UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures

Daniel Bär[†], Chris Biemann[†], Iryna Gurevych[†‡], and Torsten Zesch[†‡]

[†]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt

[‡]Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for Educational Research and Educational Information

www.ukp.tu-darmstadt.de

Abstract

We present the UKP system which performed best in the Semantic Textual Similarity (STS) task at SemEval-2012 in two out of three metrics. It uses a simple log-linear regression model, trained on the training data, to combine multiple text similarity measures of varying complexity. These range from simple character and word n-grams and common subsequences to complex features such as Explicit Semantic Analysis vector comparisons and aggregation of word similarity based on lexical-semantic resources. Further, we employ a lexical substitution system and statistical machine translation to add additional lexemes, which alleviates lexical gaps. Our final models, one per dataset, consist of a log-linear combination of about 20 features, out of the possible 300+ features implemented.

1 Introduction

The goal of the pilot Semantic Textual Similarity (STS) task at SemEval-2012 is to measure the degree of semantic equivalence between pairs of sentences. STS is fundamental to a variety of tasks and applications such as question answering (Lin and Pantel, 2001), text reuse detection (Clough et al., 2002) or automatic essay grading (Attali and Burstein, 2006). STS is also closely related to textual entailment (TE) (Dagan et al., 2006) and paraphrase recognition (Dolan et al., 2004). It differs from both tasks, though, insofar as those operate on binary similarity decisions while STS is defined as a graded notion of similarity. STS further requires a bidirectional similarity relationship to hold between a pair of sentences rather than a unidirectional entailment relation as for the TE task.

A multitude of measures for computing similarity between texts have been proposed in the past based on surface-level and/or semantic content features (Mihalcea et al., 2006; Landauer et al., 1998; Gabrilovich and Markovitch, 2007). The existing measures exhibit two major limitations, though: Firstly, measures are typically used in separation. Thereby, the assumption is made that a single measure inherently captures all text characteristics which are necessary for computing similarity. Secondly, existing measures typically exclude similarity features beyond *content* per se, thereby implying that similarity can be computed by comparing text content exclusively, leaving out any other text characteristics. While we can only briefly tackle the second issue here, we explicitly address the first one by combining several measures using a supervised machine learning approach. With this, we hope to take advantage of the different facets and intuitions that are captured in the single measures.

In the following section, we describe the feature space in detail. Section 3 describes the machine learning setup. After describing our submitted runs, we discuss the results and conclude.

2 Text Similarity Measures

We now describe the various features we have tried, also listing features that did not prove useful.

2.1 Simple String-based Measures

String Similarity Measures These measures operate on string sequences. The *longest common*

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 435–440,
Montréal, Canada, June 7-8, 2012.

substring measure (Gusfield, 1997) compares the length of the longest contiguous sequence of characters. The *longest common subsequence* measure (Allison and Dix, 1986) drops the contiguity requirement and allows to detect similarity in case of word insertions/deletions. *Greedy String Tiling* (Wise, 1996) allows to deal with reordered text parts as it determines a set of shared contiguous substrings, whereby each substring is a match of maximal length. We further used the following measures, which, however, did not make it into the final models, since they were subsumed by the other measures: *Jaro (1989)*, *Jaro-Winkler* (Winkler, 1990), *Monge and Elkan (1997)*, and *Levenshtein (1966)*.

Character/word n-grams We compare character n-grams following the implementation by Barrón-Cedeño et al. (2010), thereby generalizing the original trigram variant to $n = 2, 3, \ldots, 15$. We also compare word n-grams using the Jaccard coefficient as previously done by Lyon et al. (2001), and the containment measure (Broder, 1997). As high n led to instabilities of the classifier due to their high intercorrelation, only $n = 1, 2, 3, 4$ was used.

2.2 Semantic Similarity Measures

Pairwise Word Similarity The measures for computing word similarity on a semantic level operate on a graph-based representation of words and the semantic relations among them within a lexical-semantic resource. For this system, we used the algorithms by Jiang and Conrath (1997), Lin (1998a), and Resnik (1995) on WordNet (Fellbaum, 1998).

In order to scale the resulting pairwise word similarities to the text level, we applied the aggregation strategy by Mihalcea et al. (2006): The sum of the *idf*-weighted similarity scores of each word with the best-matching counterpart in the other text is computed in both directions, then averaged. In our experiments, the measure by Resnik (1995) proved to be superior to the other measures and was used in all word similarity settings throughout this paper.

Explicit Semantic Analysis We also used the vector space model *Explicit Semantic Analysis* (ESA) (Gabrilovich and Markovitch, 2007). Besides WordNet, we used two additional lexical-semantic resources for the construction of the ESA vector space: Wikipedia and Wiktionary[1].

Textual Entailment We experimented with using the BIUTEE textual entailment system (Stern and Dagan, 2011) for generating entailment scores to serve as features for the classifier. However, these features were not selected by the classifier.

Distributional Thesaurus We used similarities from a Distributional Thesaurus (similar to Lin (1998b)) computed on 10M dependency-parsed sentences of English newswire as a source for pairwise word similarity, one additional feature per POS tag. However, only the feature based on cardinal numbers (CD) was selected in the final models.

2.3 Text Expansion Mechanisms

Lexical Substitution System We used the lexical substitution system based on supervised word sense disambiguation (Biemann, 2012). This system automatically provides substitutions for a set of about 1,000 frequent English nouns with high precision. For each covered noun, we added the substitutions to the text and computed the pairwise word similarity for the texts as described above. This feature alleviates the lexical gap for a subset of words.

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. Thereby, the idea was that in the translation process additional lexemes are introduced which alleviate potential lexical gaps. The system was trained on Europarl made available by Koehn (2005), using the following configuration which was not optimized for this task: WMT11[2] baseline without tuning, with MGIZA alignment. The largest improvement was reached for computing pairwise word similarity (as described above) on the concatenation of the original text and the three back-translations.

2.4 Measures Related to Structure and Style

In our system, we also used measures which go beyond content and capture similarity along the structure and style dimensions inherent to texts. However, as we report later on, for this content-

[1] www.wiktionary.org

[2] 0-5-grams, grow-diag-final-and alignment, msd-bidirectional-fe reodering, interpolation and kndiscount

oriented task they were not selected by the classifier. Nonetheless, we briefly list them for completeness.

Structural similarity between texts can be detected by computing **stopword n-grams** (Stamatatos, 2011). Thereby, all content-bearing words are removed while stopwords are preserved. Stopword n-grams of both texts are compared using the containment measure (Broder, 1997). In our experiments, we tested n-gram sizes for $n = 2, 3, \ldots, 10$.

We also compute **part-of-speech n-grams** for various POS tags which we then compare using the containment measure and the Jaccard coefficient.

We also used two similarity measures between pairs of words (Hatzivassiloglou et al., 1999): **Word pair order** tells whether two words occur in the same order in both texts (with any number of words in between), **word pair distance** counts the number of words which lie between those of a given pair.

To compare texts along the stylistic dimension, we further use a **function word frequencies** measure (Dinu and Popescu, 2009) which operates on a set of 70 function words identified by Mosteller and Wallace (1964). Function word frequency vectors are computed and compared by Pearson correlation.

We also include a number of measures which capture statistical properties of texts, such as **type-token ratio** (TTR) (Templin, 1957) and **sequential TTR** (McCarthy and Jarvis, 2010).

3 System Description

We first run each of the similarity measures introduced above separately. We then use the resulting scores as features for a machine learning classifier.

Pre-processing Our system is based on DKPro[3], a collection of software components for natural language processing built upon the Apache UIMA framework. During the pre-processing phase, we tokenize the input texts and lemmatize using the TreeTagger implementation (Schmid, 1994). For some measures, we additionally apply a stopword filter.

Feature Generation We now compute similarity scores for the input texts with all measures and for all configurations introduced in Section 2. This resulted in 300+ individual score vectors which served as features for the following step.

[3] http://dkpro-core-asl.googlecode.com

Run	Features
1	Greedy String Tiling Longest common subsequence (2 normalizations) Longest common substring Character 2-, 3-, and 4-grams 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) Word Similarity (Resnik (1995) on WordNet aggregated according to Mihalcea et al. (2006); 2 variants: complete texts + difference only) Explicit Semantic Analysis (Wikipedia, Wiktionary) Distributional Thesaurus (POS: Cardinal numbers)
2	All Features of Run 1 Lexical Substitution for Word Sim. (complete texts) SMT for Word Sim. (complete texts as above)
3	All Features of Run 2 Random numbers from $[4.5, 5]$ for surprise datasets

Table 1: Feature sets of our three system configurations

Feature Combination The feature combination step uses the pre-computed similarity scores, and combines their $\log$-transformed values using a linear regression classifier from the WEKA toolkit (Hall et al., 2009). We trained the classifier on the training datasets of the STS task. During the development cycle, we evaluated using 10-fold cross-validation.

Post-processing For Runs 2 and 3, we applied a post-processing filter which stripped all characters off the texts which are not in the character range [a-zA-Z0-9]. If the texts match, we set their similarity score to 5.0 regardless of the classifier's output.

4 Submitted Runs

Run 1 During the development cycle, we identified 19 features (see Table 1) which achieved the best performance on the training data. For each of the known datasets, we trained a separate classifier and applied it to the test data. For the surprise datasets, we trained the classifier on a joint dataset of all known training datasets.

Run 2 For the Run 2, we were interested in the effects of two additional features: lexical substitution and statistical machine translation. We added the corresponding measures to the feature set of Run 1 and followed the same evaluation procedure.

Run 3 For the third run, we used the same feature set as for Run 2, but returned random numbers from $[4.5, 5]$ for the sentence pairs in the surprise datasets.

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

Table 2: Best results for single measures, grouped by dimension, on the training datasets *MSRpar*, *MSRvid*, and *SMTeuroparl*, using 10-fold cross-validation

5 Results on Training Data

Evaluation was carried out using the official scorer which computes Pearson correlation of the human rated similarity scores with the the system's output.

In Table 2, we report the results achieved on each of the training datasets using 10-fold cross-validation. The best results were achieved for the feature set of Run 2, with Pearson's $r = .724$, $r = .868$, and $r = .742$ for the datasets *MSRpar*, *MSRvid*, and *SMTeuroparl*, respectively. While individual classes of content similarity measures achieved good results, a different class performed best for each dataset. However, text similarity measures related to structure and style achieved only poor results on the training data. This was to be expected due to the nature of the data, though.

6 Results on Test Data

Besides the Pearson correlation for the union of all datasets (*ALL*), the organizers introduced two additional evaluation metrics after system submission: *ALLnrm* computes Pearson correlation after the system outputs for each dataset are fitted to the gold standard using least squares, and *Mean* refers to the weighted mean across all datasets, where the weight depends on the number of pairs in each dataset.

In Table 3, we report the offical results achieved on the test data. The best configuration of our system was Run 2 which was ranked #1 for the evaluation metrics *ALL* ($r = .823$)[4] and *Mean* ($r = .677$), and #2 for *ALLnrm* ($r = .857$). An exhaustive overview of all participating systems can be found in the STS task description (Agirre et al., 2012).

$\#_1$	$\#_2$	$\#_3$	Sys.	r_1	r_2	r_3	PAR	VID	SE	WN	SN
1	**2**	**1**	**UKP2**	**.823**	**.857**	**.677**	**.683**	**.873**	**.528**	**.664**	**.493**
2	3	5	TL	.813	.856	.660	.698	.862	.361	.704	.468
3	1	2	TL	.813	.863	.675	.734	.880	.477	.679	.398
4	4	4	UKP1	.811	.855	.670	.682	.870	.511	.664	.467
5	6	13	UNT	.784	.844	.616	.535	.875	.420	.671	.403
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
87	85	70	B/L	.311	.673	.435	.433	.299	.454	.586	.390

Table 3: Official results on the test data for the top 5 participating runs out of 89 which were achieved on the known datasets *MSRpar*, *MSRvid*, and *SMTeuroparl*, as well as on the surprise datasets *OnWN* and *SMTnews*. 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). The provided baseline is shown at the bottom of this table.

7 Conclusions and Future Work

In this paper, we presented the UKP system, which performed best across the three official evaluation metrics in the pilot Semantic Textual Similarity (STS) task at SemEval-2012. While we did not reach the highest scores on any of the single datasets, our system was most robust across different data. In future work, it would be interesting to inspect the performance of a system that combines the output of all participating systems in a single linear model.

We also propose that two major issues with the datasets are tackled in future work: (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*)?

Acknowledgements This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Klaus Tschira Foundation under project No. 00.133.2008.

[4] 99% confidence interval: $.807 \leq r \leq .837$

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 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*.

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

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2.0. *Journal of Technology, Learning, and Assessment*, 4(3).

Alberto Barrón-Cedeño, Paolo Rosso, Eneko Agirre, and Gorka Labaka. 2010. Plagiarism Detection across Distant Language Pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45.

Chris Biemann. 2012. Creating a System for Lexical Substitutions from Scratch using Crowdsourcing. *Language Resources and Evaluation: Special Issue on Collaboratively Constructed Language Resources*, 46(2).

Andrei Z. Broder. 1997. On the resemblance and containment of documents. *Proceedings of the Compression and Complexity of Sequences*, pages 21–29.

Paul Clough, Robert Gaizauskas, Scott S.L. Piao, and Yorick Wilks. 2002. METER: MEasuring TExt Reuse. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 152–159.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges*, Lecture Notes in Computer Science, pages 177–190. Springer.

Liviu P. Dinu and Marius Popescu. 2009. Ordinal measures in authorship identification. In *Proceedings of the 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 62–66.

William B. Dolan, Chris Quirk, and Chris Brockett. 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.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 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.

Matthew A. Jaro. 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.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2):259–284.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

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

Dekang Lin. 1998a. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, pages 296–304.

Dekang Lin. 1998b. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 768–774.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 118–125.

Philip M. McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisti-

cated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–92.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780.

Alvaro Monge and Charles Elkan. 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.

Frederick Mosteller and David L. Wallace. 1964. *Inference and disputed authorship: The Federalist*. Addison-Wesley.

Philip Resnik. 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.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Efstathios Stamatatos. 2011. Plagiarism detection using stopword n-grams. *Journal of the American Society for Information Science and Technology*, 62(12):2512–2527.

Asher Stern and Ido Dagan. 2011. A Confidence Model for Syntactically-Motivated Entailment Proofs. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 455–462.

Mildred C. Templin. 1957. *Certain language skills in children*. University of Minnesota Press.

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

Michael J. Wise. 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.