new test data was given to the participants to evaluate their final system configurations. The new data comprises sentence pairs which are taken from news headlines (*HDL* in Table 6.6), machine translation evaluation exercises (*SMT*), and glosses (*OnWN* and *FNWN*). In total, 34 teams (compared to 35 in 2012) submitted 89 systems (compared to 88 in 2012, up to 3 per participant) in these exercises. Additionally, the task organizers provided the same cosine similarity baseline as last year. This year, the task organizers further officially report the results of the two best-performing systems from 2012 (Bär et al., 2012a; Sarić et al., 2012) on the new 2013 data.

We report the official results (Agirre et al., 2013) in Table 6.6 and visualize them

[17] http://www-nlp.stanford.edu/wiki/STS

[18] In the offical results (Agirre et al., 2012), the cosine similarity baseline is reported on rank #73. While the detailed results for both our system as well as the two system configurations by Sarić et al. (2012) are reported as well, they are not listed directly as part of the ranked results.

in Figure 6.2. This year, evaluation was only carried out in terms of the *Mean* metric (see Section 6.5.2) which refers to the weighted mean across the Pearson correlations of all datasets, where the weight corresponds to the number of text pairs in each dataset. The results show that the top-performing system (Han et al., 2013) outperforms the publicly available systems by Bär et al. (2012a) and Sarić et al. (2012). It is to be noted that the publicly available version[19] of our experimental setup is slightly different from the version that performed best in the SemEval-2012 workshop: It does not use a distributional thesaurus and only a single text expansion mechanism (lexical substitution only, no statistical machine translation), as the corresponding code and the accompanying resources cannot be readily made available to the public. Nonetheless, it is particularly interesting to note that the system by Sarić et al. (2012) is ranked only #27 on this year's data, while it was ranked #2 in 2012. Obviously, this system is highly sensitive to both the training and test data. In contrast, our baseline system is ranked #6 on this year's data. As Agirre et al. (2013) put it, our system (Bär et al., 2012a) "performed extremely well when trained on all available training [data], with no special tweaking for each dataset". This is in line with our findings reported in Section 6.5 that our system does not necessarily show the best results on any of the single datasets, but it is most robust across different data.

6.9 Chapter Summary

In this chapter, we presented an intrinsic evaluation of our system in the context of the pilot *Semantic Textual Similarity (STS) Task* at the *Semantic Evaluation* workshop. We showed that our system performed best in two out of the three official evaluation metrics. While we did not reach the highest scores on any of the single datasets (see Table 6.5), our system was most robust across different data.

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 build a machine learning classifier which learns combinations of the text similarity scores very well 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 the key to systems that will generalize even better.

[19] http://code.google.com/p/dkpro-similarity-asl/wiki/SemEval2013

Chapter 7

Extrinsic Evaluation

In this chapter, we conduct an extrinisic evaluation of our system on the task of *text reuse detection*. We first describe the task in Section 7.1, and then report details of our experimental setup and the features used in Section 7.2. We discuss the evaluation results on three standard datasets: the *Wikipedia Rewrite Corpus*, the *METER Corpus*, and the *Webis Crowd Paraphrase Corpus*, which we already introduced in Section 4.2.1. Finally, we 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 Chapter 3). However, as discussed in Chapter 5, the existing similarity measures typically exhibit a major limitation: They compute similarity only on the features that describe the *content* dimension of the given texts. Thus, they so far ignore any other text characteristics, e.g. the structural and stylistic text dimensions.

Figure 7.1 shows an example of text reuse taken from the Wikipedia Rewrite Corpus (Clough and Stevenson, 2011) (see Section 4.2.1 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

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 7.1: 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 source text was split into two individual sentences and the order of the reused parts was 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.[1] In order to use such features as indicators of text reuse, we propose to further include measures of *stylistic similarity*.

7.2 Text Similarity Measures

For this task, we employ a set of text similarity measures which follows the description in Section 5.3 with a minor set of modifications: While we used the set of *structural* and *stylistic* similarity measures as originally described, we slightly modified the set of *content* similarity measures to better fit the task of text reuse detection. In Table 7.1, we list the measures we used in the extrinsic evaluation and describe the configuration parameters as necessary. In the following, we report on all measures which differ from the basic setup described in Section 5.3.

This time, we additionally used *Latent Semantic Analysis* (Landauer et al., 1998) for computing text similarity, where the construction of the semantic space was done using the evaluation corpora (see Section 7.4). Details on this measure can be found in Section 3.2.4.

For assessing *word n-gram similarity*, we employed the same measures as originally described in Section 5.3. This time, we additionally use the original system name *Ferret* (Lyon et al., 2004) to refer to the variant with $n = 3$ using the Jaccard coefficient.

7.3 Experimental Setup

The system we used for the extrinsic evaluation in the context of text reuse detection again builds upon the composite model introduced in Chapter 5, and is a slight modification of the system which we already used for an intrinsic evaluation in Chapter 6. It again follows the idea that text similarity can be best detected if a composition of measures is used. We refer the reader to Section 5.4 for an elaborate

[1] The type-token ratio (Templin, 1957) of the texts is .79 and .71, respectively.

Table 7.1
Compositional and *non-compositional* text similarity measures introduced in Chapter 3, along with their configurations, which we used in the extrinsic evaluation

Text Similarity Measure	Configurations
Compositional Measures	
Mihalcea et al. (2006)	Resnik (1995), Jiang and Conrath (1997), and Lin (1998a) on WordNet
Non-compositional Measures	
Character n-gram profiles	$n = 2, \ldots, 15$
Explicit Semantic Analysis	Wikipedia, Wiktionary
Greedy String Tiling	
Latent Semantic Analysis	Semantic space on the evaluation corpora
Longest common subsequence	2 normalizations
Longest common substring	
Word n-grams	Jaccard/Containment, $n = 1, \ldots, 15$ *Ferret*: $n = 3$ using Jaccard coefficient

discussion on the technical aspects of the system. The system we used for the experiments throughout this chapter modifies the original experimental setup as follows:

Pre-processing no modifications; see Section 5.4

Feature Generation no modifications; see Section 5.4

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 Chapter 6 for the intrinsic evaluation which predicts numerical scores for each text pair, we now combine the features using 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 7.1.

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 within the *content*, *structure*, and *style* dimensions individually. That way, we were able to determine the performance of multiple measures within a single text dimension.

Table 7.2
Performance of selected similarity measures on the Wikipedia Rewrite Corpus, the METER Corpus, and the Webis Crowd Paraphrase Corpus, grouped by similarity dimension

Text Similarity Feature	WP Rewrite		METER		Webis CPC	
	Acc.	$\bar{F}_1$	Acc.	$\bar{F}_1$	Acc.	$\bar{F}_1$
Majority Class Baseline	.400	.143	.715	.417	.517	.341
Ferret Baseline	.642	.517	.684	.535	.794	.789
Content Similarity						
Character 5-gram Profiles	.642	.537	.715	.417	.753	.742
ESA (Wikipedia)	.474	.323	.711	.484	.760	.753
Greedy String Tiling	.558	.457	.755	.645	.805	.800
Longest Common Substring	.621	.524	.719	.467	.743	.736
Resnik Word Similarity	.632	.500	.715	.417	.666	.656
Word 2-grams Containment	.747	.683	.727	.692	.801	.797
Structural Similarity						
Word Pair Distance	.611	.489	.715	.417	.775	.767
Word Pair Ordering	.642	.494	.715	.417	.785	.780
POS 3-grams Containment	.642	.554	.731	.701	.787	.783
Stopword 3-grams	.632	.515	.715	.417	.778	.776
Stopword 7-grams	.653	.527	.652	.482	.753	.750
Stylistic Similarity						
Function Word Frequencies	.453	.296	.715	.417	.727	.719
Sequential TTR	.400	.220	.715	.417	.667	.638
Sentence Ratio	.389	.268	.755	.625	.657	.653
Token Ratio	.432	.222	.755	.619	.778	.774
Type-Token Ratio	.379	.197	.715	.417	.723	.712

Run 3 In the third configuration, we combined the measures across text dimensions in order 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.

7.4 Results & Discussion

In this section, we report the results obtained on the evaluation datasets, thereby demonstrating the effectiveness of the proposed composite model for 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). We already described these datasets in detail in Section 4.2.1, and summarized their properties in Table 4.1 on page 30.

Evaluation was carried out in terms of accuracy and $\bar{F}_1$ score. By accuracy,

Table 7.3
Results for the best classification on the Wikipedia Rewrite Corpus for the original 4-way classification

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.400	.143
Ferret Baseline	.642	.517
Chong et al. (2010)[3]	*.705*	*.641*
Clough and Stevenson (2011)		
- our re-implementation[4]	.726	.658
- *as reported in their work*	*.800*	*.757*
Our System	**.842**	**.811**

we refer to the number of correctly predicted texts divided by the total number of texts. As we reported in Section 4.2.1, the class distributions in the datasets are skewed.[2] We hence 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). That way, systems that perform well on only some of the classes but perform poorly on others are penalized when computing the overall $\bar{F}_1$ score. A good performance across all classes is necessary to achieve a good overall performance.

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

In the Wikipedia Rewrite Corpus, text reuse occurs between a source text and each of the short answers given by human subjects. According to the degree of rewrite, the dataset is 4-way classified as *cut & paste*, *light revision*, *heavy revision*, and *no plagiarism*. We reported further details on this dataset in Section 4.2.1.

We summarize the results on this dataset in Table 7.3.[5] 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.[4]

[2] We reported the class distributions along with the dataset descriptions in Section 4.2.1

[3] 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.

[4] 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).

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

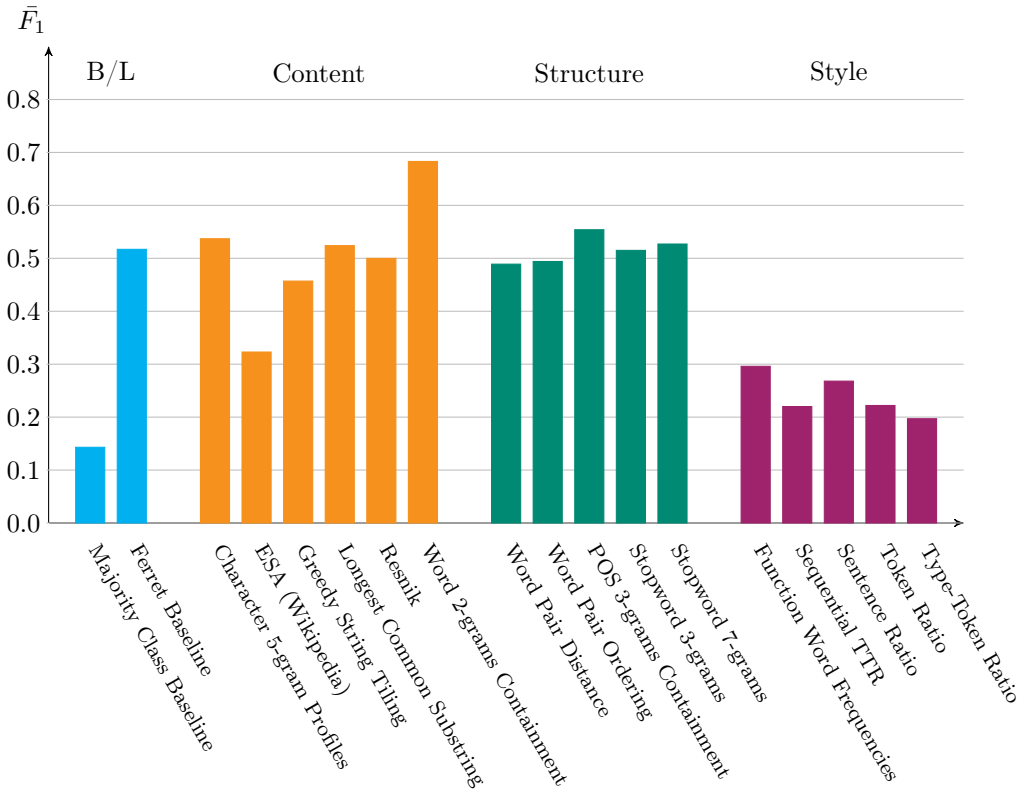


Figure 7.2: Detailed results for a selected set of individual text similarity measures, grouped by similarity dimension, on the Wikipedia Rewrite Corpus

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 chapter. We report our findings in Sections 7.4.2 and 7.4.3.

In Table 7.2, we further report the detailed results for a selected set of individual text similarity measures, grouped by similarity dimension.[6] A graphical overview of these results is shown in Figure 7.2. Due to space limitations, we only report a selected set of best-performing measures per dimension and compare them with the baselines: While the majority class baseline performs very poorly on this dataset ($\bar{F}_1 = .143$), the *Ferret* baseline achieves $\bar{F}_1 = .517$. Some content similarity measures such as *word* 2*-grams containment* show a reasonable performance ($\bar{F}_1 = .683$), while structural measures cannot exceed $\bar{F}_1 = .554$, and stylistic measures perform only slightly better than the majority class baseline ($\bar{F}_1 = .296$).

In Table 7.4, we report the best results for the combinations of text similarity measures within and across dimensions, and visualize them in Figure 7.3. 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*

[6]Table 7.2 also lists the detailed results for the METER Corpus and the Webis Crowd Paraphrase Corpus. We will discuss these results in the corresponding Sections 7.4.2 and 7.4.3.

Table 7.4
Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Wikipedia Rewrite Corpus for the original 4-way classification

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.747	.693
Structure	.716	.660
Style	.442	.398

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations across dimensions		
Content + Style	.800	.757
Content + Structure	**.842**	**.811**
Structure + Style	.632	.569
Content + Structure + Style	.832	.798

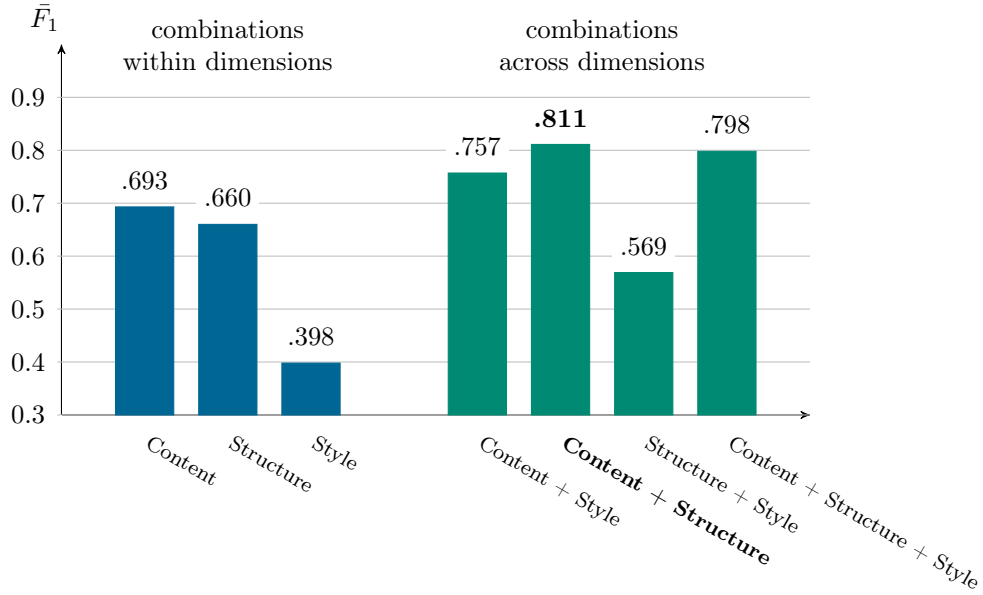


Figure 7.3: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Wikipedia Rewrite Corpus

and *structure*. This supports our hypothesis that computing text similarity indeed benefits from dimensions other than *content*. The effects of dimension combination held true regardless of the classifier used, even though the decision tree classifier performed consistently better than Naive Bayes.

Error Analysis We present the confusion matrix for our best configuration in Table 7.5. In total, 15 texts out of 95 have been classified with the wrong label. While all texts except a single one in the class *no plagiarism* have been classified correctly, 67% of errors (10 texts) are due to misclassifications in the *light* and *heavy revision* classes. We assume that these errors are due to questionable gold standard annotations as the annotation guidelines for these two classes are highly similar (Clough and Stevenson, 2011). For the *light revision* class, the annotators "could alter the text in some basic ways", thereby "altering the grammatical structure (i.e. paraphrasing)." Likewise, for the *heavy revision* class, the annotation manual expected the annotators to "rephrase the text to generate an answer with the same meaning as the source text, but expressed using different words and structure."

Table 7.5
Confusion matrix (expected class vs. classification result) for the best classification on the Wikipedia Rewrite Corpus for the original 4-way classification

exp. \ class.	cut&paste	light rev.	heavy rev.	no plag.
cut&paste	**15**	1	1	2
light rev.	3	**13**	3	0
heavy rev.	2	2	**15**	0
no plag.	0	0	1	**37**

Table 7.6
Results on the Wikipedia Rewrite Corpus for the folded 3-way classification

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.400	.190
Ferret Baseline	.768	.745
Clough and Stevenson (2011)[7]	.821	.788
Our System	**.884**	**.859**

Table 7.7
Confusion matrix on the Wikipedia Rewrite Corpus for the folded 3-way classification

exp. \ class.	cut&paste	potential	no plag.
cut&paste	**14**	3	2
potential	5	**33**	0
no plag.	0	1	**37**

As each text of this dataset was written by only a single person for a given rewrite category, we decided to conduct an annotation study, in which we were mostly interested in the inter-rater agreement of the subjects. We asked 3 participants to rate the degree of text reuse and provided them with the original annotation guidelines. We used a generalization of Scott's (1955) π-measure for calculating a chance-corrected inter-rater agreement for multiple raters, which is known as Fleiss' (1971) κ and Carletta's (1996) K.[8] In summary, the results[9] of our study support our hypothesis that the annotators mostly disagree for the *light* and *heavy revision* classes, with fair[10] agreements of $\kappa = .34$ and $\kappa = .28$, respectively. For the *cut & paste* and *no plagiarism* classes, we observe moderate[10] agreements, $\kappa = .53$ and $\kappa = .56$, respectively. The detailed annotations can be found in Appendix A.2.1.

Based on these insights, we decided to fold the *light* and *heavy revision* classes

[7] 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.

[8] An exhaustive discussion of inter-rater agreement measures is given by Artstein and Poesio (2008).

[9] The detailed results are reported in Appendix A.2.1.

[10] Strength of agreement for κ values according to Landis and Koch (1977)

Table 7.8
Results on the Wikipedia Rewrite Corpus for the folded binary classification

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.600	.375
Ferret Baseline	.937	.935
Clough and Stevenson (2011)		
- our re-implementation	.958	.957
- *as reported*	*.947*	*n/a*
Our System	**.968**	**.967**

Table 7.9
Confusion matrix on the Wikipedia Rewrite Corpus for the folded binary classification

exp. \ class.	plagiarism	no plag.
plagiarism	**55**	2
no plag.	1	**37**

into a single class *potential plagiarism*. This approach was also briefly discussed by Clough and Stevenson (2011), though not carried out in their work. We report the results and the confusion matrix in Tables 7.6 and 7.7. As the classification task gets easier through the reduction to three classes, the results for the Ferret baseline improve, from $\bar{F}_1 = .517$ to $\bar{F}_1 = .745$. The re-implementation of the system by Clough and Stevenson (2011) achieves $\bar{F}_1 = .788$. Our system again outperforms all other systems with $\bar{F}_1 = .859$.

We hypothesize that fine-grained distinctions are not always necessary—depending on the particular task at hand. We thus decided to go one step further and fold all potential cases of text reuse. This variant of the dataset results in a binary classification of plagiarized/non-plagiarized texts. We present the results and the corresponding confusion matrix in Tables 7.8 and 7.9. In this simplified setting, even the Ferret baseline achieves an excellent performance of $\bar{F}_1 = .935$. Our system still slightly outperforms ($\bar{F}_1 = .967$) the re-implementation of the system by Clough and Stevenson (2011).

An interesting observation across all three variants of the dataset is that the same three texts always constitute severe error instances where e.g. a *cut & paste* text is falsely labeled as *no plagiarism*, which is more severe than mislabeling a *light revision* as a *heavy revision*. Two of the three cases account for the texts which describe the PageRank algorithm. One of these instances was falsely labeled as *cut & paste* while it is non-plagiarized, and the other one vice-versa. In the envisioned semi-supervised setting, the remaining less severe error instances, where e.g. a *light revision* was classified as a *heavy revision*, can be reviewed by a user of the system. We suppose it is even hard for users to draw a strict line between possibly reused and non-reused texts[11], as this heavily depends on external effects such as user intentions and the task at hand.

[11] As also indicated by the only fair inter-annotator agreements discussed above.

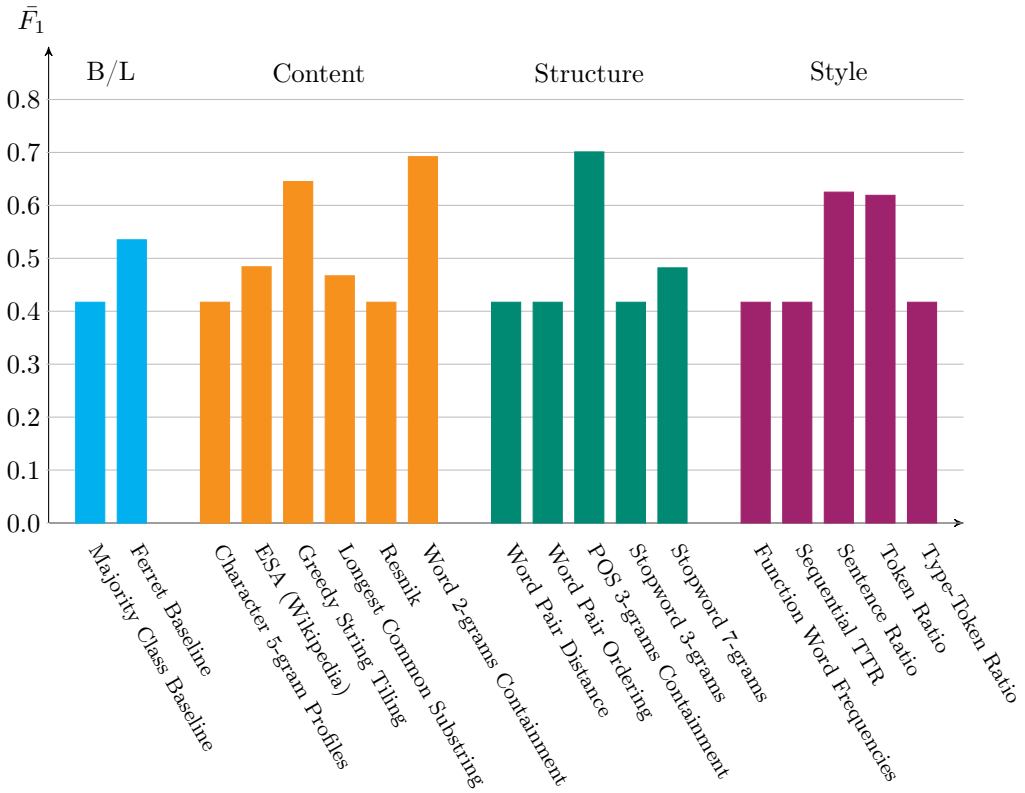


Figure 7.4: Detailed results for a selected set of individual text similarity measures, grouped by similarity dimension, on the METER Corpus

7.4.2 METER Corpus

In the METER Corpus, text reuse occurs between news sources from the UK Press Association (PA) and corresponding newspaper articles. All newspaper articles have been annotated whether they are *wholly derived*, *partially derived*, or *non-derived* from the PA sources. We reported details on this dataset in Section 4.2.1. For the evaluation of our system on this dataset, we followed Sánchez-Vega et al. (2010) and folded the three annotation classes 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 7.10. 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 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. As in the previous section, we further report the detailed results for a selected set of individual text similarity measures in Table 7.2 on page 70, and visualize them in Figure 7.4. From these figures, we learn that many text similarity measures cannot exceed the simple majority class baseline ($\bar{F}_1 = .417$) when applied individually.

In Table 7.11, we show that the performance of text reuse detection always improves over individual measures (see Table 7.2 on page 70) when we combine the measures within their respective dimensions. The corresponding chart is shown

Table 7.10
Results for the best classification on the METER Corpus

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.715	.417
Ferret Baseline	.684	.535
Clough and Stevenson (2011)[7]	.692	.680
Sánchez-Vega et al. (2010)	*.783*	*.705*
Our System	**.802**	**.768**

in Figure 7.5. 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.

Error Analysis We present the confusion matrix for our best configuration in Table 7.12. In total, 50 texts out of 253 have been classified incorrectly: 30 instances of text reuse have not been identified by the classifier, and 20 non-reused texts have been mistakenly labeled as such. However, the original annotations have been carried out by only a single annotator (Gaizauskas et al., 2001) which may have resulted in subjective judgments. Thus, as for the previous dataset in Section 7.4.1, we conducted an annotation study with three annotators to gain further insights into the data. The results[8] show that for 61% of all texts the annotators fully agree. The chance-corrected Fleiss' (1971) agreement $\kappa = .47$ is moderate[9]. The detailed annotations can be found in Appendix A.2.2.

For the 30 instances of text reuse which have not been identified by the classifier, it is particularly interesting to note that many errors are due to the fact that a lower overall text similarity between the possibly reused text and the original source does not necessarily entail the label *no reuse*. The newspaper article about the English singer-songwriter Liam Gallagher, for example, is originally labeled as text reuse. However, our classifier falsely assigned the label *no reuse*. It turns out, though, that the reused text is about four times as long as the original press agency source, with lots of new facts being introduced there. Consequently, only a low similarity score can be computed between the additional material in the newspaper article

Table 7.11
Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the METER Corpus

Text Similarity Dim.	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.759	.712
Structure	.731	.701
Style	.755	.672

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations across dimensions		
Content + Style	.779	.733
Content + Structure	.739	.713
Structure + Style	.767	.739
Content + Structure + Style	**.802**	**.768**

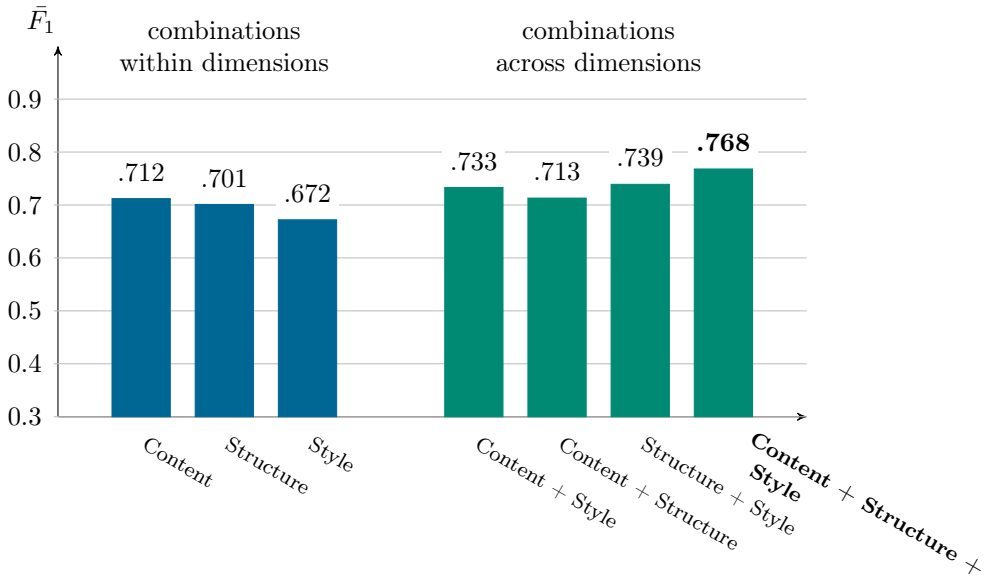


Figure 7.5: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the METER Corpus

and the original source, and the overall similarity score decreases. We conclude that applications will benefit from an improved classifier which better deals with such instances. For example, similarity features could be computed per section, not per document, which would allow to also identify potential instances of text reuse for only partially matching texts.

7.4.3 Webis Crowd Paraphrase Corpus

In the Webis Crowd Paraphrase Corpus, text reuse occurs between book excerpts and the corresponding paraphrases that were acquired using crowdsourcing. The dataset is binary classified into positive and negative samples. We reported further details on this dataset in Section 4.2.1.

We summarize the results on this dataset in Table 7.13. 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

Table 7.12
Confusion matrix for the best classification on the METER Corpus

exp. \ class.	reuse	no reuse
reuse	**151**	30
no reuse	20	**52**

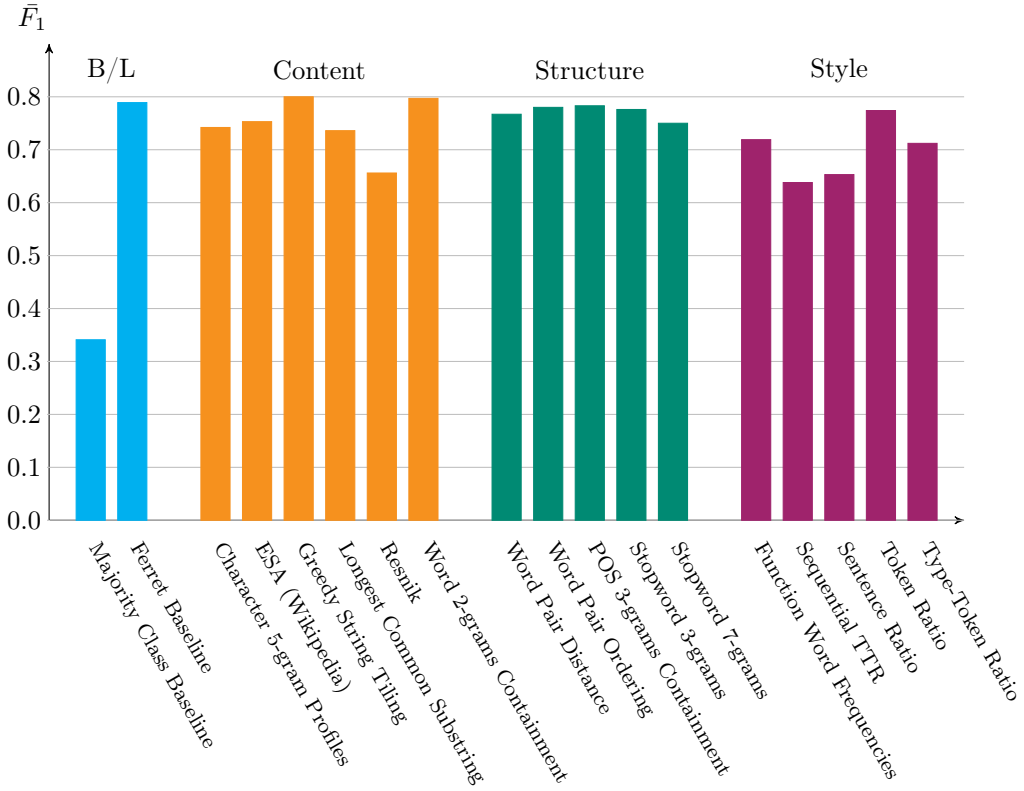


Figure 7.6: Detailed results for a selected set of individual text similarity measures, grouped by similarity dimension, on the Webis Crowd Paraphrase Corpus

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 in the previous sections, we report the detailed results for a selected set of individual text similarity measures in Table 7.2 on page 70 and visualize them in Figure 7.6. These figures show that regardless of the similarity dimension many measures achieve a very reasonable performance when applied individually, with the measures *Greedy String Tiling* and *word 2-grams containment* performing best.

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 7.15 as compared to Table 7.2 on page 70). The corresponding chart is shown in Figure 7.7. 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

Table 7.13
Results for the best classification on the Webis Crowd Paraphrase Corpus

System	Acc.	$\bar{F}_1$
Majority Class Baseline	.517	.341
Ferret Baseline	.794	.789
Clough and Stevenson (2011)[7]	.798	.795
Burrows et al. (2013)	*.839*	*.837*
Our System	**.853**	**.852**

Table 7.14
Text similarity measures used to achieve the best results on the Webis Crowd Paraphrase Corpus, grouped by text dimension

Dimension	Text Similarity Measure	Configurations
Content	Explicit Semantic Analysis	WordNet, with + w/o stopwords
	Greedy String Tiling	
	Jaro	
	Longest Common Subsequence	2 normalizations
	Longest Common Substring	
	n-gram Jaccard	$n = \{6, 14, 15\}$
	Mihalcea et al. (2006)	Resnik (1995) on WordNet (SMT wrapper)
Structure	Word Pair Ordering	
	POS 2-grams	Jaccard
	Stopword 6-grams	
Style	Function Word Frequencies	
	Sequential TTR	
	Token Ratio	

be expected, as their system uses a feature set which also addresses the content dimension exclusively.

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 $\bar{F}_1 = .852$. The final feature set contains 17 text similarity features which are listed in Table 7.14.

Error Analysis We present the confusion matrix for our best classification in Table 7.16. In total, 1,172 (15%) out of 7,859 text pairs have been classified incorrectly. Out of these, our classifier mistakenly labeled 759 instances of negative samples as true paraphrases, while 413 cases of true paraphrases were not recognized.

In the following, we give an example for the error class of *false positives*. In this

Table 7.15
Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Webis Crowd Paraphrase Corpus

Text Similarity Dim.	Acc.	$\bar{F}_1$
Combinations within dimensions		
Content	.840	.839
Structure	.816	.814
Style	.819	.817

Text Similarity Dimension	Acc.	$\bar{F}_1$
Combinations across dimensions		
Content + Style	.844	.843
Content + Structure	.838	.838
Structure + Style	.831	.830
Content + Structure + Style	**.853**	**.852**

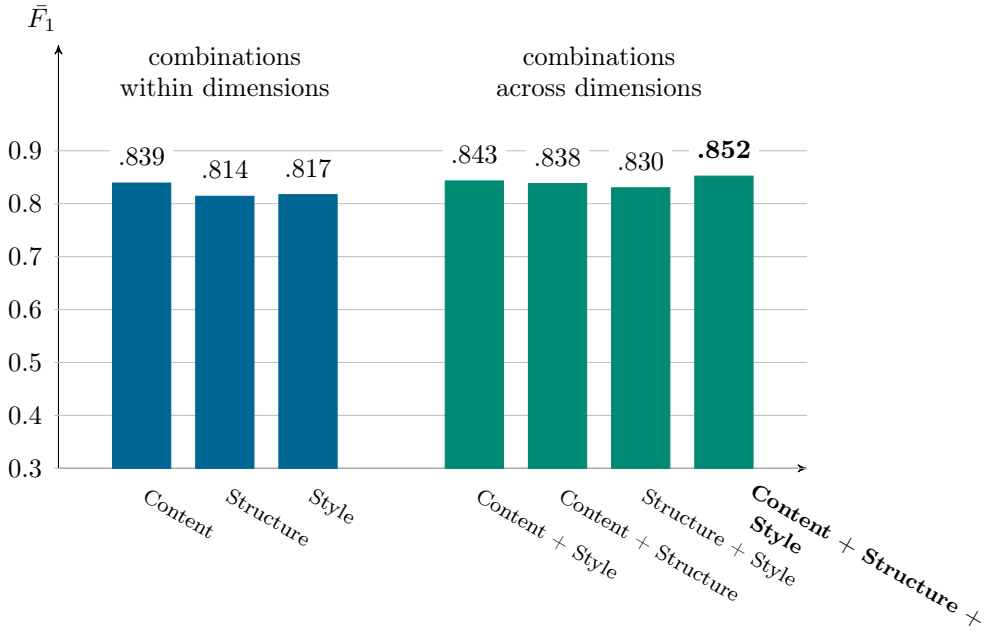


Figure 7.7: Results of the best combinations of text similarity measures within (left) and across (right) dimensions on the Webis Crowd Paraphrase Corpus

example, a non-paraphrased text pair is erroneously classified as a true paraphrase.

(a) *"I have heard many accounts of him," said Emily, "all differing from each other: I think, however, that the generality of people rather incline to Mrs. Dalton's opinion than to yours, Lady Margaret." "I can easily believe it."* (7.1)

(b) *"I have heard many accounts of him," said Emily, "all different from each other: I think, however, that the generality of the people rather inclined to the view of Ms Dalton to yours, Lady Margaret." "That I can not believe."*

As we can see, both texts in Example 7.1 are highly similar with respect to all investigated text dimensions.[12] We attribute such errors to the particular proper-

[12]Example similarity scores for the text dimensions *content*, *structure*, and *style*: Explicit Semantic Analysis (WordNet) 0.86, Word Pair Ordering 0.98, Function Word Frequencies 0.83

Table 7.16
Confusion matrix for the best classification on the Webis Crowd Paraphrase Corpus

exp. \ class.	paraphrase	no para.
paraphrase	**3,654**	413
no para.	759	**3,033**

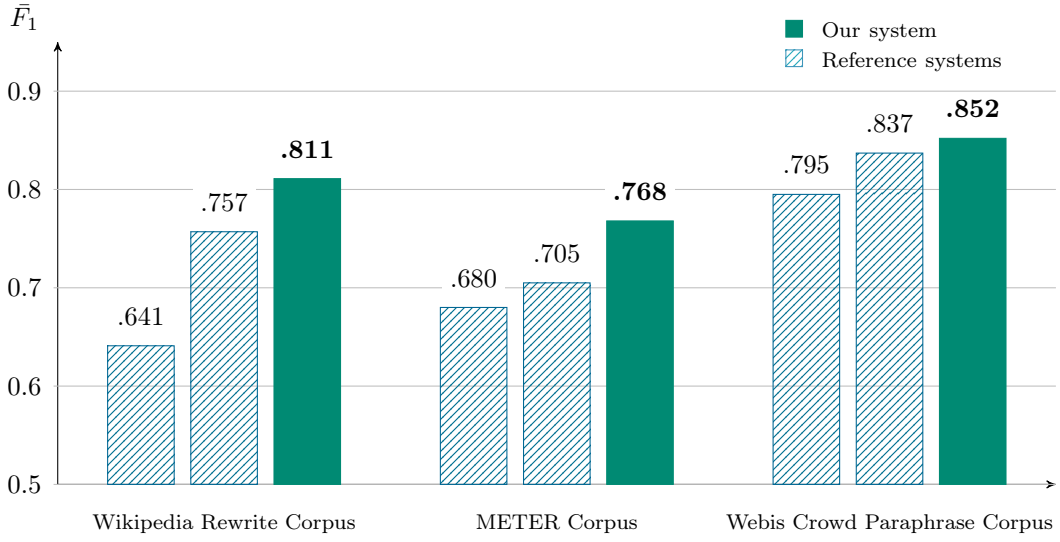


Figure 7.8: Our system (shown in opaque green) outperforms all previous systems (patterned bars) across the three evaluation datasets.

ties of this dataset, which differ from those of the Wikipedia Rewrite Corpus and the METER Corpus (see Sections 7.4.1 and 7.4.2). For those two datasets, the more similar two texts are, the higher their degree of text reuse. For the Webis Crowd Paraphrase Corpus, however, a different interpretation needs to be learned by the classifier: Here, (near-)duplicates and texts with automated word-by-word substitutions, which will receive high similarity scores by any of our content similarity measures, are in fact annotated as *bad* paraphrases, i.e. negative samples. Unrelated texts, empty samples, or texts alike also belong to the class of negative samples. In consequence, positive instances are only those in the medium similarity range. We assume that the unusual definition of positive and negative instances makes it more difficult to learn a proper model for the given data.

7.5 Chapter Summary

In this section, we conducted an extrinisic evaluation of our system on the task of *text reuse detection*. The motivation here was the hypothesis that *content* features alone are not a reliable indicator for text reuse detection, as a reused text may also contain modifications such as split sentences, changed order of reused parts, or stylistic variance (see Figure 7.1). 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 4.2.1. In Figure 7.8, we show a graphical comparison of the best results across all three datasets. As we can see here, the composition of measures consistently outperforms previous systems 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 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*.

Chapter 8

NLP Applications

In the previous chapters, we discussed how we evaluated the performance of the proposed composite model either intrinsically in an isolated setting, or extrinsically by using text similarity as a means for solving a particular problem such as the detection of text reuse. In this chapter, we now stress the importance of text similarity for real-world natural language processing applications. In Section 8.1, we introduce the application scenario *Self-Organizing Wikis*, where wiki users are supported in their everyday tasks by means of natural language processing in general, and text similarity in particular. Furthermore, in Section 8.2 we discuss two applications where text similarity is highly beneficial: *question answering* and *textual entailment recognition*. We conclude this chapter with a summary.

8.1 Self-Organizing Wikis

We first introduce the *Self-Organizing Wikis* scenario in Section 8.1.1. In Section 8.1.2, we discuss the architecture of our system, detail the technical aspects of the system, and introduce the system's core components along with a walk-through example of the information flow. In Section 8.1.3, we describe two use cases within the application scenario where text similarity is particularly beneficial: *duplicate detection* and *link discovery*. Finally, in Section 8.1.4, we elaborate on further use cases which have been implemented successfully as part of our system.

8.1.1 Scenario Description

Wikis are web-based collaborative content authoring systems (Leuf and Cunningham, 2001). As they offer fast and simple means for adding and editing content, they are used for various purposes such as creating encyclopedias (e.g. *Wikipedia*[1]), constructing dictionaries (e.g. *Wiktionary*[2]), or hosting online communities (e.g. *ACLWiki*[3]). However, wikis do not enforce their users to structure pages or add complementary metadata. They thus often end up as a mass of unmanageable pages with meaningless page titles and no usable link structure (Buffa, 2006).

[1] http://www.wikipedia.org
[2] http://www.wiktionary.org
[3] http://aclweb.org/aclwiki

To solve this issue, we present the *Self-Organizing Wikis* application scenario. The basic idea is to use natural language processing techniques to support wiki users with their everyday tasks of adding, organizing, and finding content. As a system implementation, we therefore created the *Wikulu* system[4] which integrates natural language processing techniques seamlessly with any wiki by means of an HTTP proxy architecture. While we stress the importance of text similarity for this application, our system is not limited to text similarity only, but allows to employ any type of natural language processing techniques. We present further details of our system architecture in Section 8.1.2.

Supporting wiki users with natural language processing has not attracted a lot of research attention yet. A notable exception is the work by Witte and Gitzinger (2007). They propose an architecture to connect wikis to services providing NLP functionality which are based on the *General Architecture for Text Engineering (GATE)* (Cunningham et al., 2002). Contrary to Wikulu, though, their system does not integrate transparently with an underlying wiki engine, but rather uses a separate application to perform natural language processing. Thereby, wiki users can leverage the power of NLP algorithms, but have to interrupt their current work to switch to a different application. Moreover, their system is only loosely coupled with the underlying wiki engine. While it allows to read and write to existing pages, it does not allow further modifications such as adding custom user interface controls.

A lot of work in the wiki community is done in the context of Wikipedia. For example, the *FastestFox*[5] plug-in for Wikipedia is able to suggest links to related articles. However, unlike Wikulu, FastestFox is tailored towards Wikipedia and cannot be used with any other wiki platform.

8.1.2 System Architecture

In this section, we detail our system architecture and describe what is necessary to make NLP algorithms available through our system. The core system is joint work with Johannes Hoffart (Hoffart et al., 2009b) and Nicolai Erbs (Bär et al., 2011a).

We show the overall architecture of our *Wikulu* system in Figure 8.1. Wikulu is implemented as an HTTP proxy server which intercepts the communication between the web browser and the underlying wiki engine, and further builds upon a modular architecture and allows to run any NLP component which is based on *Apache UIMA* (Ferrucci and Lally, 2004) using an extensible plugin mechanism. In order to run Wikulu, no further modifications to the original wiki installation are necessary other than activating the proxy server in a user's web browser. Currently, our system contains adaptors for two widely used wiki engines: *MediaWiki*[6] and *TWiki*[7]. Adaptors for other wiki engines can be added with minimal effort.

In Figure 8.2, we show a screenshot of what the integration of Wikulu with Wikipedia looks like.[8] The additional user interface components are integrated

[4]The name is based on the Hawaiian terms *wiki* (en. *fast*) and *kukulu* (*to organize*).

[5]http://www.smarterfox.com

[6]http://www.mediawiki.org (e.g. used by Wikipedia)

[7]http://www.twiki.org

[8]As screenshots only provide a limited overview of Wikulu's capabilities, we refer the reader to a screencast: http://www.ukp.tu-darmstadt.de/research/current-projects/wikulu

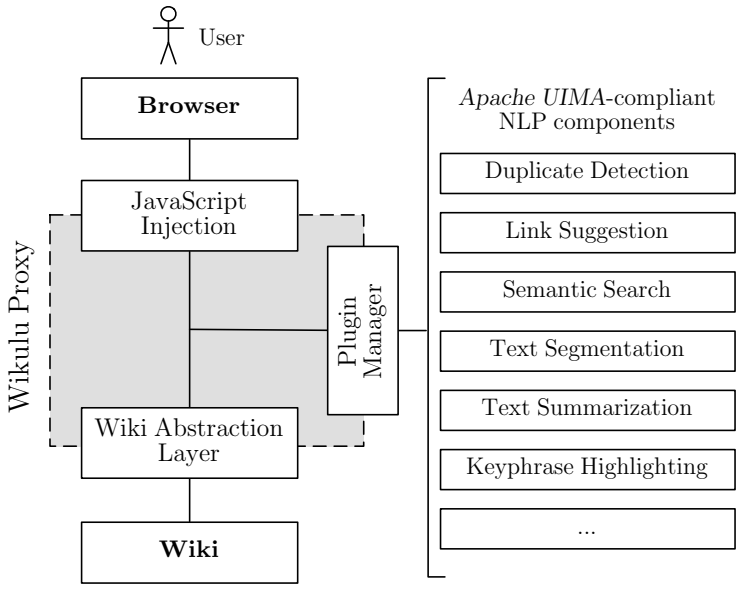


Figure 8.1: Wikulu acts as a proxy server which intercepts the communication between the web browser and the underlying wiki engine. Its plugin manager allows to integrate any *Apache UIMA*-compliant NLP component.

into Wikipedia's default toolbar (highlighted by a red box in the screenshot). In this example, the user has requested to highlight *keyphrases* (i.e. selected words or phrases which describe the main idea of the text) by clicking the corresponding button in the augmented toolbar. Wikulu then invokes the corresponding NLP component, and highlights the keyphrases in the article, so that the user can quickly grasp the main idea of the wiki article.

In the following, we introduce each module in detail: (a) the *proxy server* which allows to use Wikulu with any underlying wiki engine, (b) the *JavaScript injection* that bridges the gap between the client- and server-side code, (c) the *plugin manager* which gives access to any *Apache UIMA*-based NLP component, and (d) the *wiki abstraction layer* which offers a high-level interface to typical wiki operations such as reading and writing the wiki contents. To illustrate Wikulu's information flow, we further conclude this section with a walk-through example.

Proxy Server

Wikulu is designed to work with any underlying wiki engine such as *MediaWiki* or *TWiki*. Consequently, we implemented it as an HTTP proxy server which allows it to be enabled at any time by changing the proxy settings of a user's web browser.[9] The proxy server intercepts all requests between the wiki user and the underlying wiki engine. For example, Wikulu allows to redirect particular requests to its language processing components or augment the default wiki toolbar by additional functionality.

[9]The process of enabling a custom proxy server can be simplified by using web browser extensions such as *FoxyProxy* (http://getfoxyproxy.org).

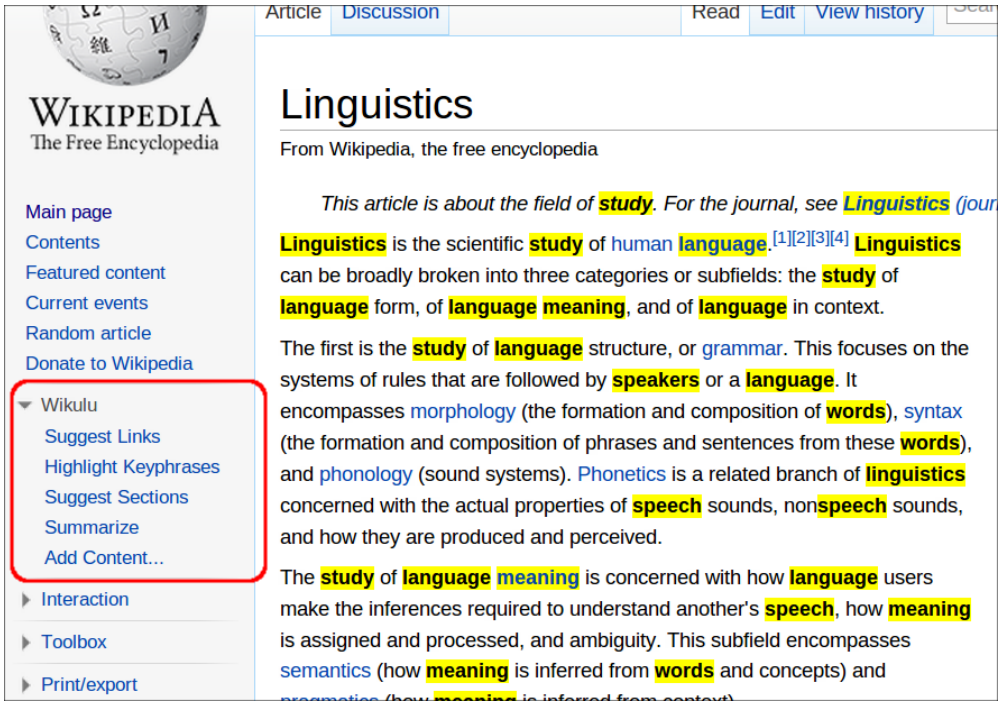


Figure 8.2: Integration of Wikulu with Wikipedia. The augmented toolbar (red box) and the results of a keyphrase extraction algorithm (yellow text spans) are highlighted.

JavaScript Injection

Wikulu modifies the requests between the web browser and the target wiki by injecting custom client-side JavaScript code. Wikulu is thus capable of altering the default behavior of the wiki engine, e.g. replacing a keyword-based retrieval by enhanced search methods, adding novel behavior such as additional toolbar buttons or advanced input fields, or augmenting the originating web page after a certain request has been processed, e.g. an NLP algorithm has been run.

Plugin Manager

Wikulu does not perform language processing itself. It relies on *Apache UIMA*-compliant NLP components (such as a dedicated *text similarity* component) which use wiki pages (or parts thereof) as input texts. Wikulu offers a sophisticated plugin manager which takes care of dynamically loading those components. The plugin loader is designed to run plugins either every time a wiki page loads, or manually by picking them from the augmented wiki toolbar.

The NLP components are available as server-side Java classes. Via direct web remoting[10], those components are made accessible through a JavaScript proxy object. Wikulu offers a generic language processing plugin which takes the current page contents as input text, runs an NLP component, and writes its output back to the wiki. To run a custom *Apache UIMA*-compliant NLP component with Wikulu, one just needs to plug that particular NLP component into the generic plugin. No further adaptations to the generic plugin are necessary. However, more advanced

[10] http://directwebremoting.org

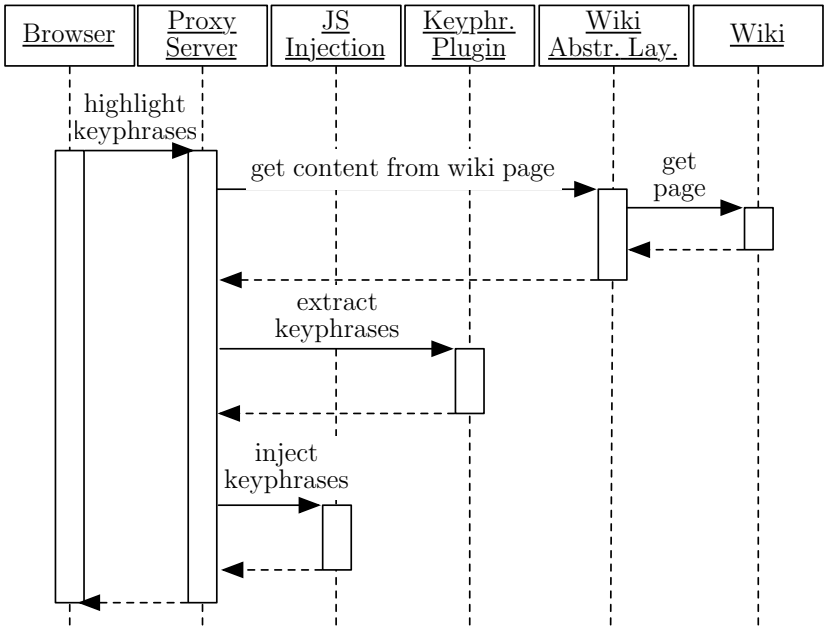


Figure 8.3: Illustration of Wikulu's information flow when a user has requested to highlight *keyphrases* on the current page

users may create fully customized plugins.

Wiki Abstraction Layer

Wikulu communicates with the underlying wiki engine via an abstraction layer. That layer provides a generic interface for accessing and manipulating the underlying wiki engine. Thereby, Wikulu can both be tightly coupled to a certain wiki instance such as *MediaWiki* or *TWiki*, while being flexible at the same time to adapt to a changing environment. New adaptors for other target wiki engines such as *Confluence*[11] can be added with minimal effort.

Walk-Through Example

In Figure 8.3, we show an illustration of Wikulu's information flow. Let us assume that a user encounters a wiki page which is rather lengthy. The traditional option would be to read through all contents of the page to get the main idea of the wiki article. However, by using Wikulu she can rely on a number of NLP components which help her grasp the main idea in less time.

For example, she realizes that Wikulu's keyphrase extraction component might help her to better grasp the idea of this page at a glance, so she activates Wikulu by setting her web browser to pass all requests through the proxy server. After applying the settings, the JavaScript injection module adds additional links to the wiki's toolbar on the originating wiki page (as shown in Figure 8.2 for Wikipedia). Having decided to apply keyphrase extraction, she then invokes that NLP component by clicking the corresponding link. Before the request is passed to that component, Wikulu extracts the wiki page contents using the high-level wiki abstraction layer.

[11] http://www.atlassian.com/software/confluence

Thereafter, the request is passed via direct web remoting to the NLP component which has been loaded by Wikulu's plugin mechanism. After processing the request, the extracted keyphrases are returned to Wikulu's custom JavaScript handlers and finally highlighted in the originating wiki page.

8.1.3 Text Similarity Use Cases

We now describe two use cases within the Self-Organizing Wikis application scenario where users are supported by means of text similarity techniques: *duplicate detection* and *link discovery*.

Duplicate Detection

Whenever users add new content to a wiki there is the danger of duplicating already contained information. In order to avoid duplication, users would need comprehensive knowledge of what content is already present in the wiki, which is almost impossible for large wikis like Wikipedia. Wikulu helps to detect potential duplicates by computing the text similarity between newly added content and each existing wiki page. If a potential duplicate is detected, the user is notified and may decide to augment the duplicate page instead of adding a new one. Wikulu relies on text similarity measures such as *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007) and *Latent Semantic Analysis* (Landauer et al., 1998), which have been introduced in Chapter 3.

In Figure 8.4, we show a screenshot of the implementation in Wikulu. Here, the user has requested that text similarity is computed between the current page about the *Web Ontology Language* and all other pages in the wiki. As we can see in the (shortened) screenshot, some wiki pages have a much higher similarity to the current page than others. In consequence, a user may decide to inspect those pages further, e.g. to merge some of them with the current page in order to avoid content duplication.

Link Discovery

While many wiki users readily add textual contents to wikis, they often restrain from also adding links to related pages as this is a very tedious task. However, links in wikis are crucial as they allow users to quickly navigate from one page to another, or browse through the wiki. Therefore, it may be reasonable to augment a page about the topic *sentiment analysis* (Pang and Lee, 2008) by a link to a page providing related information such as evaluation datasets. Wikulu supports users in this tedious task by automatically suggesting links. Link suggestion thereby is a two-step process: (a) first, suitable text phrases are extracted which might be worth to place a link on, and (b) for each phrase, related pages are ranked by comparing their relevance to the current page, and then presented to the user. The user may thus decide whether she wants to use a detected phrase as a link or not, and if so, which other wiki page to link this phrase to. Wikulu currently integrates link suggestion algorithms by Geva (2007) and Itakura and Clarke (2007).

In Figure 8.5, we show a screenshot of the link discovery implementation in Wikulu. Here, suitable text phrases to place a link on are highlighted in green and

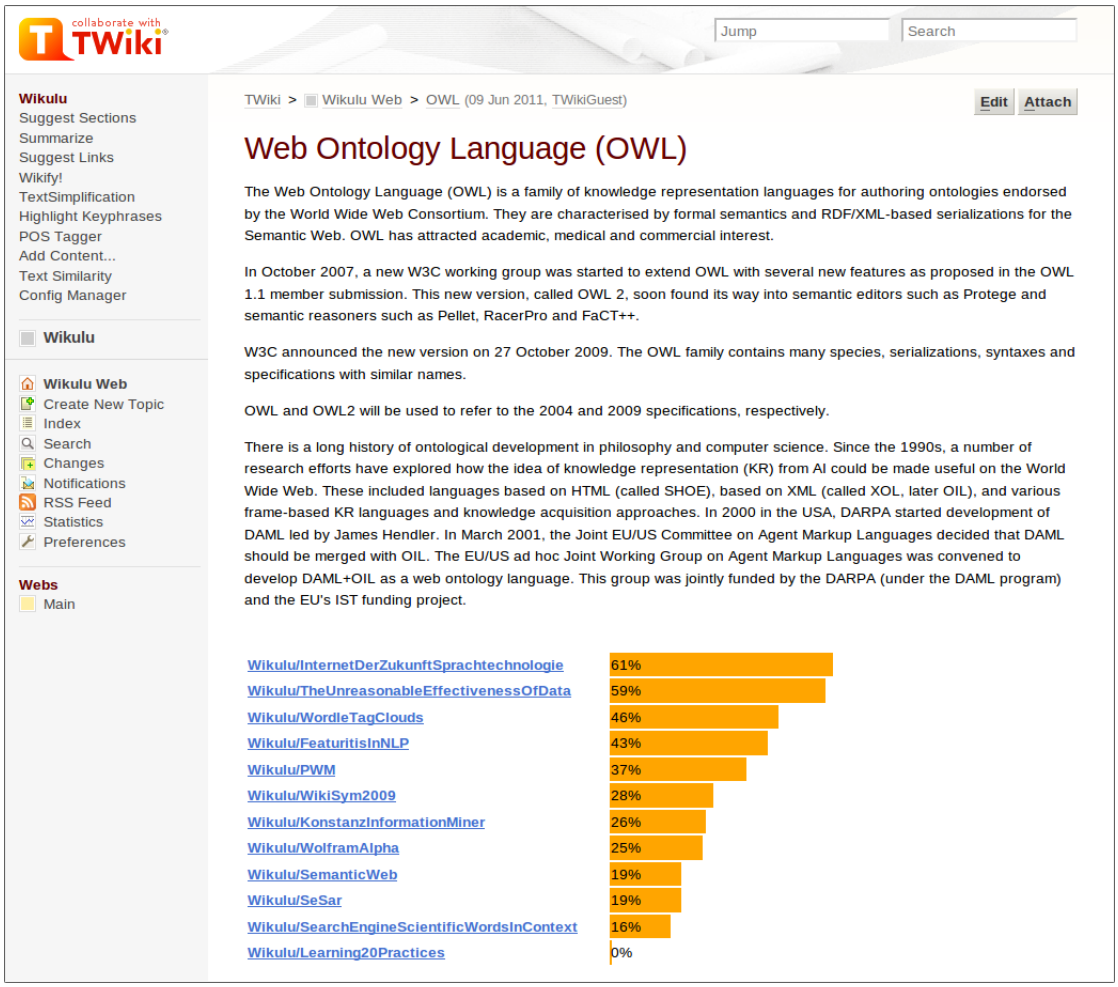


Figure 8.4: Screenshot of *duplicate detection* implemented in Wikulu. Text similarity was computed between the current page and all other pages in the wiki.

augmented with a "+" button to the right. If the user chooses to accept one of these suggestions, she just needs to click the button in order to select one of the selected target pages. The highlighted text phrase is then transformed into a hyperlink.

Hoffart et al. (2009b) further applied text similarity techniques for link discovery in the *INEX 2009 Link the Wiki competition* (Huang et al., 2010). In this international competition, Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) was used to find suitable target pages for a given article where links are to be placed on. In a qualitative analysis, Hoffart et al. (2009b) showed that text similarity is beneficial for identifying link targets of ambiguous link anchors.

8.1.4 Further Use Cases

In the following, we briefly summarize further use cases which have been implemented as part of the Self-Organizing Wikis application scenario, but do not necessarily employ text similarity techniques. We envision, for example, that wiki users are supported with reading longer texts by means of techniques for *keyphrase highlighting*. That is, using keyphrase extraction methods such as *TextRank* (Mihalcea and Tarau, 2004) to highlight important words on the current wiki page so that a

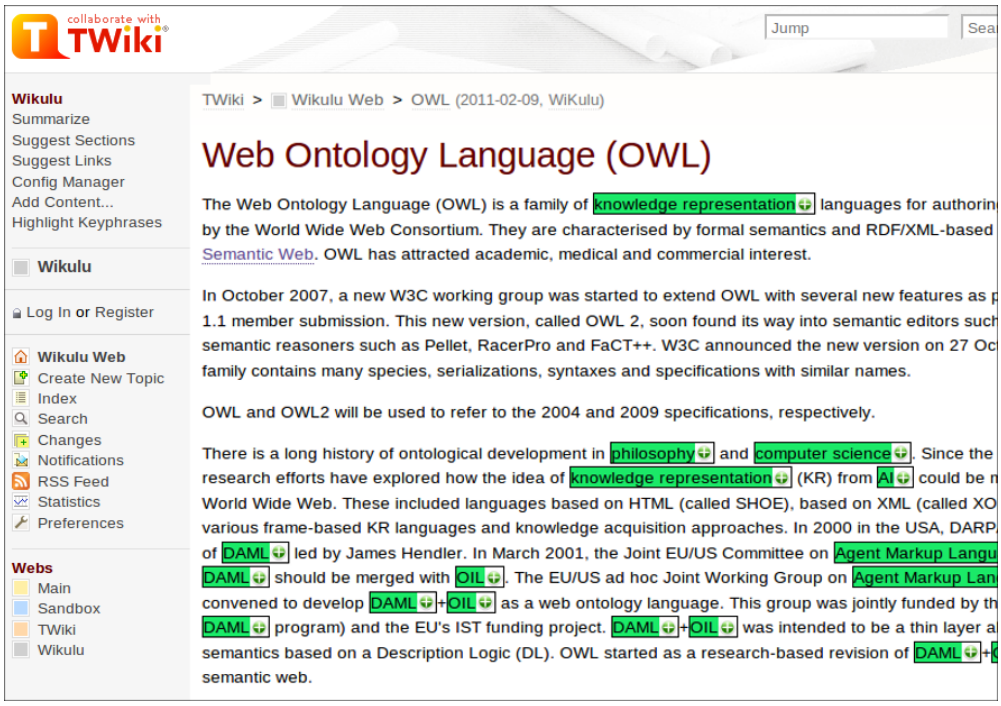


Figure 8.5: Screenshot of *link discovery* implemented in Wikulu. Suitable text phrases to place a link were automatically identified and are highlighted in green.

user can get a better grasp of the idea that the texts conveys in less time. In the following, we give an overview of such use cases. This is joint work with Nicolai Erbs (Bär et al., 2011a).

Semantic Search

The capabilities of a wiki's built-in search engine are typically rather limited as it traditionally performs mostly keyword-based retrieval. If the query keyword is not found in the wiki, the query returns an empty result set. However, a page might exist which is semantically related to the keyword, and should thus yield a match.

As the search engine is typically a core part of the wiki system, it is rather difficult to modify its behavior. However, by leveraging Wikulu's architecture, we can replace the default search mechanisms by algorithms which allow for *semantic search* to alleviate the vocabulary mismatch problem (Gurevych et al., 2007).

Segmenting Long Pages

Due to the open editing policy of wikis, pages tend to grow rather fast. For users, it is thus a major challenge to keep an overview of what content is present on a certain page. Wikulu therefore supports users by analyzing long pages through employing text segmentation algorithms which detect topically coherent segments of text. It then suggests segment boundaries which the user may or may not accept for inserting a subheading which makes pages easier to read and better to navigate. When accepting one or more of these suggested boundaries, Wikulu stores them persistently in the wiki. Wikulu currently integrates text segmentation methods such as *TextTiling* (Hearst, 1997) or *C99* (Choi, 2000).

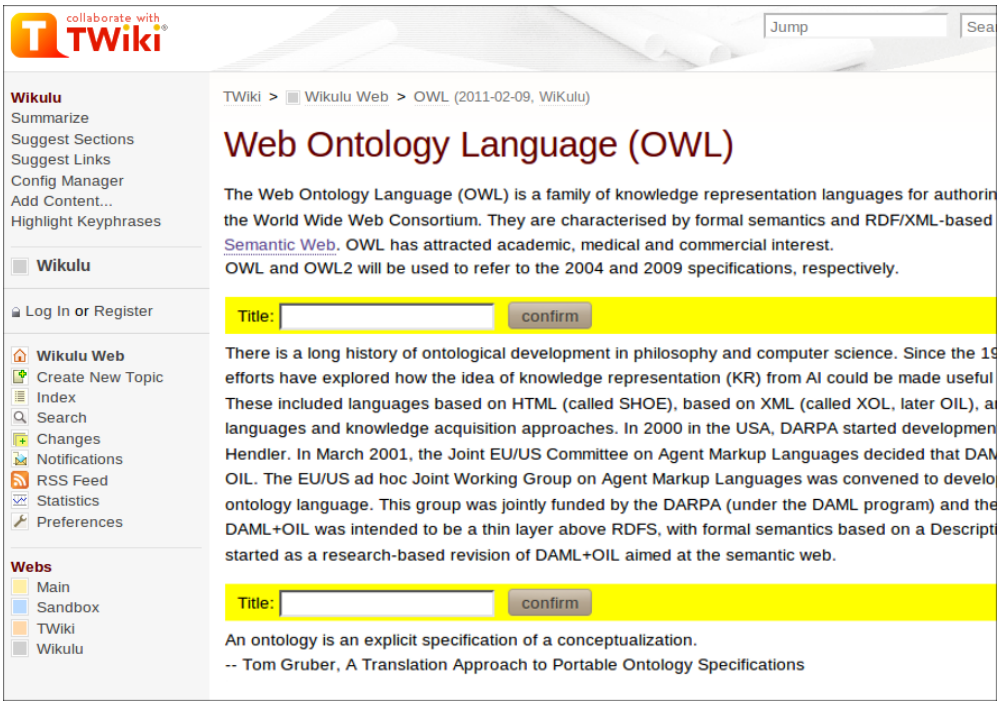


Figure 8.6: Screenshot of *text segmentation* implemented in Wikulu. For the original wiki page, three segments (boundaries highlighted by yellow bars) were automatically suggested.

We show an example screenshot in Figure 8.6. Here, for the original wiki page three segments were proposed by the employed text segmentation method. The segment boundaries are highlighted by yellow bars and further allow to manually add a segment title. For future version, we envision that suitable segment titles may be proposed automatically.

Summarizing Pages

Similarly to segmenting pages, Wikulu makes long wiki pages more accessible by generating an extractive summary. While abstractive summaries create a summary in own words, extractive summaries analyze the original wiki text sentence-by-sentence, rank each sentence, and return a list of the most important ones (Carenini and Cheung, 2008; Erkan and Radev, 2004). Wikulu integrates extractive text summarization methods such as *LexRank* (Erkan and Radev, 2004).

In Figure 8.7, we show the results of the current text summarization component contained in our Wikulu system. For the original wiki page about the *Web Ontology Language* (see Figure 8.4), an extractive summary was generated. In our system implementation, we list the top-ranked sentences as a list which can be inspected by the wiki user much faster than the original article.

Highlighting Keyphrases

Another approach to assist users in better grasping the idea of a wiki page at a glance is to highlight important keyphrases. As Tucker and Whittaker (2009) have shown, highlighting important phrases assists users with reading longer texts and

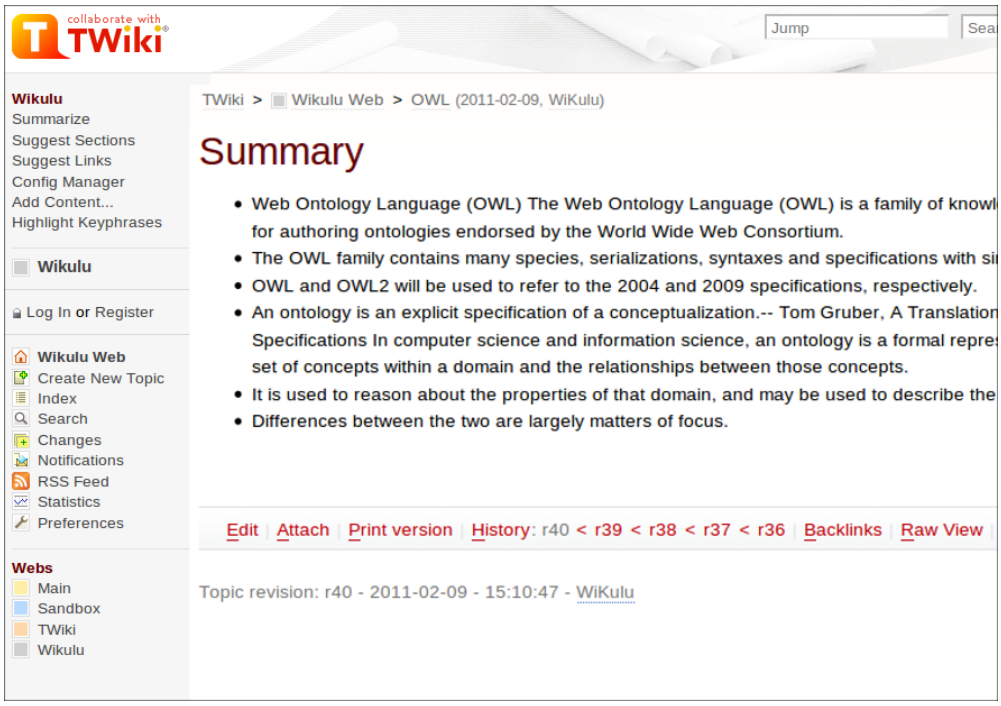


Figure 8.7: Screenshot of *text summarization* implemented in Wikulu. For the original wiki page, an extractive summary containing the top-ranked sentences was generated.

yields faster understanding. Wikulu thus improves readability by employing automatic keyphrase extraction algorithms. Additionally, Wikulu allows to dynamically adjust the number of keyphrases shown by presenting a slider to the user. We integrated keyphrase extraction methods such as *TextRank* (Mihalcea and Tarau, 2004) and *KEA* (Witten et al., 1999). An example screenshot in which keyphrases in a Wikipedia article were highlighted was shown in Figure 8.2 on page 88.

Besides the use cases outlined above, further use cases for supporting wiki users include (i) visually analyzing the results of NLP algorithms, (ii) educational purposes, and (iii) enabling semantic wikis. We briefly describe them in the following.

Visually Analyzing the Results of NLP Algorithms

Wikulu facilitates analyzing the results of NLP algorithms by using wiki pages as input documents and visualizing the results directly on that page. Consider an NLP algorithm which performs sentiment analysis (Pang and Lee, 2008). Typically, we were to put our analysis sentences in a text file, launch the NLP application, process the file, and would read the output from either a built-in console or a separate output file. This procedure suffers from two major drawbacks: (a) it is inconvenient to copy existing data into a custom input format which can be fed into the NLP system, and (b) the textual output does not allow presenting the results in a visually rich manner. Wikulu tackles both challenges by using wiki pages as input/output documents.

For example, by running a part-of-speech tagger right from within the wiki, its output can be written back to the originating wiki page, resulting in visually rich, possibly interactive presentations. An example screenshot is shown in Figure 8.8

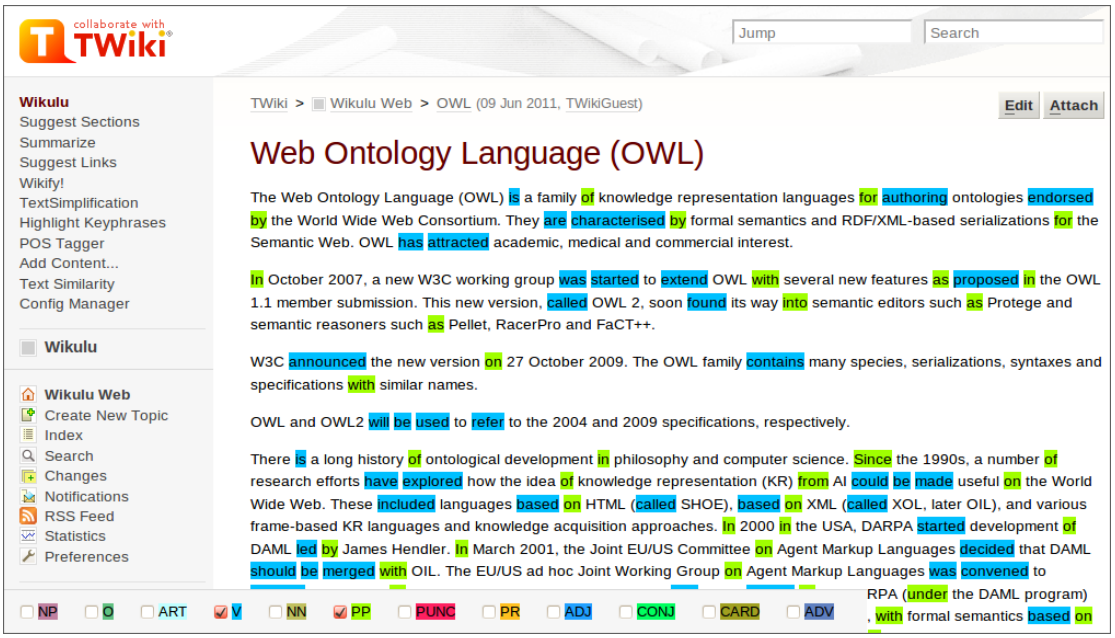


Figure 8.8: Screenshot of the visual result analysis implemented in Wikulu. The output of a part-of-speech tagger is shown as a visually rich presentation on the original wiki page.

where verbs and prepositions are highlighted.

Educational Purposes

Wikulu is a handy tool for educational purposes as it allows to (a) rapidly create textual data in a collaborative manner, and (b) visualize the results of NLP algorithms on such data, as described above. That way, students can gather hands-on experience by experimenting with NLP components in an easy-to-use wiki system. They can both collaboratively edit input documents, and explore possible results of e.g. different configurations of NLP components. In our system prototype, we integrated the highlighting of parts-of-speech tags which have been determined by a part-of-speech tagger (Schmid, 1994).

Enabling Semantic Wikis

Semantic wikis such as the *Semantic MediaWiki* (Krötzsch et al., 2006) augment standard wikis with machine-readable semantic annotations of pages and links. As those annotations have to be entered manually, this step is often skipped by users which severely limits the usefulness of semantic wikis. Wikulu could support users e.g. by automatically suggesting the type of a link by means of relation detection or the type of a page by means of text categorization. Thus, Wikulu could constitute an important step towards the *semantification* of the content contained in wikis.

8.2 Further Applications

In this section, we report details on two further applications which are not yet part of our Wikulu system, but are closely related and may benefit future versions of

our system: *question answering* and *textual entailment recognition*. The work on question answering was conducted by Walter et al. (2012), who built upon our open source framework *DKPro Similarity* (see Chapter 9) to carry out the experiments. The work on textual entailment recognition was carried out as a set of own experiments, and resulted in a contribution to a novel open source platform for textual inference. In the following, we report details on both applications.

8.2.1 Question Answering

Walter et al. (2012) experimented with question answering over *Linked Data*. The task of question answering thereby addresses the challenge of making machines capable of answering queries which are formulated in natural language. The term *Linked Data* refers to the best practices of publishing any kind of structured data on the Web which is connected by typed links (Bizer et al., 2009a). Prominent datasets that adopt Linked Data are *DBpedia*[12] (Bizer et al., 2009b), *Freebase*[13], or *YAGO*[14] (Suchanek et al., 2007; Hoffart et al., 2013).

In their experiments, Walter et al. (2012) introduce a system which follows a layered approach to question answering: Only questions that cannot be answered using simple processing steps are passed on to more complex and hence more expensive components. Their architecture consists of five layers where the final layer computes text similarity using *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007) (see Section 3.2) based on our implementation which is available as part of our open source framework *DKPro Similarity*, which we will introduce in Chapter 9.

Evaluation was carried out on the DBpedia test set which was provided by the *2nd Open Challenge on Question Answering over Linked Data (QALD-2)*.[15] The final dataset comprises 72 questions.[16] We list some examples below in Example 8.1.

(a) *What is the currency of the Czech Republic?* (8.1)

(b) *Who was the wife of U.S. president Lincoln?*

(c) *Was Natalie Portman born in the United States?*

Walter et al. (2012) particularly study the impact of the individual layers on the overall system performance. The evaluation metric used is F' score which is defined as the harmonic mean of the coverage and the F score.[17] A simple lookup is performed first, where natural language expressions of the query such as `Czech Republic` are mapped onto uniform resource identifiers such as `<http://dbpedia.org/resource/Czech Republic>`, and a SPARQL query is then performed to answer the question. This method achieves a poor overall F' score of 0.25. However, taking all layers into account up to Explicit Semantic Analysis achives the best overall performance with $F' = 0.52$.

[12] http://dbpedia.org
[13] http://www.freebase.com
[14] http://www.mpi-inf.mpg.de/yago-naga/yago
[15] http://www.sc.cit-ec.uni-bielefeld.de/qald-2
[16] Walter et al. (2012) filtered some questions which were not solvable.
[17] For details, we refer the reader to the work by Walter et al. (2012).

Table 8.1
Statistics for the five datasets which have been used for our experiments with textual entailment. All text pairs in these datasets are accompanied by human judgments whether an entailment relationship holds nor not. The datasets are (roughly) equally split into positive (i.e. entailment) and negative (i.e. non-entailment) samples.

Dataset	Length in Terms (⌀)	Training Pairs (positive/negative)	Test Pairs (positive/negative)	Agreement (Fleiss' (1971) κ)
RTE1 (Dagan et al., 2006)	2–76 (20)	287 (143/144)	800 (400/400)	0.60
RTE2 (Bar-Haim et al., 2006)	4–110 (20)	800 (400/400)	800 (400/400)	0.78
RTE3 (Giampiccolo et al., 2007)	4–117 (22)	800 (412/388)	800 (410/390)	0.75
RTE4 (Giampiccolo et al., 2008)	4–134 (26)	–	1,000 (500/500)	n/a
RTE5 (Bentivogli et al., 2009)	4–264 (59)	600 (300/300)	600 (300/300)	0.97 [18]

The results show that taking more sophisticated processing layers into account benefits the overall system performance. In particular, using all layers up to the final text similarity layer shows the best overall results. From these results we conclude that text similarity techniques can indeed benefit question answering. While in these experiments only Explicit Semantic Analysis has been used as a measure of choice, we argue that a composition of text similarity measures (see Chapter 5) may be interesting to explore in future work and lead to even better overall results.

8.2.2 Textual Entailment Recognition

In Section 1.3, we introduced *textual entailment* as a task that is highly related to text similarity. It differs, though, in that it is defined as a unidirectional relationship between two texts, and in that it operates on binary judgments rather than continuous similarity scores. However, we argue that text similarity measures are a highly valuable tool in any textual entailment recognition setting: For a given document collection, text similarity measures can be employed in a first step to filter those text pairs that are highly similar to each other. In a second step, these pairs can then be further investigated by a dedicated textual entailment system.

Experimental Setup

We therefore conducted experiments on the datasets from the *Recognizing Textual Entailment (RTE)* challenge series (Dagan et al., 2006). We used the data from the first five challenges and will refer to it by the original name *RTE1* to *RTE5*.[19] The datasets are (roughly) equally distributed into positive (i.e. entailment) and negative (i.e. non-entailment) samples. We list the detailed dataset statistics in Table 8.1. The system we used for the experiments throughout this section modifies the original experimental setup described in Chapter 5 as follows:

Pre-processing no modifications; see Section 5.4

[18] The exceptionally high agreement is due to the fact that Bentivogli et al. (2009) filtered out "annotators' mistakes" and left only "real disagreements". The reported agreement is based on the filtered annotations.

[19] http://aclweb.org/aclwiki/index.php?title=Textual_Entailment_Resource_Pool

Table 8.2
Results across the RTE1–5 datasets in terms of *confidence-weighted score (RTE1)* or *average precision (RTE2–5)*, respectively. Additionally, we list the results of the best and worst reference systems for comparison.

System	RTE1	RTE2	RTE3	RTE4	RTE5
Our System	.592	.594	.667	.656	.628
Best Reference System[20]	.686	.808	.881	.741	.701
Worst Reference System[20]	.507	.504	.496	.494	.530

Feature Generation The text similarity measures that we used for the experiments are the same as for the extrinsic evaluation. We described the set of measures in Section 7.2: We used the measures along the *structural* and *stylistic* text dimensions as originally described, and used the measures along the *content* dimension as summarized in Table 7.1 on page 69.

Feature Combination We still use the pre-computed similarity scores as input for the machine learning classifier, but combine them using an SVM classifer (*SMO* implementation) and a multinomial logistic regression classifier (*Logistic* implementation) from the Weka toolkit (Hall et al., 2009). These classifiers are able to predict nominal classes of textual entailment.

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

We trained the classifiers on the available training data for each dataset, e.g. for RTE1 we trained on the RTE1 training data and tested on the RTE1 test data. For RTE4, however, no training data was provided and it was suggested to train the system on the RTE3 dataset. We followed this suggestion in the evaluation.

In the following, we report the best results obtained on the RTE1 to RTE5 datasets. RTE1 to RTE3 contain pairs of a *text* and a *hypothesis* (see Section 1.3) for which an entailment relationship holds or not. In addition, RTE4 and RTE5 contain both two- and three-way classifications, where the third class is *unknown*, i.e. it is not possible to determine whether the text entails the hypothesis (Giampiccolo et al., 2008). For the evaluation on the RTE4 and RTE5 data, however, we sticked to the classical binary task with positive and negative entailment relationships.

Evaluation was carried out in terms of *accuracy*, i.e. the number of correctly predicted pairs divided by the total number of pairs, and a score which additionally includes how *confident* a system is about the predicted scores. In RTE1, the weighted score is called *confidence-weighted score (CWS)* and computes the *average precision* on both positive and negative entailment pairs (Dagan et al., 2006). From RTE2 onwards, CWS was replaced by the common variant of the average precision (AP) which operates only on the positive entailment pairs (Bar-Haim et al., 2006). In the following, we report all results of our system in terms of CWS/AP rather than accuracy, as we assume it is important for a system to be able to tell how confident it is about the judgments.

[20] For the reference systems both the *accuracy* as well as a confidence-weighted evaluation metric (*confidence-weighted score/average precision*) are reported. We report the *best* and *worst* system with respect to the *confidence-weighted score* or *average precision*, respectively.

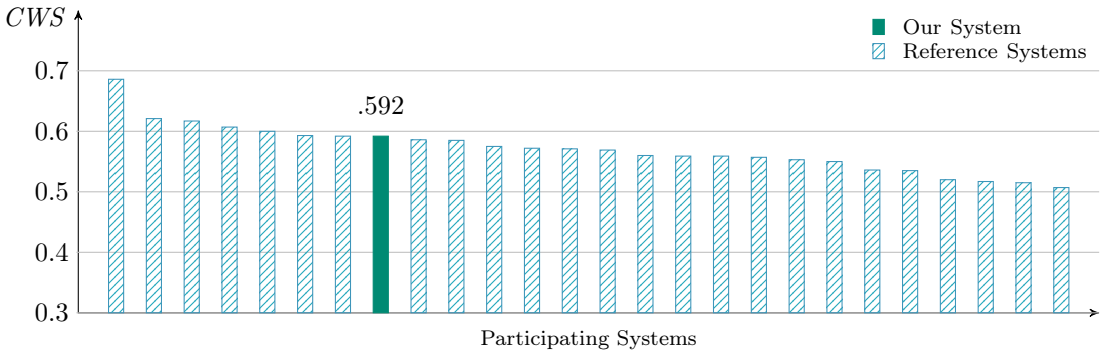


Figure 8.9: Results on the RTE1 dataset in terms of *confidence-weighted score*. Our system (shown in opaque green) is ranked #8 out of 26 systems in total.[21]

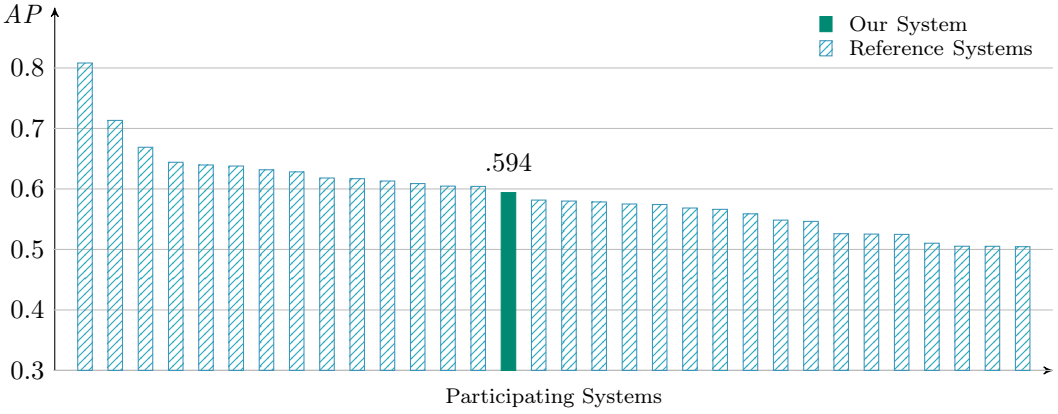


Figure 8.10: Results on the RTE2 dataset in terms of *average precision*. Our system (shown in opaque green) is ranked #15 out of 32 systems in total.[21]

Results

In Table 8.2, we report our best results across the five RTE datasets. In addition, we report the best and worst reference systems for comparison.[20] Thereby, we compare to the reference systems in terms of *confidence-weighted score* or *average precision*, respectively.[21] The corresponding charts are shown in Figures 8.9 to 8.13.

In summary, we see a varying performance of our system across the five datasets. However, the results of our best system configuration is always at least in the medium range compared to the reference systems. This is particularly interesting to note, as our system does not contain any measures which are specifically tailored towards the recognition of textual entailment. Moreover, for the RTE4 dataset (see Figure 8.12), our system is ranked #2 in terms of average precision, only to be outperformed by a single reference system.[21]

We conclude that our system, which uses a composition of text similarity measures, shows particularly promising results for confidence-weighted evaluation metrics where not only the classification into positive and negative entailment pairs is important, but the system also needs to tell how confident it is about its decision.

[21] We compare only with systems which operate on all entailment pairs and report results for *accuracy* as well as for the *confidence-weighted score* or *average precision*, respectively.

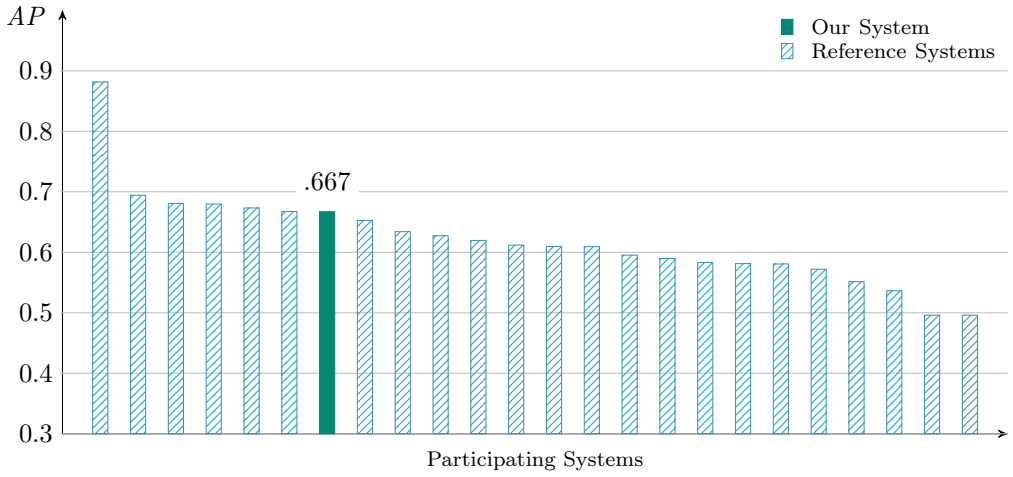


Figure 8.11: Results on the RTE3 dataset in terms of *average precision*. Our system (shown in opaque green) is ranked #7 out of 24 systems in total.[21]

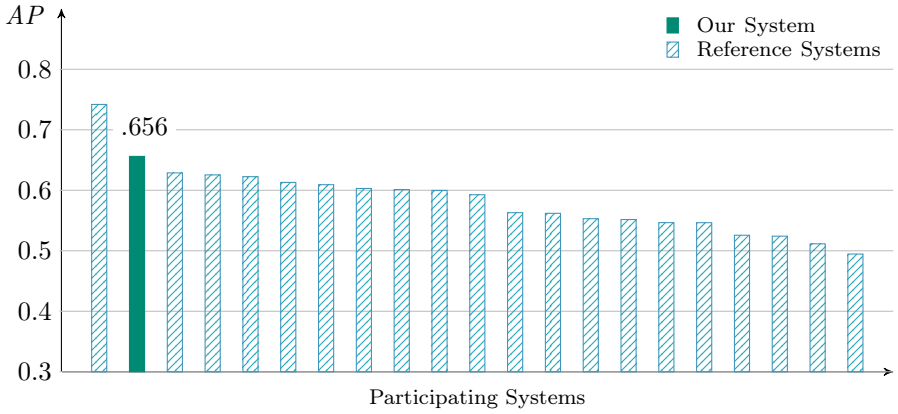


Figure 8.12: Results on the RTE4 dataset (binary classification) in terms of *average precision*. Our system (shown in opaque green) is ranked #2 out of 21 systems in total.[21]

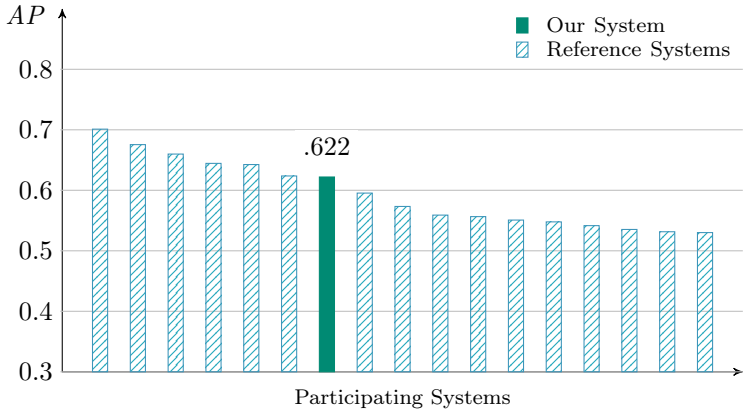


Figure 8.13: Results on the RTE5 dataset (binary classification) in terms of *average precision*. Our system (shown in opaque green) is ranked #7 out of 17 systems in total.[21]

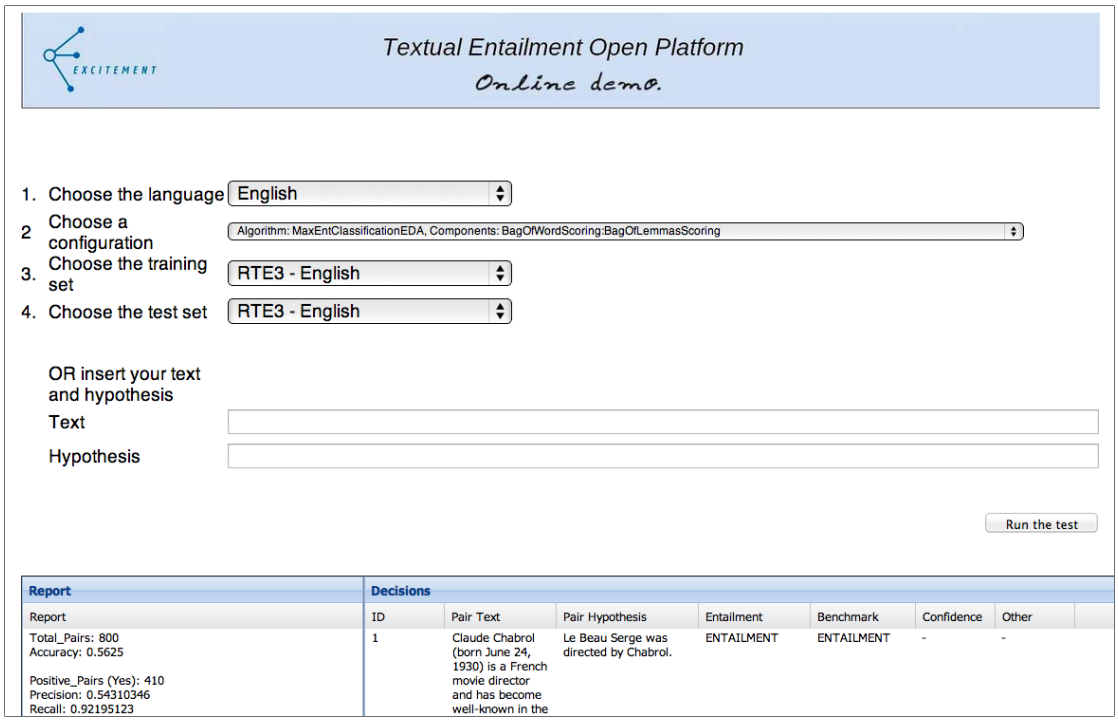


Figure 8.14: Demonstration system of the *Excitement* textual entailment platform: In this screenshot, the entailment decision algorithm *MaxEntClassificationEDA* was used to classify the text pairs of the RTE3 dataset.[22]

System Integration

We believe that the above insights can be very beneficial for future work on textual entailment recognition. Employing text similarity measures in the first phase of a textual entailment system can reduce the number of potentially positive entailment pairs, before then applying sophisticated textual entailment components.

As a first step towards such a combined system, we integrated our open source text similarity framework *DKPro Similarity* (see Chapter 9) with the EU-funded *Excitement* project.[23] *Excitement* is a novel textual entailment platform which is intended to provide a generic tool for textual inference to both the scientific community and enterprise developers. The project will soon be released publicly.[24] We show a screenshot of the current demonstration system in Figure 8.14.

The Excitement platform is designed in a way that it can support a multitude of *entailment decision algorithms (EDA)*. In order to integrate DKPro Similarity with Excitement, we thus created two novel algorithms based on our comprehensive set of text similarity measures. We describe our contributions in the following.

Simple EDA The simple decision algorithm takes a single text similarity measure as an input. It computes text similarity with the given measure, and transforms the continuous similarity scores into a binary classification whether an entailment relationship holds or not based on a given threshold.

[22] The demonstration system is available at http://hlt-services4.fbk.eu/eop

[23] Exploring Customer Interactions through Textual Entailment, http://www.excitement-project.eu

[24] The repository is available at http://github.com/hltfbk/Excitement-Open-Platform

Classification EDA The advanced decision algorithm takes a multitude of text similarity measures as an input. It basically resembles our setup which we used for the experiments on the RTE datasets as described above: First, a machine learning classifier is trained on the similarity scores which are computed for the given training data. Then the classifier is applied to the given test data and outputs binary decisions whether an entailment relationship holds or not. For this implementation, we used a Naive Bayes classifier. However, it can be easily adapted to any other classifier from the Weka toolkit (Hall et al., 2009).

By contributing two entailment decision algorithms based on DKPro Similarity to the Excitement platform, we hope to stimulate further research in this field, as we envision that text similarity measures are valuable tools in the process of recognizing textual entailment.

8.3 Chapter Summary

In this chapter, we stressed the importance of text similarity for real-world applications. We first presented the *Self-Organizing Wikis* application scenario which supports wiki users in their everyday tasks of adding, organizing, and finding content. We discussed our *Wikulu* system implementation, which employs a flexible architecture that features a modular plugin mechanism for natural language processing components in general, and text similarity components in particular. We introduced a number of use cases in the context of this application scenario, and highlighted the importance of text similarity techniques for two particular use cases: *duplicate detection* and *link suggestion*. In the second part of this chapter, we reported on two further applications which are not yet part of the Wikulu system, but are closely related and may benefit future versions of our system: *question answering* and *textual entailment recognition*. For question answering, our open source text similarity framework *DKPro Similarity*, which we will dicuss in detail in the following chapter, has been used successfully in these experiments and has helped to better answer questions that are formulated in natural language. For textual entailment recognition, we showed that text similarity is a promising means for further research in this field, and contributed two *entailment decision algorithms* to the novel *Excitement* open source platform for textual inference.

Chapter 9

DKPro Similarity

In this chapter, we present *DKPro Similarity*, a generalized framework for text similarity. The framework resulted from the work on our system which we discussed in Chapter 5 and which we then tailored towards the intrinsic and extrinsic evaluation in Chapters 6 and 7. In the following, we first start with a brief note on the motivations of this framework in Section 9.1 and compare *DKPro Similarity* with a number of related frameworks in Section 9.2. In Section 9.3, we then describe the overall architecture of the framework. In Section 9.4, we briefly list the text similarity measures which are available to date, as well as the DKPro components which allow the integration of similarity computation with any Apache UIMA-based language processing pipeline. In Section 9.5, we introduce a number of experimental setups which are specific instantiations of the proposed composite model introduced in Section 5 and which can be run out-of-the-box.

9.1 Motivation

"We plan to explore building an open source toolkit for integrating and applying diverse linguistic analysis modules to the STS task," Agirre et al. (2012) announced in the task description of the pilot *Semantic Textual Similarity Task* (see Chapter 6), as they realized that only a few generalized similarity frameworks exist at all. We are currently not aware of any designated text similarity framework which goes beyond simple lexical similarity or contains more than a small number of measures, even though related frameworks exist, which we discuss in Section 9.2.

In order to fill this gap, we present *DKPro Similarity*, an open source framework for text similarity. DKPro Similarity is designed to complement DKPro Core[1], a collection of software components for natural language processing based on the Apache UIMA framework (Ferrucci and Lally, 2004). Our goal is 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. The Java implementation is publicly available at Google Code[2] under the Apache Software

[1] http://code.google.com/p/dkpro-core-asl
[2] http://code.google.com/p/dkpro-similarity-asl

License v2 and partly under GNU GPL v3.

Just recently, Fokkens et al. (2013) stressed that a key aspect of research is to *build upon* prior work. However, they point out that it is still an enormous effort to reproduce results previously reported in the literature. This is mostly due to the fact that previous experimental setups are not reported in enough detail in the literature (e.g. due to space limitations), or not made publicly available as source code. Fokkens et al. (2013) investigated, for example, the effects of major experimental steps on the overall system performance, e.g. variations in pre-processing steps or different versions of software libraries and resources used. As they showed, despite enormous efforts it may be impossible to reproduce previously reported results.

To alleviate this shortcoming, DKPro Similarity also comes with a set of full-featured experimental setups which can be run out-of-the-box and be used for future systems to built upon. 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. As Fokkens et al. (2013) put it, "sharing is key to facilitating reuse." We will elaborate on the available experimental setups in Section 9.5.

DKPro Similarity is a mature framework which has been effectively used in a number of tasks such as question answering (Walter et al., 2012), text reuse detection (Bär et al., 2012b), and textual entailment recognition (see Section 8.2.2). DKPro Similarity also participated in the *Semantic Textual Similarity Task* at SemEval-2012 (Agirre et al., 2012), where a combination of the available text similarity measures has shown the best performance among all participating systems (see Chapter 6). In follow-up work in the second exercise of this series (Agirre et al., 2013), which we described in Section 6.8, DKPro Similarity served as a baseline system which was officially recommended to all task participants.[3] A number of participants reported that they were either inspired by our system, or directly built upon it for the generation of text similarity features or as a basis for the whole text similarity system. We refer the reader to Section 6.8 for further details.

9.2 Related Frameworks

In the following, we discuss related frameworks and give insights where DKPro Similarity uses implementations of the existing libraries. That way, DKPro Similarity brings together the scattered efforts by offering access to all measures through common interfaces. It goes far beyond the functionality of the original libraries as it generalizes the resources used, allows a tight integration with any UIMA-based pipeline, and comes with full-featured experimental setups which are pre-configured stand-alone text similarity systems that can be run out-of-the-box.

S-Space Package Even though not a designated text similarity library, the S-Space Package (Jurgens and Stevens, 2010)[4] contains some text similarity measures such as Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (see Section 3.2). However, it is primarily focused on word space models which operate on

[3] http://www-nlp.stanford.edu/wiki/STS

[4] http://github.com/fozziethebeat/S-Space

word distributions in text. Besides such algorithms, it offers a variety of interfaces, data structures, evaluation datasets and metrics, and global operation utilities e.g. for dimension reduction using Singular Value Decomposition or randomized projections, which are particularly useful with such distributional word space models. DKPro Similarity integrates LSA based on the S-Space Package.

Semantic Vectors The Semantic Vectors package is another package for distributional semantics (Widdows and Cohen, 2010).[5] It contains text similarity measures such as LSA and allows for comparing documents within a given vector space. The main focus lies on word space models with a number of dimension reduction techniques, and applications on word spaces such as automatic thesaurus generation.

WordNet::Similarity The open source package by Pedersen et al. (2004)[6] is a popular Perl library for the similarity computation on WordNet. It comprises six word similarity measures which operate on WordNet, e.g. *Jiang and Conrath (1997)* or *Resnik (1995)*. Unfortunately, though, no strategies have been added to the package yet which aggregate the word similarity scores for complete texts in a similar manner as described in Section 3.1. In DKPro Similarity, we offer native Java implementations of all measures contained in WordNet::Similarity, and allow to go beyond WordNet and use the measures with any lexical-semantic resource of choice, e.g. Wiktionary or Wikipedia.

SimMetrics Library The Java library by Chapman et al. (2005)[7] exclusively comprises text similarity measures which compute lexical similarity on string sequences and compare texts without any semantic processing. It contains measures such as the *Levenshtein (1966)* or *Monge and Elkan (1997)* distance metrics. In DKPro Similarity, some string-based measures (see Section 3.2) are based on implementations from this library.

SecondString Toolkit The freely available library by Cohen et al. (2003a)[8] is similar to SimMetrics, and also implemented in Java. It also contains several well-known text similarity measures on string sequences, and includes many of the measures which are also part of the SimMetrics Library. Some string-based measures in DKPro Similarity are based on the SecondString Toolkit.

SEMILAR Toolkit The package by Rus et al. (2013) comprises both a graphical user interface as well as a developer toolkit for text similarity. The toolkit offers a rich set of diverse text similarity measures such as ones operating on string sequences, the aggregation measure by Mihalcea et al. (2006) which operates on a number of word similarity measures, or Latent Semantic Analysis. In our opinion, though, the focus of this package rather lies on its graphical user interface. This allows to manage

[5] http://code.google.com/p/semanticvectors
[6] http://sourceforge.net/projects/wn-similarity
[7] http://sourceforge.net/projects/simmetrics
[8] http://sourceforge.net/projects/secondstring

different projects, inspect the training and test data along with its annotations, and allows to run text similarity measures in a convenient end-user setting.

9.3 Architecture

DKPro Similarity is designed to operate in either 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 (Ferrucci and Lally, 2004). That way, it allows performing any number of languge processing steps, e.g. coreference or named-entitiy 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 m = new GreedyStringTiling();
2 double similarity = m.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*.[9] By using the common `TextSimilarityMeasure` interface, it is easy to replace Greedy String Tiling with any measure of choice, such as *Latent Semantic Analysis* (Landauer et al., 1998) or *Explicit Semantic Analysis* (Gabrilovich and Markovitch, 2007). We give an overview of measures available in DKPro Similarity in Section 9.4.

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 9.1, we show a graphical overview of the integration of text similarity measures (right) with a UIMA-based pipeline (left). The pipeline starts by reading

[9]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.

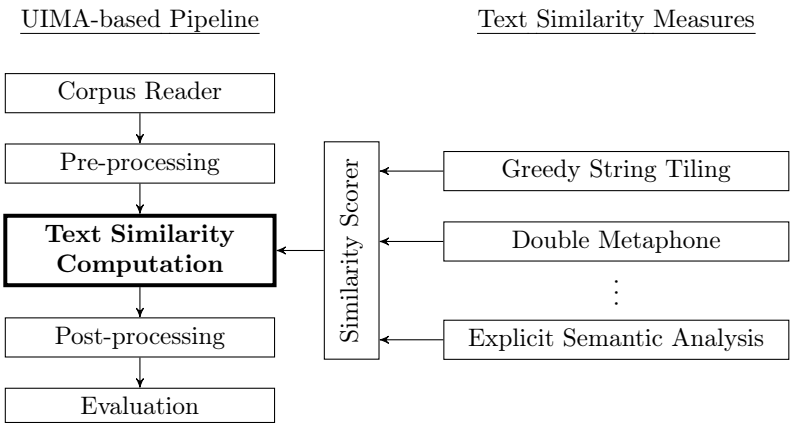


Figure 9.1: DKPro Similarity allows to integrate any text similarity measure (right) which conforms to standardized interfaces into a UIMA-based language processing pipeline (left) by means of a dedicated *Similarity Scorer* component (middle).

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 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:[10]

```
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. We detail the new set of components offered by DKPro Similarity in the following section.

9.4 Measures & Components

A large number of measures presented in Section 3 are already available in DKPro Similarity. We give an overview of the available measures in Table 9.1. Among many others, popular measures implemented in DKPro Similarity include compositional

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

Table 9.1
Overview of the text similarity measures which are available in DKPro Similarity. We also report the corresponding section were each measure has been described in detail.

Dimension	Text Similarity Measure	Section
Content	*Compositional Measures*	
	Word Similarity Aggregation by Mihalcea et al. (2006)	3.1.1 on page 19
	String distance measures	
	Greedy String Tiling (Wise, 1996)	3.2.1 on page 22
	Jaro distance (Jaro, 1989)	3.2.1 on page 22
	Jaro-Winkler distance (Winkler, 1990)	3.2.1 on page 22
	Levenshtein distance (Levenshtein, 1966)	3.2.1 on page 22
	Longest common subsequence (Allison and Dix, 1986)	3.2.1 on page 22
	Longest common substring (Gusfield, 1997)	3.2.1 on page 22
	Monge Elkan distance (Monge and Elkan, 1997)	3.2.1 on page 22
	n-gram Models	
	Character n-gram profiles (Keselj et al., 2003)	3.2.2 on page 24
	Word n-gram profiles (Lyon et al., 2001)	3.2.2 on page 24
	Vector Space Models	
	Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007)	3.2.5 on page 25
	General Vector Space Model (Salton and McGill, 1983)	3.2.3 on page 24
	Latent Semantic Analysis (Landauer et al., 1998)	3.2.4 on page 25
	Lexical Substitution	
	Lexical Substitution Wrapper (Biemann, 2012a,b)	6.3 on page 50
	Microsoft Bing Statistical Machine Translator Wrapper[11]	6.3 on page 50
Structure	Part-of-speech n-grams	5.3.2 on page 42
	Stopword n-grams (Stamatatos, 2011)	5.3.2 on page 42
	Word Pair Distance (Hatzivassiloglou et al., 1999)	5.3.2 on page 42
	Word Pair Order (Hatzivassiloglou et al., 1999)	5.3.2 on page 42
Style	Average Token/Sentence Length	5.3.3 on page 43
	Function Word Frequencies (Dinu and Popescu, 2009)	5.3.3 on page 43
	Sequential Type-Token Ratio (McCarthy and Jarvis, 2010)	5.3.3 on page 43
	Token/Sentence Length Ratio	5.3.3 on page 43
	Type-Token Ratio (Templin, 1957)	5.3.3 on page 43

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). In Chapters 6 and 7, we already reported further details on the employed text similarity measures.

In addition, DKPro Similarity includes components which allow the integration of text similarity measures with any UIMA-based pipeline, as outlined in Figure 9.1.

[11]Similar to the statistical machine translation system employed in the intrinsic evaluation (see Section 6.3), we provide a text similarity wrapper based on the Microsoft Bing translator (`http://www.bing.com/translator`) which performs a round-trip translation on the given input texts before passing on the transformed texts to a particular text similarity measure.

In the following, we introduce these components along with their resources.

Readers & Datasets DKPro Similarity includes corpus readers specifically tailored towards combining the input texts in a number of ways, e.g. all possible combinations, or each text paired with n others by random. Standard datasets for which readers come pre-packaged include, among others, the SemEval-2012 STS data (see Chapter 6), the METER corpus (see Chapter 7), or the RTE 1–5 data (Dagan et al., 2006). As far as license terms allow redistribution, the datasets themselves are integrated into the framework as well and are readily available as Maven modules.

Similarity Scorer The *Similarity Scorer* allows to integrate any text similarity measure (which is decoupled from UIMA by default) into a UIMA-based pipeline. It builds upon the standardized text similarity interfaces and thus allows to easily exchange the similarity measure as well as to specify the data types the measure should operate on, e.g. tokens or lemmas.

Machine Learning In previous research, we have shown that a combination of text similarity measures greatly benefits the system performance due to the fact that different measures capture different text characteristics (Bär et al., 2012b). DKPro Similarity thus provides adapters for Weka's machine learning algorithms (Hall et al., 2009) and allows to first pre-compute sets of text similarity scores which can then be used as features for various machine learning classifiers.

Evaluation Metrics In the final step of a UIMA pipeline, the processed data is read by a dedicated evaluation component. DKPro Similarity ships with a set of components which for example compute Pearson and Spearman correlation with human judgments, or apply task-specific evaluation metrics such as average precision and other confidence-weighted scores as used, for example, in the RTE challenges (Dagan et al., 2006).

9.5 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 9.1 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.[12]

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

DKPro Similarity contains two major types of experimental setups: (i) those for an *intrinsic evaluation* allow to evaluate the system performance in an isolated setting by comparing the system results with a human gold standard, and (ii) those for an *extrinsic evaluation* allow to evaluate the system with respect to a particular task at hand, where text similarity is a means for solving a particular problem, e.g. recognizing textual entailment.

Intrinsic Evaluation DKPro Similarity contains the setup (Bär et al., 2012a) which participated in the *Semantic Textual Similarity Task* at SemEval-2012 (Agirre et al., 2012) and which has now become one of the official baseline systems for the second task of this series.[13] The system uses a simple log-linear regression model to combine multiple text similarity measures of varying complexity. The provided setup allows to evaluate how well the system output resembles human similarity judgments on short texts which are taken from five different sources, e.g. paraphrases of news texts. We report details on the intrinsic evaluation in Chapter 6.

Extrinsic Evaluation DKPro Similarity includes two setups for an extrinsic evaluation: detecting text reuse, and recognizing textual entailment.

For detecting text reuse (Clough et al., 2002), the experimental setup we provide (Bär et al., 2012b) combines a multitude of text similarity measures along different text characteristics. Thereby, it makes extensive use of measures along structural and stylistic text characteristics. Across three standard evaluation datasets, the system consistently outperforms all previous work. We already reported details on the extrinsic evaluation in Chapter 7.

For recognizing textual entailment, we provide a setup which is similar in configuration to the one described above, but contains corpus readers and evaluation components precisely tailored towards the RTE challenge series (Dagan et al., 2006). We believe that our setup can be used for filtering those text pairs which need further analysis by a dedicated textual entailment system. We already reported details on the provided setup in Section 8.2.2.

9.6 Chapter Summary

In this chapter, we described how our work culminated in DKPro Similarity, an open source framework designed to streamline the development of text similarity measures. All measures conform to standardized interfaces and can either be used as stand-alone components in any experimental setup, or can be tightly coupled with a full-featured UIMA-based language processing pipeline in order to allow for advanced processing capabilities.

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

[13] In 2013, the Semantic Textual Similarity Task is a shared task of the Second Joint Conference on Lexical and Computational Semantics, http://ixa2.si.ehu.es/sts

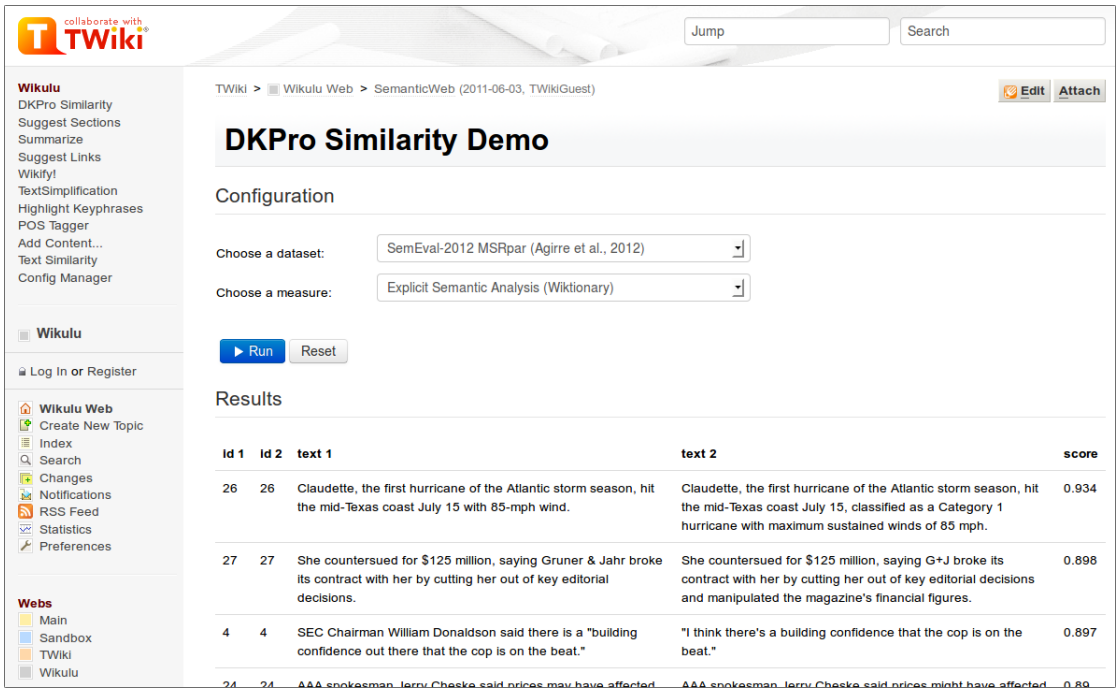


Figure 9.2: Screenshot of the *DKPro Similarity* system demonstration: Text similarity was computed for the *MSRpar* dataset taken from the SemEval-2012 exercises (see Section 6.2) with the *Explicit Semantic Analysis* measure on Wiktionary (see Section 3.2.5). The resulting similarity scores are shown along with the text pairs at the bottom of the screenshot.

getting started, which guides first-time users through both the stand-alone and the UIMA-coupled modes.

For the same reason, we also created a system demonstration for DKPro Similarity which we presented at the *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)* conference in Sofia, Bulgaria, in August 2013 (Bär et al., 2013). A screenshot of the system demonstration is shown in Figure 9.2. We implemented the demonstration as part of the Wikulu platform which we introduced in Section 8.1. In the demonstration, the user can easily choose a standard dataset to operate on, e.g. a dataset used in the intrinsic evaluation as part of the Semantic Textual Similarity Task at SemEval-2012 (see Chapter 6), or enter two custom texts. The user then selects one of the available text similarity measures contained in DKPro Similarity and activates the system. The similarity computation is then invoked and the results are directly written back to the originating web page for manual inspection. We hope that this system demonstration also lowers the entry barrier to DKPro Similarity, and also helps to quickly inspect or compare the performance of different text similarity measures for given text pairs.

Chapter 10

Summary and Conclusions

In the natural language processing community, *text similarity* is fundamental to a variety of tasks and applications such as automatic essay grading or paraphrase recognition. 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 the 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 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. We thereby build upon the idea of *conceptual spaces*—a theory which allows to model aspects of the external world in a generic framework. 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 composite model. As we showed, a system based on this model consistently outperformed prior work and competing systems in both an intrinsic and an extrinsic evaluation, which we summarize in the following.

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)* workshop. 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.

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 dataset stylistic

measures showed poor performance and in consequence the best performance was reached when combining only measures from the *content* and the *structure* dimensions. Across all datasets, our system outperformed all previous work on this data.

In our opinion, the presented research is a fundamental step towards a more well-defined notion of *text similarity*. The central point of our discussion is that *text similarity* cannot be seen as a fixed, axomatic notion. Rather, we need to define—per task—*in what way* two texts should be considered *similar*. In this work, we identified the text dimensions *content*, *structure*, and *style* which may be a promising starting point for future studies, even though other dimensions may be more suitable, depending on the concrete task and application at hand. We believe that our work will benefit any other task where text similarity computation is fundamental. For example, *plagiarism detection* is traditionally limited to the comparison of features along the *content* dimension. However, we see great potential for improvement by including, for example, measures for grammar analysis, lexical complexity, or measures assessing text organization with respect to the discourse elements. However, each task exhibits particular characteristics which influence the choice of a suitable set of text dimensions. A particular dimension may or may not contribute to an overall improvement based on the nature of the data.

Resulting from the insight that the community still lacks a generalized framework for text similarity, our efforts culminated in the open source text similarity framework *DKPro Similarity*. While some frameworks for similarity computation already exist, e.g. the *S-Space Package* or the *SimMetrics Library*, none of them goes beyond simple lexical similarity or contains more than a small number of measures. We thus introduced 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 also hope to stimulate further research in this field as well as future developments of text similarity measures and experimental setups.

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 similarity judgments—both for judgments by human subjects as well as by similarity measures.

Agirre et al. (2013) also point out that there is a need of "more nuanced" datasets that can be used for evaluation. Thereby, they refer to datasets which include, for example, modality, polarity, or sentiment. In the process of annotating evaluation datasets with human judgments, human annotators are typically asked to rate the

degree of similarity between two given texts. However, in this setting it remains completely unclear how the annotators are influenced by the aforementioned linguistic phenomena. An idea for future datasets is to accompany text pairs not only with a single text similarity score, but with a set of scores along different text dimensions. These dimensions could then go beyond *content*, *structure*, and *style* and include dimensions for sentiment or modality. In consequence, text similarity measures could then be precicely evaluated against one or more of these dimensions.

We further believe that future work on text similarity may greatly benefit from the inclusion of more elaborate textual features. Up to now, we did not address any features which explicitly express, for example, semantic relations between the texts such as *elaboration* or *contradiction* as proposed by the *Cross-document Structure Theory* (Zhang et al., 2003). We believe that the inclusion of such features may better allow similarity measures to compute text similarity in the same way that humans do.

Bibliography

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.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 Shared Task: Semantic Textual Similarity. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 32–43, Atlanta, GA, USA.

Allison, L. and Dix, T. I. (1986). A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23:305–310.

Anderka, M. and Stein, B. (2009). The ESA Retrieval Model Revisited. In *Proceedings of the 32th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 670–671, Boston, MA, USA.

Artstein, R. and Poesio, M. (2008). Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

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

Banea, C., Hassan, S., Mohler, M., and Mihalcea, R. (2012). UNT: A Supervised Synergistic Approach to 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 635–642, Montreal, Canada.

Bär, D., Biemann, C., Gurevych, I., and Zesch, T. (2012a). UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada.

Bär, D., Erbs, N., Zesch, T., and Gurevych, I. (2011a). Wikulu: An Extensible Architecture for Integrating Natural Language Processing Techniques with Wikis.

In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. System Demonstrations*, pages 74–79, Portland, OR, USA.

Bär, D., Zesch, T., and Gurevych, I. (2011b). 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. (2012b). 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, Sofia, Bulgaria.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. (2006). The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the 2nd Pascal Challenges Workshop on Recognizing Textual Entailment*, Venice, Italy.

Barrón-Cedeño, A., Rosso, P., Agirre, E., and Labaka, G. (2010). Plagiarism Detection across Distant Language Pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45, Beijing, China.

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.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. (2009). The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the 2nd Text Analysis Conference*, Gaithersburg, MD, USA.

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.

Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). DBpedia - A Crystallization Point for the Web of Data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165.

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.

Brants, T. and Franz, A. (2006). Web 1T 5-gram corpus version 1.1. Technical report, Google Research.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, Brisbane, Australia.

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.

Buffa, M. (2006). Intranet Wikis. In *Proceedings of the IntraWebs Workshop, in conjunction with the 15th International World Wide Web Conference*, Edinburgh, Scotland.

Burrows, S., Potthast, M., and Stein, B. (2013). Paraphrase Acquisition via Crowdsourcing and Machine Learning. *Transactions on Intelligent Systems and Technology*, 4(3):43:1–21.

Buscaldi, D., Roux, J. L., Flores, J. J. G., and Popescu, A. (2013). LIPN-CORE: Semantic Text Similarity using n-grams, WordNet, Syntactic Analysis, ESA and Information Retrieval based Features. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 162–168, Atlanta, GA, USA.

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.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-)Evaluation of Machine Translation. In *Proceedings of the 2nd Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2008). Further Meta-Evaluation of Machine Translation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation*, pages 70–106, Columbus, OH, USA.

Carenini, G. and Cheung, J. C. K. (2008). Extractive vs. NLG-based Abstractive Summarization of Evaluative Text: The Effect of Corpus Controversiality. In *Proceedings of the 5th International Natural Language Generation Conference*, pages 33–41, Salt Fork, OH, USA.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

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.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27.

Chapman, S., Norton, B., and Ciravegna, F. (2005). Armadillo : Integrating Knowledge for the Semantic Web. In *Proceedings of the Dagstuhl Seminar in Machine Learning for the Semantic Web*, Dagstuhl Castle, Germany.

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.

Choi, F. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 26–33, Stroudsburg, PA, 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.

Cohen, W. W., Ravikumar, P., and Fienberg, S. (2003a). A Comparison of String Metrics for Matching Names and Records. In *Proceedings of the KDD Workshop on Data Cleaning and Object Consolidation*, Washington, DC, USA.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003b). A Comparison of String Distance Metrics for Name-Matching Tasks. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, pages 73–78, Acapulco, Mexico.

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.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, pages 168–175.

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.

Decock, L. and Douven, I. (2010). Similarity After Goodman. *Review of Philosophy and Psychology*, pages 1–15.

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.

Erkan, G. and Radev, D. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

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.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Fokkens, A., van Erp, M., Postma, M., Pedersen, T., Vossen, P., and Freire, N. (2013). Offspring from Reproduction Problems: What Replication Failure Teaches Us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1691–1701, Sofia, Bulgaria.

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.

Geva, S. (2007). GPX: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia. In *Proceedings of the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 404–416, Dagstuhl Castle, Germany.

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E., and Dolan, B. (2008). The Fourth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the 1st Text Analysis Conference*, Gaithersburg, MD, USA.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, Czech Republic.

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

Gurevych, I., Müller, C., and Zesch, T. (2007). What to be? Electronic Career Guidance Based on Semantic Relatedness. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 1032–1039, Prague, Czech Republic.

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.

Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 44–52, Atlanta, GA, USA.

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.

Heilman, M. and Madnani, N. (2012). ETS: Discriminative Edit Models for Paraphrase Scoring. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 529–535, Montreal, Canada.

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.

Hoffart, J., Bär, D., Zesch, T., and Gurevych, I. (2009a). Discovering Links Using Semantic Relatedness. In *Proceedings of the 8th Workshop of the Initiative of the Evaluation of XML Retrieval*, pages 314–325, Brisbane, Australia.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61.

Hoffart, J., Zesch, T., and Gurevych, I. (2009b). An Architecture to Support Intelligent User Interfaces for Wikis by Means of Natural Language Processing. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, Orlando, FL, USA.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 57–60, New York, NY, USA.

Huang, W. C. D., Geva, S., and Trotman, A. (2010). Overview of the INEX 2009 Link the Wiki Track. In *Proceedings of the 8th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 312–323, Brisbane, Australia.

Hughes, T. and Ramage, D. (2007). Lexical semantic relatedness with random graph walks. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 581–589, Prague, Czech Republic.

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.

Itakura, K. Y. and Clarke, C. L. (2007). University of Waterloo at INEX2007: Adhoc and Link-the-Wiki Tracks. In *Proceedings of the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 417–425, Dagstuhl Castle, Germany.

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.

Jurgens, D. and Stevens, K. (2010). The S-Space Package: An Open Source Package for Word Space Models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 30–35, Uppsala, Sweden.

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.

Krötzsch, M., Vrandecic, D., and Völkel, M. (2006). Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference*, pages 935–942, Athens, GA, USA.

Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present Day American English*. Brown University Press.

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

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

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.

Leech, G. (1993). 100 million words of English. *English Today*, 9(1):9–15.

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.

Leuf, B. and Cunningham, W. (2001). *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional.

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.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

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.

Marsi, E., Moen, H., Bungum, L., Sizov, G., Gambäck, B., and Lynum, A. (2013). NTNU-CORE: Combining strong features for semantic similarity. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 66–73, Atlanta, GA, USA.

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.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain.

Miller, G. and Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

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.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). WordNet::Similarity – Measuring the Relatedness of Concepts. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: Demonstration Papers*, pages 38–41, Boston, MA, USA.

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 *Proceedings of the 1st International Conference of the Cross-Language Evaluation Forum: Evaluation Labs and Workshops*, Padua, Italy.

Potthast, M., Gollub, T., Hagen, M., Graßegger, 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 *Proceedings of the 3rd International Conference of the Cross-Language Evaluation Forum: Evaluation Labs and Workshops*, 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.

Rus, V., Lintean, M., Banjade, R., Niraula, N., and Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 163–168, Sofia, Bulgaria.

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.

Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly*, 19(3):321–325.

Severyn, A., Nicosia, M., and Moschitti, A. (2013). iKernels-Core: Tree Kernel Learning for Textual Similarity. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 53–58, Atlanta, GA, USA.

Shepard, R. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140.

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.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, Banff, Canada.

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.

Tucker, S. and Whittaker, S. (2009). Have A Say Over What You See: Evaluating Interactive Compression Techniques. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 37–46, Sanibel Island, FL, USA.

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.

Widdows, D. and Cohen, T. (2010). The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. In *4th IEEE International Conference on Semantic Computing*, pages 9–15, Pittsburgh, PA, 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.

Witte, R. and Gitzinger, T. (2007). Connecting wikis and natural language processing systems. In *Proceedings of the International Symposium on Wikis*, pages 165–176, Montreal, Canada.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254–255, Berkeley, CA, USA.

Wu, S., Zhu, D., Carterette, B., and Liu, H. (2013). MayoClinicNLP–CORE: Semantic representations for textual similarity. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 148–154, Atlanta, GA, 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.

Xu, J. and Croft, W. B. (1996). Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, Zurich, Switzerland.

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.

Zesch, T., Müller, C., and Gurevych, I. (2008). Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 861–867, Chicago, IL, USA.

Zhang, Z., Otterbacher, J., and Radev, D. (2003). Learning Cross-document Structural Relationships Using Boosting. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 124–130, New Orleans, LA, USA.

Zhu, T. T. and Lan, M. (2013). ECNUCS: Measuring Short Text Semantic Equivalence Using Multiple Similarity Measurements. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 124–131, Atlanta, GA, USA.

Appendix A

Human Annotations

In Chapter 4, we introduced common evaluation datasets for both an intrinsic and an extrinsic evaluation. In this appendix, we report the detailed annotation results for a number of re-rating studies which we conducted. In Appendix A.1, we report details on the studies for the intrinsic evaluation datasets, and in Appendix A.2 on those for the extrinsic evaluation datasets.

A.1 Intrinsic Evaluation

We conducted two re-rating studies for the evaluation datasets *30 Sentence Pairs* (Li et al., 2006) and *50 Short Texts* (Lee et al., 2005) which we introduced in Section 4.1.1. Our goal was to evaluate whether the human judgments that come with the datasets are stable across time and subjects. In the following, we therefore report the detailed annotation results for both datasets and compare them with the original text similarity scores.[1]

A.1.1 30 Sentence Pairs

We asked 10 human subjects per text pair in a crowdsourcing setting[2] "How close do these sentences come to meaning the same thing?", which was the same question as in the original study by Li et al. (2006). In Table A.1, we present the original text similarity scores as reported by Li et al. (2006) along with the averaged ratings by our subjects. We report both scores in a $[0, 5]$ interval where 0 means *not similar at all* and 5 is *perfectly similar*. For better comparison, we scaled the original scores from a $[0, 4]$ to the $[0, 5]$ interval. From the high correlation (Spearman's rank correlation $\rho = 0.91$), we conclude that the human judgments are indeed stable across time and subjects. Further details are reported in Section 4.1.1.

[1] Download available at http://www.ukp.tu-darmstadt.de/data/text-similarity

[2] Amazon Mechanical Turk (http://mturk.com) via CrowdFlower (http://crowdflower.com)

Table A.1
Human annotations of the *30 Sentence Pairs* dataset compared with the original text similarity scores reported by Li et al. (2006)

ID	Humans	Li et al. (2006)	ID	Humans	Li et al. (2006)
1	1.67	0.05	51	3.70	0.69
5	1.60	0.03	52	4.18	2.43
9	1.82	0.03	53	4.60	2.41
13	2.67	0.54	54	4.33	1.80
17	2.73	0.24	55	3.50	2.03
21	2.55	0.21	56	4.55	2.94
25	2.00	0.33	57	4.30	3.14
29	2.08	0.06	58	4.08	2.95
33	2.70	0.73	59	4.58	4.31
37	2.20	0.65	60	4.50	2.90
41	2.58	1.41	61	3.75	2.61
47	3.82	1.74	62	4.90	3.86
48	4.36	1.78	63	4.82	2.79
49	3.00	1.46	64	5.00	4.78
50	3.92	2.35	65	4.73	3.26

A.1.2 50 Short Texts

In this study, we selected a subset of 50 text pairs from the dataset by Lee et al. (2005). We selected the pairs in a way that they have a uniform distribution of judgments across the full similarity range. We then asked 3 human annotators to rate "How similar are the given texts?". In Table A.2, we report the scores originally reported by Lee et al. (2005) as well as the ratings by the three subjects, both within a $[1, 5]$ interval where 1 is *not similar at all* and 5 is *perfectly similar*. Again, the resulting Spearman correlation between the aggregated results of the annotators and the original scores is high ($\rho = 0.88$) and thus shows that judgments are stable across time and subjects. Further details on the *50 Short Texts* dataset are reported in Section 4.1.1.

Table A.2
Human annotations of a subset of the *50 Short Texts* dataset compared with the original text similarity scores reported by Lee et al. (2005)

ID 1	ID 2	Lee et al. (2005)	Rater 1	Rater 2	Rater 3
1	6	2.50	1	2	2
1	14	5.00	4	4	4
1	32	2.40	1	4	1
1	33	4.80	4	3	4
2	32	1.50	1	2	1
2	45	1.10	1	2	1
3	16	3.40	1	2	1
4	8	3.90	2	4	4

Table A.2
Human annotations of a subset of the *50 Short Texts* dataset compared with the original text similarity scores reported by Lee et al. (2005) *(continued)*

ID 1	ID 2	Lee et al. (2005)	Rater 1	Rater 2	Rater 3
4	10	3.10	2	2	3
4	12	3.60	2	4	3
4	14	1.30	1	1	1
4	16	3.70	2	3	3
4	31	1.60	1	1	1
4	36	4.00	1	3	4
4	44	1.90	1	2	1
5	42	3.80	3	5	2
6	39	2.00	1	4	1
7	24	2.44	1	1	1
7	27	2.70	2	1	1
7	31	1.00	1	1	1
7	36	3.90	2	3	4
8	11	3.10	1	3	3
8	21	4.70	5	4	3
8	46	2.20	1	2	1
9	47	4.60	3	4	2
10	18	1.82	1	1	1
10	28	2.82	1	4	2
11	21	3.50	1	3	2
11	42	4.10	4	2	5
12	16	4.25	3	5	4
12	36	3.60	2	2	5
14	33	4.90	3	4	4
14	45	1.90	1	3	2
18	20	3.33	2	5	3
18	41	1.27	1	2	1
20	37	4.50	3	4	4
21	27	3.00	1	2	1
21	36	3.20	1	2	3
22	37	1.50	1	1	1
23	36	2.90	1	1	1
24	36	2.64	1	1	2
25	26	5.00	5	4	5
26	29	2.10	1	2	1
30	31	2.33	1	3	1
32	45	4.22	2	3	3
32	50	4.60	4	4	4
37	46	1.73	1	2	1
40	43	4.33	2	2	3
40	46	1.44	1	1	1
42	44	2.90	2	2	2

A.2 Extrinsic Evaluation

The Wikipedia Rewrite Corpus (Clough and Stevenson, 2011) and the METER Corpus (Gaizauskas et al., 2001), which we used for an extrinsic evaluation, contain text pairs along with human classifications of the degree of text reuse (see Chapter 7). As each text of these datasets was written by only a single person for a given text reuse category, we conducted an annotation study in which we were mostly interested in the inter-rater agreement of the subjects. As reported in Section 7.4, we asked 3 participants to rate the degree of text reuse and provided them with the original annotation guidelines. In the following, we present the detailed results of the annotation studies.[3] Further analyses and the chance-corrected inter-rater agreements can be found in Section 7.4.

A.2.1 Wikipedia Rewrite Corpus

The rewritten texts that are contained in the Wikipedia Rewrite Corpus are answers to 5 questions about topics of computer science (see Section 4.2.1 for details). The texts are named according to these questions (ID ending in `task`n, $n = a, \ldots, e$) and were paired with the original question (named `orig_task`n in the original dataset). The possible rewrite labels are *no plagiarism* (non), *heavy revision* (heavy), *light revision* (light), and *cut & paste* (cut). The original classification is reported in the column *Gold*. The results of this study show that the annotators mostly disagree for the *light* and *heavy revision* classes. We report details on the analysis in Section 7.4.1.

Table A.3
Human annotations of the Wikipedia Rewrite Corpus compared with the original classifications reported by Clough and Stevenson (2011)

ID	Gold	Rater 1	Rater 2	Rater 3
g0pA_taska	non	non	non	non
g0pA_taskb	cut	cut	cut	cut
g0pA_taskc	light	heavy	light	light
g0pA_taskd	heavy	non	heavy	heavy
g0pA_taske	non	non	non	non
g0pB_taska	non	non	non	non
g0pB_taskb	non	non	non	non
g0pB_taskc	cut	non	heavy	light
g0pB_taskd	light	light	non	light
g0pB_taske	heavy	heavy	heavy	heavy
g0pC_taska	heavy	non	heavy	heavy
g0pC_taskb	non	non	non	non
g0pC_taskc	non	non	non	non
g0pC_taskd	cut	cut	cut	light
g0pC_taske	light	light	light	light
g0pD_taska	cut	cut	heavy	cut
g0pD_taskb	light	light	light	heavy

[3]Download available at http://www.ukp.tu-darmstadt.de/data/text-similarity

Table A.3
Human annotations of the Wikipedia Rewrite Corpus compared with the original classifications reported by Clough and Stevenson (2011) *(continued)*

ID	Gold	Rater 1	Rater 2	Rater 3
g0pD_taskc	heavy	heavy	heavy	heavy
g0pD_taskd	non	non	non	non
g0pD_taske	non	non	non	non
g0pE_taska	light	cut	cut	cut
g0pE_taskb	heavy	non	cut	cut
g0pE_taskc	non	non	non	non
g0pE_taskd	non	non	non	non
g0pE_taske	cut	cut	cut	cut
g1pA_taska	non	non	non	non
g1pA_taskb	heavy	non	heavy	heavy
g1pA_taskc	light	non	light	heavy
g1pA_taskd	cut	heavy	non	cut
g1pA_taske	non	non	non	non
g1pB_taska	non	non	non	non
g1pB_taskb	non	non	non	non
g1pB_taskc	heavy	heavy	light	heavy
g1pB_taskd	light	non	non	heavy
g1pB_taske	cut	non	light	cut
g1pD_taska	light	heavy	heavy	heavy
g1pD_taskb	cut	non	cut	light
g1pD_taskc	non	non	non	non
g1pD_taskd	non	non	non	non
g1pD_taske	heavy	non	non	heavy
g2pA_taska	non	non	non	non
g2pA_taskb	heavy	heavy	non	light
g2pA_taskc	light	light	cut	light
g2pA_taskd	cut	cut	heavy	non
g2pA_taske	non	non	non	non
g2pB_taska	non	non	non	non
g2pB_taskb	non	non	non	heavy
g2pB_taskc	heavy	non	heavy	heavy
g2pB_taskd	light	non	cut	light
g2pB_taske	cut	cut	cut	cut
g2pC_taska	cut	non	cut	cut
g2pC_taskb	non	non	non	non
g2pC_taskc	non	non	non	non
g2pC_taskd	heavy	non	heavy	cut
g2pC_taske	light	non	heavy	heavy
g2pE_taska	heavy	heavy	heavy	heavy
g2pE_taskb	light	light	light	light
g2pE_taskc	cut	non	non	non
g2pE_taskd	non	non	non	non
g2pE_taske	non	non	non	non
g3pA_taska	non	non	heavy	non

Table A.3
Human annotations of the Wikipedia Rewrite Corpus compared with the original classifications reported by Clough and Stevenson (2011) *(continued)*

ID	Gold	Rater 1	Rater 2	Rater 3
g3pA_taskb	heavy	light	heavy	light
g3pA_taskc	light	light	non	light
g3pA_taskd	cut	cut	light	cut
g3pA_taske	non	non	heavy	heavy
g3pB_taska	non	non	non	heavy
g3pB_taskb	non	non	non	non
g3pB_taskc	heavy	non	non	heavy
g3pB_taskd	light	non	light	light
g3pB_taske	cut	light	cut	cut
g3pC_taska	cut	cut	cut	cut
g3pC_taskb	non	non	non	non
g3pC_taskc	non	non	non	non
g3pC_taskd	heavy	heavy	heavy	light
g3pC_taske	light	light	light	light
g4pB_taska	non	non	non	non
g4pB_taskb	non	non	non	non
g4pB_taskc	heavy	non	cut	cut
g4pB_taskd	light	heavy	cut	light
g4pB_taske	cut	cut	cut	cut
g4pC_taska	cut	cut	cut	cut
g4pC_taskb	non	non	non	non
g4pC_taskc	non	non	non	non
g4pC_taskd	heavy	light	light	cut
g4pC_taske	light	light	light	cut
g4pD_taska	light	heavy	light	non
g4pD_taskb	cut	non	non	non
g4pD_taskc	non	non	non	non
g4pD_taskd	non	non	non	non
g4pD_taske	heavy	heavy	heavy	light
g4pE_taska	heavy	non	heavy	non
g4pE_taskb	light	cut	heavy	cut
g4pE_taskc	cut	non	cut	cut
g4pE_taskd	non	non	non	non
g4pE_taske	non	heavy	non	non

A.2.2 METER Corpus

Several of the texts contained in the METER Corpus have more than a single UK Press Association source where it is unclear which (if not all) of the source stories have been used to generate the rewritten story. However, as discussed in Section 4.2.1, 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 which we used for evaluation. The texts were paired with their respective single source story and annotated by 3 subjects in our annotation study. In the following, we present the results of this study. We use a binary classification into *reused* (R) and *non-reused* (N) texts. The original classification is reported in column G. The results show that for 61% of all texts the annotators fully agree. Details on the results on this dataset can be found in Section 7.4.2.

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001)

ID	G	R_1	R_2	R_3
showbiz/27.09.99/millenium/millenium116_mirror	R	R	R	N
courts/12.08.99/traveller/traveller188_telegraph	N	N	N	N
showbiz/14.12.99/panto/panto168_star	R	N	N	N
showbiz/12.08.99/blackadder/blackadder71_mail	R	N	N	N
showbiz/07.01.00/depp/depp146_star	R	N	N	N
courts/02.11.99/wight/wight287_times	R	R	R	R
showbiz/12.08.99/blackadder/blackadder66_star	N	N	N	N
courts/26.01.00/war/war697_independent	R	R	R	R
courts/14.07.99/fowler/fowler203_star	R	R	R	R
showbiz/21.06.00/gallagher/gallagher98_sun	R	N	N	N
showbiz/10.08.99/enfield/enfield50_express	R	N	N	N
showbiz/27.01.00/eastenders/eastenders130_star	R	N	N	N
courts/16.07.99/baby/baby85_star	N	R	N	N
courts/22.11.99/rubbish/rubbish623_star	R	R	R	R
showbiz/08.08.99/shaw/shaw10_mail	N	R	R	N
showbiz/12.08.99/blackadder/blackadder73_express	N	N	N	N
showbiz/08.08.99/shaw/shaw6_star	R	R	R	R
courts/12.08.99/granny/granny190_telegraph	R	N	N	R
showbiz/12.08.99/blackadder/blackadder60_sun	R	N	N	N
courts/17.04.00/informant/informant755_times	R	R	N	R
courts/17.04.00/nursery/nursery757_telegraph	N	R	N	N
showbiz/14.12.99/united/united172_times	R	N	R	N
courts/02.11.99/chips/chips274_mirror	N	N	R	N
courts/12.07.99/walker/walker9_mirror	R	R	N	N
showbiz/07.01.00/oasis/oasis148_star	R	N	N	N
showbiz/17.04.00/holden/holden111_mirror	R	R	R	R
showbiz/21.06.00/winslet/winslet107_guardian	R	R	R	R
courts/21.02.00/home/home512_star	N	R	N	N
courts/08.12.99/chief/chief372_independent	R	R	R	R
courts/28.09.99/head/head588_guardian	R	R	R	R
courts/12.08.99/traveller/traveller185_times	N	N	N	N
courts/08.12.99/chief/chief349_times	R	R	R	R
courts/12.07.99/policeman/policeman38_times	R	R	R	R
courts/04.10.99/raider/raider266_independent	R	R	R	R
showbiz/09.08.99/stretch/stretch25_star	N	N	N	R
courts/16.07.99/fan/fan80_star	R	R	N	R

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001) *(continued)*

ID	G	R_1	R_2	R_3
showbiz/12.08.99/spice/spice62_sun	R	R	R	N
courts/17.04.00/farmer/farmer758_telegraph	N	N	R	N
courts/22.11.99/babies/babies620_mirror	R	N	R	N
courts/09.08.99/royal/royal152_star	N	R	N	N
courts/09.08.99/porn/porn162_telegraph	R	R	R	N
courts/07.10.99/punch/punch248_times	R	R	R	R
courts/26.01.00/driver/driver668_sun	R	R	R	R
courts/15.03.00/baby/baby411_guardian	R	R	R	N
courts/01.03.00/francis/francis386_mail	R	R	N	N
courts/17.04.00/nursery/nursery749_mail	N	N	N	R
courts/22.11.99/salesman/salesman637_times	R	R	R	R
courts/12.08.99/traveller/traveller171_sun	R	N	N	R
courts/01.03.00/francis/francis393_guardian	R	N	N	N
showbiz/27.09.99/beegees/beegees119_times	R	R	R	R
courts/16.07.99/torture/torture83_star	R	R	R	R
courts/21.02.00/mp/mp527_independent	R	R	R	R
showbiz/12.08.99/spice/spice68_star	N	N	N	N
courts/02.11.99/chips/chips295_telegraph	N	R	N	N
showbiz/07.01.00/street/street149_star	R	R	R	R
courts/27.01.00/drink/drink716_times	R	R	R	R
showbiz/17.04.00/holden/holden112_star	R	R	R	R
courts/18.02.00/stakeout/stakeout550_telegraph	R	R	R	R
courts/26.01.00/bahamas/bahamas701_times	N	N	N	N
courts/12.07.99/policeman/policeman47_telegraph	R	N	R	N
courts/22.11.99/rubbish/rubbish657_independent	R	R	R	R
courts/04.10.99/dj/dj265_independent	R	R	R	R
courts/14.12.99/boy/boy420_star	N	N	N	N
showbiz/10.08.99/enfield/enfield35_sun	R	R	N	N
courts/01.03.00/wagstaff/wagstaff391_telegraph	R	R	R	R
courts/03.04.00/harass/harass486_times	R	R	R	N
courts/03.04.00/husband/husband497_guardian	R	R	R	R
courts/21.02.00/home/home506_sun	N	N	N	N
courts/16.07.99/banker/banker125_telegraph	R	R	R	R
courts/22.11.99/pool/pool611_sun	R	R	R	N
courts/27.01.00/damages/damages732_sun	R	R	N	N
courts/15.03.00/shooting/shooting404_mail	N	N	N	N
courts/18.02.00/abuse/abuse551_telegraph	N	N	R	R
courts/27.01.00/drink/drink728_independent	R	R	R	R
showbiz/14.12.99/united/united166_mirror	R	N	R	N
courts/17.04.00/informant/informant766_guardian	R	R	R	R
courts/09.08.99/porn/porn145_sun	R	R	N	R
courts/18.02.00/cars/cars530_mirror	N	N	N	N
showbiz/14.12.99/panto/panto163_sun	R	R	N	N
showbiz/22.11.99/christmas/christmas161_independent	R	R	N	N

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001) *(continued)*

ID	G	R_1	R_2	R_3
courts/12.07.99/lovers/lovers55_guardian	R	R	R	R
courts/07.10.99/hamilton/hamilton251_telegraph	R	R	N	N
courts/27.01.00/drink/drink708_mirror	R	R	R	R
courts/14.12.99/boy/boy422_mail	N	N	N	N
showbiz/09.08.99/casualty/casualty21_mirror	R	R	R	N
courts/12.07.99/police/police53_guardian	R	R	R	R
courts/12.08.99/traveller/traveller181_express	R	R	R	R
courts/27.09.99/pinochetcharge/pinochetcharge609_guardian	N	R	N	R
courts/01.03.00/francis/francis378_mirror	R	N	N	N
courts/16.07.99/transplant/transplant97_express	N	N	N	N
courts/21.02.00/mp/mp525_guardian	R	R	R	R
showbiz/13.08.99/spice/spice80_star	R	R	N	R
courts/17.04.00/vanessa/vanessa756_times	R	R	R	R
courts/12.07.99/walker/walker48_telegraph	R	R	R	N
courts/09.08.99/royal/royal159_times	R	R	R	R
courts/17.04.00/nursery/nursery765_guardian	R	R	R	R
showbiz/22.11.99/cilla/cilla157_mirror	R	N	N	N
courts/03.04.00/fraud/fraud496_guardian	R	R	R	N
showbiz/17.04.00/holden/holden109_sun	R	N	N	N
courts/01.03.00/wagstaff/wagstaff376_sun	R	N	N	N
courts/22.11.99/father/father644_telegraph	R	N	N	N
showbiz/07.01.00/posh/posh145_star	R	N	N	N
showbiz/12.08.99/street/street72_express	N	R	N	N
courts/07.10.99/knickers/knickers257_telegraph	R	R	R	R
courts/09.08.99/royal/royal146_sun	R	R	R	R
courts/16.07.99/transplant/transplant82_star	R	N	N	N
courts/27.09.99/pinochetcharge/pinochetcharge608_telegraph	N	N	R	R
courts/26.01.00/driver/driver675_mail	N	N	R	N
showbiz/13.08.99/elton/elton78_star	R	R	R	R
showbiz/07.01.00/street/street150_express	R	N	R	R
courts/02.11.99/chips/chips277_star	N	R	R	R
showbiz/12.08.99/blackadder/blackadder63_mirror	N	N	N	N
courts/07.01.00/soldier/soldier566_telegraph	R	R	R	R
courts/01.03.00/francis/francis385_express	R	N	R	R
courts/02.11.99/chips/chips271_sun	N	N	N	N
courts/07.01.00/soldier/soldier562_times	N	R	N	N
courts/17.04.00/vanessa/vanessa760_telegraph	R	R	N	N
courts/17.04.00/nursery/nursery753_times	N	N	N	R
showbiz/22.11.99/daytime/daytime154_sun	R	R	R	R
showbiz/10.08.99/aled/aled38_mirror	R	R	R	R
showbiz/09.08.99/stretch/stretch27_mail	N	N	N	N
showbiz/21.06.00/beckham/beckham106_times	R	N	R	N
courts/12.08.99/traveller/traveller195_guardian	R	R	R	R
courts/09.08.99/royal/royal148_mirror	R	R	N	R

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001) *(continued)*

ID	G	R_1	R_2	R_3
showbiz/08.08.99/shaw/shaw1_sun	R	R	R	R
courts/21.02.00/home/home522_telegraph	N	N	R	N
courts/01.03.00/francis/francis394_guardian	R	N	N	N
courts/21.06.00/broadmoor/broadmoor787_times	R	R	R	N
courts/17.04.00/vanessa/vanessa771_independent	R	R	R	R
courts/18.02.00/abuse/abuse537_express	N	R	R	N
showbiz/10.08.99/enfield/enfield48_mail	N	N	R	R
showbiz/09.08.99/stretch/stretch22_mirror	N	N	R	R
courts/02.11.99/chips/chips301_guardian	N	R	R	R
courts/28.09.99/head/head593_independent	R	R	R	R
courts/02.11.99/chips/chips288_times	N	R	R	N
courts/21.06.00/blast/blast781_express	R	R	R	N
courts/01.03.00/francis/francis380_star	R	N	N	N
courts/17.04.00/farmer/farmer763_guardian	R	R	R	N
courts/27.01.00/drink/drink720_telegraph	R	R	R	R
courts/03.04.00/fraud/fraud480_times	R	R	R	R
showbiz/27.01.00/webber/webber136_telegraph	R	R	N	R
showbiz/13.08.99/spicefestival/spicefestival79_star	R	R	R	R
courts/04.10.99/raider/raider253_telegraph	R	R	R	N
courts/18.02.00/presenter/presenter547_telegraph	R	R	R	R
courts/29.03.00/lavin/lavin435_sun	R	R	N	R
showbiz/21.06.00/madeley/madeley100_mirror	R	N	N	R
courts/17.04.00/nursery/nursery740_star	N	N	N	R
courts/04.10.99/zoo/zoo240_times	R	R	R	R
courts/21.02.00/sewage/sewage518_times	N	R	R	N
courts/29.03.00/ruling/ruling451_telegraph	N	R	N	N
courts/26.01.00/driver/driver679_times	R	R	R	R
courts/07.10.99/hamilton/hamilton259_guardian	R	R	R	R
courts/07.01.00/soldier/soldier559_mirror	N	N	N	N
showbiz/21.06.00/madeley/madeley99_sun	R	N	R	N
showbiz/09.08.99/stretch/stretch18_sun	N	R	N	N
courts/07.10.99/knickers/knickers234_star	R	R	R	R
showbiz/07.01.00/depp/depp143_sun	R	N	N	N
courts/27.01.00/drink/drink726_guardian	R	R	R	R
showbiz/10.08.99/enfield/enfield53_times	N	N	N	N
courts/21.02.00/sewage/sewage523_telegraph	R	R	R	R
courts/17.04.00/nursery/nursery745_express	R	R	R	R
showbiz/27.09.99/lottery/lottery115_sun	R	R	R	N
showbiz/12.08.99/street/street67_star	N	N	N	N
courts/17.04.00/informant/informant769_independent	R	R	R	R
showbiz/12.08.99/street/street64_mirror	N	N	N	N
courts/12.07.99/lovers/lovers6_sun	R	R	R	R
courts/12.07.99/lovers/lovers22_star	R	R	R	R
courts/07.10.99/hamilton/hamilton256_telegraph	N	R	N	N

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001) *(continued)*

ID	G	R_1	R_2	R_3
courts/12.07.99/walker/walker30_express	R	R	R	R
showbiz/14.12.99/christmas/christmas167_star	R	N	N	N
courts/16.07.99/transplant/transplant116_telegraph	R	R	R	R
courts/04.10.99/raider/raider227_star	R	N	R	R
courts/14.12.99/boy/boy427_telegraph	N	R	N	N
courts/03.04.00/harass/harass492_telegraph	R	R	R	R
courts/17.04.00/nursery/nursery768_independent	R	R	R	R
courts/18.02.00/cars/cars545_telegraph	N	N	R	N
showbiz/13.08.99/spice/spice77_sun	N	N	N	N
showbiz/07.01.00/oasis/oasis142_sun	R	N	R	N
showbiz/09.08.99/saints/saints20_sun	R	R	R	R
showbiz/14.12.99/christmas/christmas164_sun	R	R	N	N
courts/04.10.99/doctor/doctor233_express	N	N	N	N
showbiz/12.08.99/blackadder/blackadder76_independent	N	N	N	N
courts/27.01.00/drink/drink704_sun	R	R	R	N
courts/17.04.00/vanessa/vanessa767_guardian	R	R	R	R
courts/18.02.00/cab/cab548_telegraph	N	R	N	N
showbiz/21.06.00/beckham/beckham108_independent	R	N	N	N
courts/29.03.00/ruling/ruling447_times	N	N	N	N
courts/14.07.99/lara/lara222_telegraph	R	R	R	R
courts/28.09.99/brothel/brothel572_star	R	R	R	R
courts/21.06.00/blast/blast777_mirror	N	N	N	N
courts/15.03.00/baby/baby400_mirror	R	R	R	R
showbiz/27.01.00/lamarr/lamarr123_sun	R	R	R	N
courts/15.03.00/baby/baby413_independent	R	N	R	N
courts/14.12.99/inqbryant/inqbryant418_mirror	N	N	N	N
courts/12.08.99/traveller/traveller179_mail	N	R	N	N
courts/01.03.00/francis/francis390_telegraph	R	R	R	R
courts/16.07.99/baby/baby118_telegraph	N	N	R	N
courts/09.08.99/bennett/bennett167_telegraph	R	R	R	N
courts/14.07.99/lara/lara223_guardian	R	R	R	R
courts/14.12.99/boy/boy428_guardian	R	R	R	R
courts/17.04.00/farmer/farmer739_mirror	N	R	R	N
courts/27.01.00/damages/damages722_telegraph	R	R	R	R
showbiz/14.12.99/united/united178_independent	R	N	R	N
courts/12.07.99/silverstone/silverstone37_times	R	R	N	N
showbiz/21.06.00/fairbrass/fairbrass103_star	R	R	R	N
showbiz/21.06.00/madeley/madeley102_star	R	R	R	R
showbiz/07.01.00/street/street141_sun	R	R	R	N
showbiz/12.08.99/street/street59_sun	N	N	N	R
courts/21.02.00/sewage/sewage509_mirror	N	N	N	N
courts/21.02.00/sewage/sewage515_express	N	N	R	N
courts/26.01.00/blood/blood681_times	R	R	N	R
showbiz/21.06.00/steps/steps101_mirror	R	N	R	N

Table A.4
Human annotations of the METER Corpus compared with the original classifications reported by Gaizauskas et al. (2001) *(continued)*

ID	G	R_1	R_2	R_3
courts/08.12.99/city/city373_independent	R	R	R	R
courts/01.03.00/wagstaff/wagstaff384_express	R	R	R	R
courts/02.11.99/chips/chips306_independent	N	R	R	R
courts/09.08.99/royal/royal160_telegraph	R	R	R	N
courts/18.02.00/abuse/abuse555_independent	R	R	R	R
showbiz/21.06.00/steps/steps105_star	R	R	R	R
courts/17.04.00/vanessa/vanessa750_mail	R	R	R	R
courts/14.12.99/boy/boy430_independent	R	R	R	R
courts/01.03.00/football/football382_star	R	R	R	N
courts/01.03.00/wagstaff/wagstaff381_star	R	R	R	R
courts/02.11.99/chips/chips282_mail	N	R	R	R
courts/02.11.99/chips/chips286_express	N	R	R	R
courts/18.02.00/cars/cars535_star	N	N	N	N
courts/17.04.00/nursery/nursery737_mirror	N	N	N	N
courts/16.07.99/torture/torture72_mirror	R	R	R	R
showbiz/27.09.99/beegees/beegees120_telegraph	R	R	R	R
courts/01.03.00/francis/francis375_sun	R	N	N	N
showbiz/12.08.99/blackadder/blackadder74_guardian	R	R	R	R
courts/17.04.00/informant/informant762_telegraph	N	N	R	N
courts/28.09.99/brothel/brothel580_times	R	R	R	R
showbiz/09.08.99/saints/saints23_mirror	R	R	R	R
courts/14.12.99/australia/australia417_mirror	N	N	N	N
showbiz/12.08.99/spice/spice65_mirror	R	R	R	R
courts/12.08.99/granny/granny186_times	R	R	N	N
courts/04.10.99/pensioners/pensioners245_telegraph	R	R	R	R
courts/09.08.99/camera/camera157_times	R	R	R	R
showbiz/27.09.99/spice/spice118_star	R	N	N	N
showbiz/12.08.99/street/street70_mail	N	N	N	N
courts/12.07.99/strangle/strangle18_star	R	R	R	R
courts/12.07.99/policeman/policeman16_star	R	R	R	R
showbiz/14.07.99/thomas/thomas86_sun	R	R	N	N
courts/17.04.00/farmer/farmer743_star	R	N	R	N
showbiz/21.06.00/gallagher/gallagher97_sun	R	N	R	N
showbiz/27.09.99/spicemel/spicemel117_star	R	N	N	N
courts/01.03.00/francis/francis388_times	R	R	N	N
courts/04.10.99/pensioners/pensioners236_express	R	R	R	R
courts/29.03.00/ruling/ruling456_independent	N	N	N	N
courts/07.01.00/giggs/giggs560_mirror	R	R	R	R
courts/01.03.00/diver/diver396_guardian	R	R	R	R
courts/12.08.99/traveller/traveller176_star	R	R	N	N
showbiz/07.01.00/street/street153_telegraph	R	R	R	R