DeepPurple: Lexical, String and Affective Feature Fusion for Sentence-Level Semantic Similarity Estimation

Nikolaos Malandrakis[1], **Elias Iosif**[2], **Vassiliki Prokopi**[2], **Alexandros Potamianos**[2], **Shrikanth Narayanan**[1]
[1]Signal Analysis and Interpretation Laboratory (SAIL), USC, Los Angeles, CA 90089, USA
[2]Department of ECE, Technical University of Crete, 73100 Chania, Greece
malandra@usc.edu, iosife@telecom.tuc.gr, vprokopi@isc.tuc.gr, potam@telecom.tuc.gr, shri@sipi.usc.edu

Abstract

This paper describes our submission for the *SEM shared task of Semantic Textual Similarity. We estimate the semantic similarity between two sentences using regression models with features: 1) n-gram hit rates (lexical matches) between sentences, 2) lexical semantic similarity between non-matching words, 3) string similarity metrics, 4) affective content similarity and 5) sentence length. Domain adaptation is applied in the form of independent models and a model selection strategy achieving a mean correlation of 0.47.

1 Introduction

Text semantic similarity estimation has been an active research area, thanks to a variety of potential applications and the wide availability of data afforded by the world wide web. Semantic textual similarity (STS) estimates can be used for information extraction (Szpektor and Dagan, 2008), question answering (Harabagiu and Hickl, 2006) and machine translation (Mirkin et al., 2009). Term-level similarity has been successfully applied to problems like grammar induction (Meng and Siu, 2002) and affective text categorization (Malandrakis et al., 2011). In this work, we built on previous research and our submission to SemEval'2012 (Malandrakis et al., 2012) to create a sentence-level STS model for the shared task of *SEM 2013 (Agirre et al., 2013).

Semantic similarity between words has been well researched, with a variety of knowledge-based (Miller, 1990; Budanitsky and Hirst, 2006) and corpus-based (Baroni and Lenci, 2010; Iosif and Potamianos, 2010) metrics proposed. Moving to sentences increases the complexity exponentially and as a result has led to measurements of similarity at various levels: lexical (Malakasiotis and Androutsopoulos, 2007), syntactic (Malakasiotis, 2009; Zanzotto et al., 2009), and semantic (Rinaldi et al., 2003; Bos and Markert, 2005). Machine translation evaluation metrics can be used to estimate lexical level similarity (Finch et al., 2005; Perez and Alfonseca, 2005), including BLEU (Papineni et al., 2002), a metric using word n-gram hit rates. The pilot task of sentence STS in SemEval 2012 (Agirre et al., 2012) showed a similar trend towards multi-level similarity, with the top performing systems utilizing large amounts of partial similarity metrics and domain adaptation (the use of separate models for each input domain) (Bär et al., 2012; Šarić et al., 2012).

Our approach is originally motivated by BLEU and primarily utilizes "hard" and "soft" n-gram hit rates to estimate similarity. Compared to last year, we utilize different alignment strategies (to decide which n-grams should be compared with which). We also include string similarities (at the token and character level) and similarity of affective content, expressed through the difference in sentence arousal and valence ratings. Finally we added domain adaptation: the creation of separate models per domain and a strategy to select the most appropriate model.

2 Model

Our model is based upon that submitted for the same task in 2012 (Malandrakis et al., 2012). To estimate semantic similarity metrics we use a supervised model with features extracted using corpus-

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 103–108, Atlanta, Georgia, June 13-14, 2013.

based word-level similarity metrics. To combine these metrics into a sentence-level similarity score we use a modification of BLEU (Papineni et al., 2002) that utilizes word-level semantic similarities, string level comparisons and comparisons of affective content, detailed below.

2.1 Word level semantic similarity

Co-occurrence-based. The semantic similarity between two words, w_i and w_j, is estimated as their pointwise mutual information (Church and Hanks, 1990): $I(i,j) = \log \frac{\hat{p}(i,j)}{\hat{p}(i)\hat{p}(j)}$, where $\hat{p}(i)$ and $\hat{p}(j)$ are the occurrence probabilities of w_i and w_j, respectively, while the probability of their co-occurrence is denoted by $\hat{p}(i,j)$. In our previous participation in SemEval12-STS task (Malandrakis et al., 2012) we employed a modification of the pointwise mutual information based on the maximum sense similarity assumption (Resnik, 1995) and the minimization of the respective error in similarity estimation. In particular, exponential weights α were introduced in order to reduce the overestimation of denominator probabilities. The modified metric $I_a(i,j)$, is defined as:

$$I_a(i,j) = \frac{1}{2}\left[\log \frac{\hat{p}(i,j)}{\hat{p}^{\alpha}(i)\hat{p}(j)} + \log \frac{\hat{p}(i,j)}{\hat{p}(i)\hat{p}^{\alpha}(j)}\right]. \quad (1)$$

The weight α was estimated on the corpus of (Iosif and Potamianos, 2012) in order to maximize word sense coverage in the semantic neighborhood of each word. The $I_a(i,j)$ metric using the estimated value of $\alpha = 0.8$ was shown to significantly outperform $I(i,j)$ and to achieve state-of-the-art results on standard semantic similarity datasets (Rubenstein and Goodenough, 1965; Miller and Charles, 1998; Finkelstein et al., 2002).

Context-based: The fundamental assumption behind context-based metrics is that *similarity of context implies similarity of meaning* (Harris, 1954). A contextual window of size $2H+1$ words is centered on the word of interest w_i and lexical features are extracted. For every instance of w_i in the corpus the H words left and right of w_i formulate a feature vector v_i. For a given value of H the context-based semantic similarity between two words, w_i and w_j, is computed as the cosine of their feature vectors: $Q^H(i,j) = \frac{v_i.v_j}{||v_i||\,||v_j||}$. The elements of feature vectors can be weighted according various schemes [(Iosif and Potamianos, 2010)], while, here we use a binary scheme.

Network-based: The aforementioned similarity metrics were used for the definition of a semantic network (Iosif and Potamianos, 2013; Iosif et al., 2013). A number of similarity metrics were proposed under either the *attributional similarity* (Turney, 2006) or the *maximum sense similarity* (Resnik, 1995) assumptions of lexical semantics[1].

2.2 Sentence level similarities

To utilize word-level semantic similarities in the sentence-level task we use a modified version of BLEU (Papineni et al., 2002). The model works in two passes: the first pass identifies exact matches (similar to baseline BLEU), the second pass compares non-matched terms using semantic similarity. Non-matched terms from the hypothesis sentence are compared with all terms of the reference sentence (regardless of whether they were matched during the first pass). In the case of bigram and higher order terms, the process is applied recursively: the bigrams are decomposed into two words and the similarity between them is estimated by applying the same method to the words. All word similarity metrics used are peak-to-peak normalized in the [0,1] range, so they serve as a "degree-of-match". The semantic similarity scores from term pairs are summed (just like n-gram hits) to obtain a BLEU-like hit-rate. Alignment is performed via maximum similarity: we iterate on the hypothesis n-grams, left-to-right, and compare each with the *most similar* n-gram in the reference. The features produced by this process are "soft" hit-rates (for 1-, 2-, 3-, 4-grams)[2]. We also use the "hard" hit rates produced by baseline BLEU as features of the final model.

2.3 String similarities

We use the following string-based similarity features: 1) Longest Common Subsequence Similarity (LCSS) (Lin and Och, 2004) based on the Longest Common Subsequence (LCS) character-based dy-

[1]The network-based metrics were applied only during the training phase of the shared task, due to time limitations. They exhibited almost identical performance as the metric defined by (1), which was used in the test runs.

[2]Note that the features are computed twice on each sentence pair and then averaged.

namic programming algorithm. LCSS represents the length of the longest string (or strings) that is a substring (or are substrings) of two or more strings. 2) Skip bigram co-occurrence measures the overlap of skip-bigrams between two sentences or phrases. A skip-bigram is defined as any pair of words in the sentence order, allowing for arbitrary gaps between words (Lin and Och, 2004). 3) Containment is defined as the percentage of a sentence that is contained in another sentence. It is a number between 0 and 1, where 1 means the hypothesis sentence is fully contained in the reference sentence (Broder, 1997). We express containment as the amount of n-grams of a sentence contained in another. The containment metric is not symmetric and is calculated as: $c(X,Y) = |S(X) \cap S(Y)|/S(X)$, where $S(X)$ and $S(Y)$ are all the n-grams of sentences X and Y respectively.

2.4 Affective similarity

We used the method proposed in (Malandrakis et al., 2011) to estimate affective features. Continuous (valence and arousal) ratings in $[-1, 1]$ of any term are represented as a linear combination of a function of its semantic similarities to a set of seed words and the affective ratings of these words, as follows:

$$\hat{v}(w_j) = a_0 + \sum_{i=1}^{N} a_i \; v(w_i) \; d_{ij}, \qquad (2)$$

where w_j is the term we mean to characterize, $w_1 ... w_N$ are the seed words, $v(w_i)$ is the valence rating for seed word w_i, a_i is the weight corresponding to seed word w_i (that is estimated as described next), d_{ij} is a measure of semantic similarity between w_i and w_j (for the purposes of this work, cosine similarity between context vectors is used). The weights a_i are estimated over the Affective norms for English Words (ANEW) (Bradley and Lang, 1999) corpus.

Using this model we generate affective ratings for every content word (noun, verb, adjective or adverb) of every sentence. We assume that these can adequately describe the affective content of the sentences. To create an "affective similarity metric" we use the difference of means of the word affective ratings between two sentences.

$$\hat{d}_{\text{affect}} = 2 - |\mu(\hat{v}(s_1)) - \mu(\hat{v}(s_2))| \qquad (3)$$

where $\mu(\hat{v}(s_i))$ the mean of content word ratings included in sentence i.

2.5 Fusion

The aforementioned features are combined using one of two possible models. The first model is a Multiple Linear Regression (MLR) model

$$\hat{D}_L = a_0 + \sum_{n=1}^{k} a_n \; f_k, \qquad (4)$$

where $\hat{D}_L$ is the estimated similarity, f_k are the unsupervised semantic similarity metrics and a_n are the trainable parameters of the model.

The second model is motivated by an assumption of cognitive scaling of similarity scores: we expect that the perception of hit rates is non-linearly affected by the length of the sentences. We call this the hierarchical fusion scheme. It is a combination of (overlapping) MLR models, each matching a range of sentence lengths. The first model D_{L1} is trained with sentences with length up to l_1, i.e., $l \leq l_1$, the second model D_{L2} up to length l_2 etc. During testing, sentences with length $l \in [1, l_1]$ are decoded with D_{L1}, sentences with length $l \in (l_1, l_2]$ with model D_{L2} etc. Each of these partial models is a linear fusion model as shown in (4). In this work, we use four models with $l_1 = 10$, $l_2 = 20$, $l_3 = 30$, $l_4 = \infty$.

Domain adaptation is employed, by creating separate models per domain (training data source). Beyond that, we also create a unified model, trained on all data to be used as a fallback if an appropriate model can not be decided upon during evaluation.

3 Experimental Procedure and Results

Initially all sentences are pre-processed by the CoreNLP (Finkel et al., 2005; Toutanova et al., 2003) suite of tools, a process that includes named entity recognition, normalization, part of speech tagging, lemmatization and stemming. We evaluated multiple types of preprocessing per unsupervised metric and chose different ones depending on the metric. Word-level semantic similarities, used for soft comparisons and affective feature extraction, were computed over a corpus of 116 million web snippets collected by posing one query for every word in the Aspell spellchecker (asp,) vocabulary to the Yahoo! search engine. Word-level emotional ratings in continuous valence and arousal scales were produced by a model trained on the ANEW dataset

and using contextual similarities. Finally, string similarities were calculated over the original unmodified sentences.

Next, results are reported in terms of correlation between the generated scores and the ground truth, for each corpus in the shared task, as well as their weighted mean. **Feature selection** is applied to the large candidate feature set using a wrapper-based backward selection approach on the training data.The final feature set contains 15 features: soft hit rates calculated over content word 1- to 4-grams (4 features), soft hit rates calculated over unigrams per part-of-speech, for adjectives, nouns, adverbs, verbs (4 features), BLEU unigram hit rates for all words and content words (2 features), skip and containment similarities, containment normalized by sum of sentence lengths or product of sentence lengths (3 features) and affective similarities for arousal and valence (2 features).

Domain adaptation methods are the only difference between the three submitted runs. For all three runs we train one linear model per training set and a fallback model. For the first run, dubbed linear, the fallback model is linear and model selection during evaluation is performed by file name, therefore results for the OnWN set are produced by a model trained with OnWN data, while the rest are produced by the fallback model. The second run, dubbed length, uses a hierarchical fallback model and model selection is performed by file name. The third run, dubbed adapt, uses the same models as the first run and each test set is assigned to a model (i.e., the fallback model is never used). The test set - model (training) mapping for this run is: OnWN → OnWN, headlines → SMTnews, SMT → Europarl and FNWN → OnWN.

Table 1: Correlation performance for the linear model using lexical (L), string (S) and affect (A) features

Feature	headl.	OnWN	FNWN	SMT	mean
L	0.68	0.51	0.23	0.25	0.46
L+S	0.69	0.49	0.23	0.26	0.46
L+S+A	0.69	0.51	0.27	0.28	0.47

Results are shown in Tables 1 and 2. Results for the linear run using subsets of the final feature set are shown in Table 1. Lexical features (hit rates) are obviously the most valuable features. String similarities provided us with an improvement in the training set which is not reflected in the test set. Affect proved valuable, particularly in the most difficult sets of FNWN and SMT.

Table 2: Correlation performance on the evaluation set.

Run	headl.	OnWN	FNWN	SMT	mean
linear	0.69	0.51	0.27	0.28	0.47
length	0.65	0.51	0.25	0.28	0.46
adapt	0.62	0.51	0.33	0.30	0.46

Results for the three submission runs are shown in Table 2. Our best run was the simplest one, using a purely linear model and effectively no adaptation. Adding a more aggressive adaptation strategy improved results in the FNWN and SMT sets, so there is definitely some potential, however the improvement observed is nowhere near that observed in the training data or the same task of SemEval 2012. We have to question whether this improvement is an artifact of the rating distributions of these two sets (SMT contains virtually only high ratings, FNWN contains virtually only low ratings): such wild mismatches in priors among training and test sets can be mitigated using more elaborate machine learning algorithms (rather than employing better semantic similarity features or algorithms). Overall the system performs well in the two sets containing large similarity rating ranges.

4 Conclusions

We have improved over our previous model of sentence semantic similarity. The inclusion of string-based similarities and more so of affective content measures proved significant, but domain adaptation provided mixed results. While expanding the model to include more layers of similarity estimates is clearly a step in the right direction, further work is required to include even more layers. Using syntactic information and more levels of abstraction (e.g. concepts) are obvious next steps.

5 Acknowledgements

The first four authors have been partially funded by the PortDial project (Language Resources for Portable Multilingual Spoken Dialog Systems) supported by the EU Seventh Framework Programme (FP7), grant number 296170.

References

E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proc. SemEval*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *Proc. *SEM*.

Gnu aspell. `http://www.aspell.net`.

D. Bär, C. Biemann, I. Gurevych, and T. Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proc. SemEval*, pages 435–440.

M. Baroni and A. Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

J. Bos and K. Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, page 628635.

M. Bradley and P. Lang. 1999. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1. The Center for Research in Psychophysiology, University of Florida.

Andrei Z. Broder. 1997. On the resemblance and containment of documents. In *In Compression and Complexity of Sequences (SEQUENCES97*, pages 21–29. IEEE Computer Society.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32:13–47.

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

A. Finch, S. Y. Hwang, and E. Sumita. 2005. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of the 3rd International Workshop on Paraphrasing*, page 1724.

J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

S. Harabagiu and A. Hickl. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912.

Z. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

E. Iosif and A. Potamianos. 2010. Unsupervised semantic similarity computation between terms using web documents. *IEEE Transactions on Knowledge and Data Engineering*, 22(11):1637–1647.

E. Iosif and A. Potamianos. 2012. Semsim: Resources for normalized semantic similarity computation using lexical networks. In *Proc. Eighth International Conference on Language Resources and Evaluation*, pages 3499–3504.

Elias Iosif and Alexandros Potamianos. 2013. Similarity Computation Using Semantic Networks Created From Web-Harvested Data. *Natural Language Engineering*, (submitted).

E. Iosif, A. Potamianos, M. Giannoudaki, and K. Zervanou. 2013. Semantic similarity computation for abstract and concrete nouns using network-based distributional semantic models. In *10th International Conference on Computational Semantics (IWCS)*, pages 328–334.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

P. Malakasiotis and I. Androutsopoulos. 2007. Learning textual entailment using svms and string similarity measures. In *Proceedings of of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47.

P. Malakasiotis. 2009. Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the 47th Annual Meeting of ACL and the 4th Int. Joint Conference on Natural Language Processing of AFNLP*, pages 42–47.

N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan. 2011. Kernel models for affective lexicon creation. In *Proc. Interspeech*, pages 2977–2980.

N. Malandrakis, E. Iosif, and A. Potamianos. 2012. DeepPurple: Estimating sentence semantic similarity using n-gram regression models and web snippets. In *Proc. Sixth International Workshop on Semantic Evaluation (SemEval) – The First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 565–570.

H. Meng and K.-C. Siu. 2002. Semi-automatic acquisition of semantic structures for understanding domain-

specific natural language queries. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):172–181.

G. Miller and W. Charles. 1998. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

S. Mirkin, L. Specia, N. Cancedda, I. Dagan, M. Dymetman, and S. Idan. 2009. Source-language entailment modeling for translating unknown terms. In *Proceedings of the 47th Annual Meeting of ACL and the 4th Int. Joint Conference on Natural Language Processing of AFNLP*, pages 791–799.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

D. Perez and E. Alfonseca. 2005. Application of the bleu algorithm for recognizing textual entailments. In *Proceedings of the PASCAL Challenges Worshop on Recognising Textual Entailment*.

P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxanomy. In *Proc. of International Joint Conference for Artificial Intelligence*, pages 448–453.

F. Rinaldi, J. Dowdall, K. Kaljurand, M. Hess, and D. Molla. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the 2nd International Workshop on Paraphrasing*, pages 25–32.

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

I. Szpektor and I. Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 849–856.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180.

P. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

F. Šarić, G. Glavaš, M. Karan, J. Šnajder, and B. Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proc. SemEval*, pages 441–448.

F. Zanzotto, M. Pennacchiotti, and A. Moschitti. 2009. A machine-learning approach to textual entailment recognition. *Natural Language Engineering*, 15(4):551582.