SemantiKLUE: Robust Semantic Similarity at Multiple Levels Using Maximum Weight Matching

Thomas Proisl and **Stefan Evert** and **Paul Greiner** and **Besim Kabashi**
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department Germanistik und Komparatistik
Professur für Korpuslinguistik
Bismarckstr. 6, 91054 Erlangen, Germany
{thomas.proisl,stefan.evert,paul.greiner,besim.kabashi}@fau.de

Abstract

Being able to quantify the semantic similarity between two texts is important for many practical applications. SemantiKLUE combines unsupervised and supervised techniques into a robust system for measuring semantic similarity. At the core of the system is a word-to-word alignment of two texts using a maximum weight matching algorithm. The system participated in three SemEval-2014 shared tasks and the competitive results are evidence for its usability in that broad field of application.

1 Introduction

Semantic similarity measures the semantic equivalence between two texts ranging from total difference to complete semantic equivalence and is usually encoded as a number in a closed interval, e. g. $[0, 5]$. Here is an example for interpreting the numeric similarity scores taken from Agirre et al. (2013, 33):

0. The two sentences are on different topics.
1. The two sentences are not equivalent, but are on the same topic.
2. The two sentences are not equivalent, but share some details.
3. The two sentences are roughly equivalent, but some important information differs/missing.
4. The two sentences are mostly equivalent, but some unimportant details differ.
5. The two sentences are completely equivalent, as they mean the same thing.

Systems capable of reliably predicting the semantic similarity between two texts can be beneficial for a broad range of NLP applications, e. g. paraphrasing, MT evaluation, information extraction, question answering and summarization.

A general system for semantic similarity aiming at being applicable in such a broad scope has to be able to adapt to the use case at hand, because different use cases might, for example, require different similarity scales: For one application, two texts dealing roughly with the same topic should get a high similarity score, whereas for another application being able to distinguish between subtle differences in meaning might be important. The three SemEval-2014 shared tasks focussing on semantic similarity (cf. Sections 3, 4 and 5 for more detailed task descriptions) provide a rich testbed for such a general system, as the individual tasks and subtasks have slightly different objectives.

In the remainder of this paper, we describe SemantiKLUE, a general system for measuring semantic similarity between texts that we built based on our experience from participating in the *SEM 2013 shared task on "Semantic Textual Similarity" (Greiner et al., 2013).

2 System Description

SemantiKLUE operates in two stages. In the first, unsupervised stage, a number of similarity measures are computed. Those measures are the same for all tasks and range from simple heuristics to distributional approaches to resource-heavy methods based on WordNet and dependency structures. The idea is to have a variety of similarity measures that can capture small differences in meaning as well as broad thematical similarities. In the second, supervised stage, all similarity measures obtained in this way are passed to a support vector regression learner that is trained on the available gold standard data in order to obtain a final semantic similarity score. This way, the proper similarity scale for a given task can be learned. The few remaining outliers in the predictions for new text pairs are cut

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 532–540,
Dublin, Ireland, August 23-24 2014.

off to fit the interval required by the task definition ($[0,4]$ or $[0,5]$).

Our submissions for the individual tasks were created using incomplete versions from different developmental stages of the system. In the following sections we describe the current version of the complete system for which we also report comparable results for all tasks (cf. Sections 3– 5).

The whole system is implemented in Python.

2.1 Preprocessing

We use Stanford CoreNLP[1] for part-of-speech tagging, lemmatizing and parsing the input texts. We utilize the CCprocessed variant of the Stanford Dependencies (collapsed dependencies with propagation of conjunct dependencies; de Marneffe and Manning (2008, 13–15)) to create a graph representation of the texts using the NetworkX[2] (Hagberg et al., 2008) module. All the similarity measures described below are computed on the basis of that graph representation. It is important to keep in mind that by basing all computations on the Stanford Dependencies model we effectively ignore most of the prepositions when using measures that work on tokens.[3] For some tasks, we perform some additional task-specific preprocessing steps prior to parsing, cf. task descriptions below.

2.2 Simple Measures

We use four simple heuristic similarity measures that need very little preprocessing. The first two are word form overlap and lemma overlap between the two texts. We take the sets of word form tokens/lemmatized tokens in text A and text B and calculate the Jaccard coefficient:

$$\text{overlap} = \frac{|A \cap B|}{|A \cup B|}.$$

The third is a heuristic for the difference in text length that was used by Gale and Church (1993) as a similarity measure for aligning sentences:

$$z_i = \frac{\mathrm{d}_i}{\sigma_\mathrm{d}}, \text{ where } \mathrm{d}_i = b_i - \frac{\sum_{j=1}^{N} b_j}{\sum_{j=1}^{N} a_j} a_i.$$

For each of the N text pairs we calculate the difference d_i between the observed length of text B and the expected length of text B based on the length of text A. By dividing that difference d_i by the standard deviation of all those differences, we obtain our heuristic z_i.

The fourth is a binary feature expressing whether the two texts differ in their use of negation. We check if one of the texts contains any of the lemmata *no*, *not* or *none* and the other doesn't. That feature is motivated by the comparatively large number of sentences in the SICK dataset (Marelli et al., 2014b) that mainly differ in their use of negation, e. g. sentence pair 42 in the training data that has a gold similarity score of 3.4:

- Two people are kickboxing and spectators are watching
- Two people are kickboxing and spectators are not watching

2.3 Measures Based on Distributional Document Similarity

We obtain document similarity scores from two large-vocabulary distributional semantic models (DSMs).

The first model is based on a 10-billion word Web corpus consisting of Wackypedia and ukWaC (Baroni et al., 2009), UMBC WebBase (Han et al., 2013), and UKCOW 2012 (Schäfer and Bildhauer, 2012). Target terms and feature terms are POS-disambiguated lemmata.[4] We use parameters suggested by recent evaluation experiments: co-occurrence counts in a symmetric 4-word window, the most frequent 30,000 lexical words as features, log-likelihood scores with an additional log-transformation, and SVD dimensionality reduction of L2-normalized vectors to 1000 latent dimensions. This model provides distributional representations for 150,000 POS-disambiguated lemmata as target terms.

The second model was derived from the second release of the Google Books N-Grams database (Lin et al., 2012), using the dependency pairs provided in this version. Target and feature terms are case-folded word forms; co-occurrence ounts are based on direct syntactic relations. Here, the most frequent 50,000 word forms were used as features. All other parameters are identical to the first DSM. This model provides distributional representations for 250,000 word forms.

We compute bag-of-words centroid vectors for each text as suggested by (Schütze, 1998). For each

[1] `http://nlp.stanford.edu/software/corenlp.shtml`

[2] `http://networkx.github.com/`

[3] That is because in the CCprocessed variant of the Stanford Dependencies most prepositions are "collapsed" into dependency relations and are therefore represented as edges and not as vertices in the graph.

[4] e.g. `can_N` for the noun *can*

text pair and DSM, we calculate the cosine similarity between the two centroid vectors as a measure of their semantic similarity. We also determine the number of unknown words in both texts according to both DSMs as additional features.

2.4 Alignment-based Measures

We also use features based on word-level similarity. We separately compute similarities between words using state-of-the-art WordNet similarity measures and the two distributional semantic models described above. The words from both texts are then aligned using those similarity scores to maximize the similarity total. We use two types of alignment: One-to-one alignment where some words in the longer text remain unaligned and one-to-many alignment where all words are aligned. The one-to-many alignment is based on the one-to-one alignment and aligns each previously unaligned word in the longer text to the most similar word in the shorter text. The discussion of the alignment algorithm is based on the former case.

2.4.1 Alignment via Maximum Weight Matching

We opt for a graphical solution to the alignment problem. The similarities between the words from both texts can be modelled as a bipartite graph in which every word from text A is a vertice on the left-hand side of the graph and every word from text B a vertex on the right-hand side. Weighted edges connect every word from text A to every word from text B. The weight of an edge corresponds to the similarity between the two words it connects. In order to obtain an optimal one-to-one alignment we have to select edges in such a way that no two edges share a common vertice and that the sum of the edge weights is maximized. That corresponds to the problem of finding the maximum weight matching in the graph. SemantiKLUE utilizes the NetworkX implementation of Galil's (1986) algorithms for finding that maximum weight matching.

Figure 1 visualizes the one-to-one alignment between two sentences. For the one-to-many alignment, the previously unaligned words are aligned as indicated by the dashed lines.

2.4.2 Measures Based on Distributional Word Similarities

For each of the two DSMs described in Section 2.3 we compute the best one-to-one and the best one-to-many alignment using the cosine similarity between two words as edge weight. For each of those two alignments we compute the following two similarity measures: I) the arithmetic mean of the cosines between all the aligned words from text A and text B and II) the arithmetic mean ignoring identical word pairs.

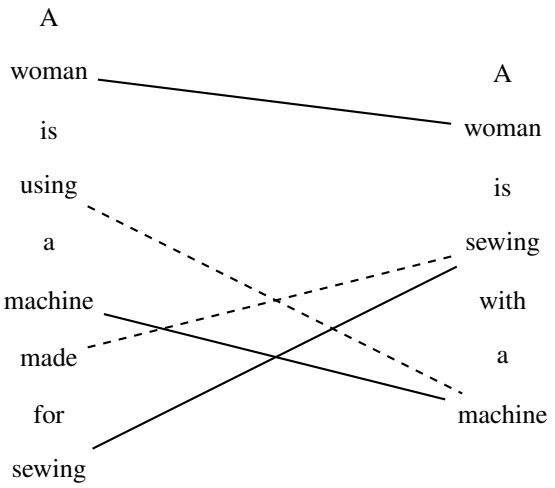

Figure 1: Alignment between a sentence pair from the SICK data set.

In addition to those eight measures, we use the lemma-based DSM for computing the distribution of cosines between lemma pairs. For both alignments, we categorize the cosines between aligned lemma pairs into five heuristically determined intervals ($[0.2, 0.35)$, $[0.35, 0.5)$, $[0.5, 0.7)$, $[0.7, 0.999)$, $[0.999, 1.0]$)[5] and use the proportions as features. Intuitively, the top bins correspond to links between identical words, paradigmatically related words and topically related words. All in all, we use a total of 18 features computed from the DSM-based alignments.

2.4.3 Measures Based on WordNet

We utilize two state-of-the-art (Budanitsky and Hirst, 2006) WordNet similarity measures for creating alignments: Leacock and Chodorow's (1998) normalized path length and Lin's (1998) universal similarity measure. For both of those similarity measures we compute the best one-to-one and the best one-to-many alignment. For each alignment we compute the following two similarity measures: I) the arithmetic mean of the similarities between the aligned words from text A and text B and II) the arithmetic mean ignoring identical word pairs.

[5] Values in the interval $[0.0, 0.2)$ are discarded as they would be collinear with the other features.

We also include the number of unknown words in both texts according to WordNet as additional features.

2.5 Measures Using the Dependency Structure

We expect that the information encoded in the dependency structure of the texts can be beneficial in determining the semantic similarity between them. Therefore, we use three heuristics for measuring similarity on the level of syntactic dependencies. The first simply measures the overlap of dependency relation labels between the two texts (cf. Section 2.2). The second utilizes the fact that the Stanford Dependencies are organized in a hierarchy (de Marneffe and Manning, 2008, 11–12) to compute Leacock and Chodorow's normalized path lengths between individual dependency relations. That measure for the similarity between dependency relations is then used to determine the best one-to-one alignment between dependency relations from text A and text B and to compute the arithmetic mean of the similarities between the aligned dependency relations. The third heuristic gives an indication of the quality of the one-to-one alignment and can be used to distinguish texts that contain the same words in different syntactic structures. It uses the one-to-one alignment created with similarity scores from the lemma-based DSM (cf. Section 2.4.2) to compute the average overlap of neighbors for all aligned word pairs. The overlap of neighbors is determined by computing the Jaccard coefficient of sets N_A and N_B. Set N_A contains all words from text B that are aligned to words from text A that are connected to the target word via a single dependency relation. N_B contains all words from text B that are connected to the word aligned to the target word in text A via a single dependency relation.

2.6 Experimental Features

As an experiment, we included features from a commercial text clustering software that is currently being developed by our team (Greiner and Evert, in preparation). We used this tool – which combines ideas from Latent Semantic Indexing and distributional semantics with multiple clustering steps – as a black box.

We loaded *all* training, development and test items for a given task into the system and applied the clustering algorithm. However, we did not make use of the resulting topic clusters. Instead, we computed cosine similarities for each pair (s_1, s_2) of sentences (or other textual units) based on the internal representation. In addition, we computed the average neighbour rank of the two sentences, based on the rank of s_2 among the nearest neighbours of s_1 and vice versa.

Since these features are generated from the task data themselves, they should adapt automatically to the range of meaning differences present in a given data set.

2.7 Machine Learning

Using all the features described above, we have a total of 39 individual features that measure semantic similarity between two texts (cf. Sections 2.2 to 2.5) and two experimental features (cf. Section 2.6). In order to obtain a single similarity score, we use the scikit-learn[6] (Pedregosa et al., 2011) implementation of support vector regression. In our cross-validation experiments we got the best results with an RBF kernel of degree 2 and a penalty $C = 0.7$, so those are the parameters we use in our experiments.

The SemEval-2014 Task 1 also includes a classification subtask for which we use the same $39 + 2$ features for training a support vector classifier. Cross-validation suggests that the best parameter setting is a polynomial kernel of degree 2 and a penalty $C = 2.5$.

3 SemEval-2014 Task 1

3.1 Task Description

The focus of the shared task on "Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment" (Marelli et al., 2014a) lies on the compositional nature of sentence semantics. By using a specially created data set (Marelli et al., 2014b) that tries to avoid multiword expressions and other idiomatic features of language outside the scope of compositional semantics, it provides a testbed for systems implementing compositional variants of distributional semantics. There is also an additional subtask for detecting the entailment relation (entailment, neutral, contradiction) between to sentences.

Although SemantiKLUE lacks a truly sophisticated component for dealing with compositional semantics (besides trying to incorporate the dependency structure of the texts), the system takes the seventh place in the official ranking by Pearson correlation with a correlation coefficient of 0.780

[6] `http://scikit-learn.org/`

(best of 17 systems: 0.828). In the entailment subtask, the system even takes the fourth place with an accuracy of 0.823 (best of 18 systems: 0.846).

3.2 Experiments

The official runs we submitted for this task were created by a work-in-progress version of SemantiKLUE that did not contain all the features described above. In this section, we report on some post-hoc experiments with the complete system using all the features as well as various subsets of features. See Table 1 for an overview of the results.

Run	r	ρ	MSE	Acc.
primary run	0.780	0.736	0.403	**0.823**
best run	0.782	0.738	0.398	**0.823**
complete system	0.798	0.754	0.373	0.820
no deps	0.793	0.748	0.383	0.817
no deps, no WN	0.763	0.713	0.432	0.793
complete + experimental	**0.801**	**0.757**	**0.367**	**0.823**
only DSM alignment	**0.729**	**0.670**	**0.484**	0.746
only WordNet	0.708	0.636	0.515	0.715
only simple	0.676	0.667	0.561	**0.754**
only DSM document	0.660	0.568	0.585	0.567
only deps	0.576	0.565	0.688	0.614

Table 1: Results for task 1 (Pearson's r, Spearman's ρ, mean squared error and accuracy).

The whole system as described above, without the experimental features, performs even a bit better in the semantic similarity subtask (taking place 6) and only slightly worse in the entailment subtask (still taking place 4) than the official submissions. Adding the experimental features slightly improves the results but does not lead to a better position in the ranking.

We are particularly interested in the impact of the resource-heavy features derived from the dependency structure of the texts and from WordNet. If we use the complete system without the dependency-based features (emulating the case of a language for which we have access to a WordNet-like resource but not to a parser), we get results that are only marginally worse than those for the complete system and lead to the same places in the rankings. Additionally leaving out WordNet has a bigger impact and results in places 9 and 8 in the rankings.

Regarding the individual feature groups, the DSM-alignment-based measures are the best feature group for predicting semantic similarity and the simple heuristic measures are the best feature group for predicting entailment.

4 SemEval-2014 Task 3

4.1 Task Description

Unlike the other tasks, which focus on similar-sized texts, the shared task on "Cross-Level Semantic Similarity" (Jurgens et al., 2014) is about measuring semantic similarity between textual units of different lengths. It comprises four subtasks comparing I) paragraphs to sentences, II) sentences to phrases, III) phrases to words and IV) words to word senses (taken from WordNet). Due to the nature of this task, performance in it might be especially useful as an indicator for the usefulness of a system in the area of summarization.

SemantiKLUE takes the fourth place out of 38 in both the official ranking by Pearson correlation and the alternative ranking by Spearman correlation.

4.2 Additional Preprocessing

For the official run we perform some additional preprocessing on the data for the two subtasks on comparing phrases to words and words to word senses. On the word level we combine the word with the glosses of all its WordNet senses and on the word sense level we replace the WordNet sense indication with its corresponding lemmata and gloss. As our post-hoc experiments show that has a negative effect on performance in the phrase-to-word subtask. Therefore, we skip the additional preprocessing on that level for our experiments described below.

4.3 Experiments

For each of the four subtasks, we perform the same experiments as described in Section 3.2: We compare the official run submitted from a work-in-progress version of SemantiKLUE with the results from the whole system; we see how the system performs without dependency-based features and WordNet-based features; we try out the experimental features; we determine the most important feature group for the subtask. Table 2 gives an overview of the results.

4.3.1 Paragraph to Sentence

Our submitted run takes the fifth place (ties with another system) in the official ranking by Pearson correlation with a correlation coefficient of 0.817 (best of 34 systems: 0.837) and seventh place in the alternative ranking by Spearman correlation.

Run	Paragraph to sent.		Sent. to phrase		Phrase to word		Word to sense	
	r	ρ	r	ρ	r	ρ	r	ρ
official	**0.817**	**0.802**	**0.754**	**0.739**	0.215	0.218	0.314	0.327
complete system	**0.817**	**0.802**	**0.754**	**0.739**	0.284	0.289	0.316	**0.330**
no deps	0.815	**0.802**	0.752	**0.739**	0.309	0.313	0.312	0.329
no deps, no WN	0.813	**0.802**	0.736	0.721	**0.335**	**0.335**	0.234	0.248
complete + experimental	0.816	0.800	0.752	0.738	0.292	0.298	**0.318**	**0.330**
only DSM alignment	0.799	0.789	**0.724**	**0.711**	**0.302**	**0.301**	0.216	0.216
only WordNet	0.787	0.769	0.664	0.641	0.186	0.171	**0.313**	**0.311**
only simple	**0.807**	**0.793**	0.686	0.672	0.128	0.121	0.089	0.093
only DSM document	0.629	0.624	0.546	0.558	0.247	0.240	0.144	0.148
only deps	0.655	0.621	0.449	0.440	0.036	0.057	−0.080	−0.076

Table 2: Results for task 3 (Pearson's r and Spearman's ρ).

The complete SemantiKLUE system gives identical results. Leaving out the resource-heavy features based on the dependency structure and WordNet diminishes the results only very slightly, though it still resolves the tie and puts the system on the sixth place in the Pearson ranking. Adding the experimental features to the complete system has a minor negative effect.

Probably due to the length of the texts, our simple heuristic measures surpass the DSM-alignment-based measures as the best feature group for predicting semantic similarity.

4.3.2 Sentence to Phrase

In this subtask, SemantiKLUE takes the fourth place in both the official ranking with a Pearson correlation coefficient of 0.754 (best of 34 systems: 0.777) and in the alternative ranking by Spearman correlation. The complete system performs identically to our submitted run and leaving out the dependency-based features has little impact on the results. Additionally also leaving out the WordNet-based features has more impact on the results and puts the system on the eighth place in the official ranking. Just as in the paragraph-to-sentence subtask, adding the experimental features to the complete system has a slightly negative effect.

For this subtask, the DSM-alignment-based measures are clearly the feature group that yields the best results.

4.3.3 Phrase to Word

For our submitted run we performed the additional preprocessing described in Section 4.2 resulting in the eleventh place in the official ranking with a Pearson correlation coefficient of 0.215 (best of 22 systems: 0.415) and the 14th place in the alternative ranking by Spearman correlation. For our experiments with the complete system we skip that additional preprocessing step, i. e. we do not add the WordNet glosses to the word, and drastically improve the results, putting our system on the third place in the official ranking. Even more interesting is the observation that leaving out the resource-heavy features further improves the results, putting the system on the second place. In consistency with those observations, the DSM-alignment-based measures are not only the strongest individual feature group but also yield better results when taken alone than the complete system.

In contrast to the first two subtasks, adding the experimental features to the complete systems has a slightly positive effect here.

4.3.4 Word to Sense

In the word-to-sense subtask, SemantiKLUE takes the third place in both the official ranking with a Pearson correlation coefficient of 0.316 (best of 20 systems: 0.381) and in the alternative ranking by Spearman correlation. The complete system performs slightly better than our submitted run and adding the experimental features gives another marginal improvement. Leaving out the dependency-based features has little impact but also leaving out the WordNet-based features severly hurts performance. The reason for that behaviour becomes clear when we look at the results for the individual feature groups: the WordNet-based measures are clearly the strongest feature group for predicting the semantic similarity between words and word senses.

5 SemEval-2014 Task 10

5.1 Task Description

The shared task on "Multilingual Semantic Textual Similarity" (Agirre et al., 2014) is a continuation of the SemEval-2012 and *SEM 2013 shared tasks

Run	deft-forum	deft-news	headlines	images	OnWN	tweet-news	w. mean
best run	0.349	0.643	**0.733**	0.773	0.855	0.640	0.694
complete (all training data)	0.432	0.638	0.660	0.736	0.810	0.659	0.676
best overall training data	0.464	0.672	0.657	0.771	0.836	0.690	0.700
best overall, no deps	0.457	0.675	0.636	0.764	0.834	0.690	0.694
best overall, no deps, no WN	0.426	0.653	0.617	0.719	0.780	0.636	0.654
best overall + experimental	0.466	0.674	0.673	0.772	0.849	0.687	0.706
best individual training data	**0.475**	0.706	0.711	0.788	0.852	0.715	0.727
best individ., no deps	0.465	0.700	0.699	0.781	0.848	**0.722**	0.722
best individ., no deps, no WN	0.448	**0.722**	0.677	0.752	0.791	0.706	0.697
best individ. + experimental	**0.475**	0.711	0.715	**0.795**	**0.864**	0.721	**0.733**

Table 3: Results for task 10.

on semantic textual similarity (Agirre et al., 2012; Agirre et al., 2013). It comprises two subtasks: English semantic textual similarity and Spanish semantic textual similarity. For each subtask, there are sentence pairs from various genres.

We only participate in the English subtask and take the 13th place out of 38 with a weighted mean of Pearson correlation coefficients of 0.694 (best system: 0.761).

5.2 Experiments

From participating in the *SEM 2013 shared task on semantic textual similarity (Greiner et al., 2013) we already know that the composition of the training data is one of the strongest influences on system performance in this task. As the individual data sets are not very similar to each other, we tried to come up with a good subset of the available training data for each data set. In doing so, we were moderately successful as the results in Table 3 show. Running the complete system with all of the available training data on all test data sets results in a lower weighted mean than our submitted run. If we stick to using the same training data for all test data sets and optimize the subset of the training data we use, we achieve a slightly better result than our submitted run (the optimal subset consists of the FNWN, headlines, MSRpar, MSRvid and OnWN data sets). Using that optimal subset of the training data and adding the experimental features to the complete system has a minor positive effect on the weighted mean, with the biggest impact on the headlines and OnWN data sets. Using the complete system without the dependency-based features gives roughly the same results but omitting all resource-heavy features has clearly a negative impact on the results.

In another experiment we try to optimize our strategy of finding the best subset of the training data for each test data set. Doing that gives us a considerably higher weighted mean than using the same training data for every test data set, putting our system on the eighth place. Using the complete system, we find that the best training data subsets for the individual test data sets are those shown in Table 4.

test set	training sets
deft-forum	FNWN, headlines, MSRvid
deft-news	FNWN, MSRpar, MSRvid
headlines	FNWN, headlines, MSRpar
images	FNWN, MSRpar, MSRvid
OnWN	FNWN, MSRvid, OnWN
tweet-news	FNWN, headlines, MSRpar, MSRvid

Table 4: Optimal subsets of training data for use with the complete SemantiKLUE system.

If we add the experimental features to the complete system and still optimize the training data subsets, we get a small boost to the results. Leaving out the dependency-based features does not really hurt performance but also omitting the WordNet-based features has a negative impact on the results.

6 Conclusion

SemantiKLUE is a robust system for predicting the semantic similarity between two texts that can also be used to predict entailment. The system achieves good or very good results in three SemEval-2014 tasks representing a broad variety of semantic similarity problems (cf. Table 5 for an overview of the results of all subtasks). Our two-staged strategy of computing several similarity measures and using them as input for a machine learning mechanism

Subtask	submitted run		complete system		winner score
	score	rank	score	rank	
Task 1, similarity	0.780	7/17	0.798	6/17	0.828
Task 1, entailment	0.823	4/18	0.820	4/18	0.846
Task 3, par-2-sent	0.817	5/34	0.817	5/34	0.837
Task 3, sent-2-phr	0.754	4/34	0.754	4/34	0.777
Task 3, phr-2-word	0.215	11/22	0.284	3/22	0.415
Task 3, word-2-sense	0.314	3/20	0.316	3/20	0.381
Task 3 overall	N/A	4/38	N/A	3/38	N/A
Task 10, deft-forum	0.349	20/38	0.464	12/38	0.531
Task 10, deft-news	0.643	22/37	0.672	19/37	0.785
Task 10, headlines	0.733	15/37	0.657	20/37	0.784
Task 10, images	0.773	16/37	0.771	17/37	0.834
Task 10, OnWN	0.855	3/36	0.836	7/36	0.875
Task 10, tweet-news	0.640	20/37	0.690	12/37	0.792
Task 10 overall	0.694	13/38	0.700	13/38	0.761

Table 5: Overview of results.

proves itself to be adaptable to the needs of the individual tasks.

Using the maximum-weight-matching algorithm for aligning words from both texts that have similar distributional semantics leads to very sound features. Even without the resource-heavy features, the system yields competitive results. In some use cases, those expensive features are almost negligible. Without being dependent on the availability of resources like a dependency parser or a WordNet-like lexical database, SemantiKLUE can easily be adapted to other languages.

Our experimental features from the commercial topic clustering software are useful in some cases; in others at least they do not hurt performance.

We feel that the heuristics based on the dependency structure of the texts do not exhaust all the possibilities that dependency parsing has to offer. In the future we would like to try out more measures based on those structures. Probably some kind of graph edit distance incorporating the similarities between both dependency relations and words might turn out to be a powerful feature.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In *First Joint Conference on Lexical and Computational Semantics*, pages 385–393. ACL.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1: Proceedings of the Main Conference and the Shared Task, pages 32–43. ACL.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Marie-Catherine de Marneffe and Christopher D. Manning, 2008. *Stanford typed dependencies manual*. Stanford University.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Zvi Galil. 1986. Efficient algorithms for finding maximum matching in graphs. *Computing Surveys*, 18(1):23–38.

Paul Greiner and Stefan Evert. in preparation. The Klugator Engine: A distributional approach to open questions in market research.

Paul Greiner, Thomas Proisl, Stefan Evert, and Besim Kabashi. 2013. KLUE-CORE: A regression model of semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1: Proceedings of the Main Conference and the Shared Task, pages 181–186. ACL.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In Gäel Varoquaux,

Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*. ACL.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. 2014. SemEval-2014 task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–283. MIT Press, Cambridge, MA.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google Books Ngram Corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174, Jeju Island, Korea. ACL.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA. Morgan Kaufmann.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*, Reykjavik. ELRA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC '12)*, pages 486–493, Istanbul, Turkey. ELRA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.