DCU-SEManiacs at SemEval-2016 Task 1: Synthetic Paragram Embeddings for Semantic Textual Similarity

Chris Hokamp, Piyush Arora
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
{chokamp,parora}@computing.dcu.ie

Abstract

We experiment with learning word representations designed to be combined into sentence-level semantic representations, using an objective function which does not directly make use of the supervised scores provided with the training data, instead opting for a simpler objective which encourages similar phrases to be close together in the embedding space. This simple objective lets us start with high-quality embeddings trained using the Paraphrase Database (PPDB) (Wieting et al., 2015; Ganitkevitch et al., 2013), and then tune these embeddings using the official STS task training data, as well as synthetic paraphrases for each test dataset, obtained by pivoting through machine translation.

Our submissions include runs which only compare the similarity of phrases in the embedding space, directly using the similarity score to produce predictions, as well as a run which uses vector similarity in addition to a suite of features we investigated for our 2015 Semeval submission.

For the crosslingual task, we simply translate the Spanish sentences to English, and use the same system we designed for the monolingual task.

1 Introduction

We describe the work carried out by the DCU-SEManiacs team on the Semantic Textual Similarity (STS) task at SemEval-2016 (Agirre et al., 2013; Agirre et al., 2014; Nakov et al., 2015).

The main ideas we investigate in our systems are:

1. Using a margin-based objective function to train high-quality sentence embeddings without using supervised scores
2. Creating new synthetic training data using machine translation to generate artificial paraphrases
3. Using ensemble models to combine features generated by our embedding networks with features obtained from other sources

1.1 Task Description

The Semeval Semantic Textual Similarity (STS) task provides participants with training data consisting of pairs of sentences annotated with gold-standard semantic similarity scores. The crowd-sourced similarity scores are given on a scale from 0 (no relation) to 5 (semantic equivalence). Thus, our aim is to use the training data to learn a model which predicts a score between 0 and 5 for unseen input pairs (Nakov et al., 2015). The monolingual STS task has been organized each year since 2012, and most approaches have viewed the learning task as a regression problem, where real-valued model output is clipped to be $0 <= \hat{y} <= 5$ (Agirre et al., 2013; Agirre et al., 2014; Nakov et al., 2015).

For two of our three STS systems, we take a novel approach to this task, and directly use the similarity scores produced by the embedding networks as the predicted score. When training the embedding networks, we use the gold scores only to *reduce* the task-internal data to segments with a high-semantic similarity – embeddings are then learned using a simplified training objective which only makes use

Proceedings of SemEval-2016, pages 656–662,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

of training pairs which are "perfect" paraphrases (see section 2.1). Interestingly, these models perform very well without access to the gold standard scores.

The 2016 edition of the STS task also introduced a pilot crosslingual STS task in addition to the monolingual STS task. The crosslingual task is similar to monolingual STS, except either member of each sentence pair may be in Spanish (language identification is *not* provided with the data). In order to use our monolingual STS system in the crosslingual task, we first automatically identify sentences which are probably in Spanish, use machine translation to translate Spanish sentences to English, then approach the crosslingual task as another monolingual task.

Although our systems performed well in both the crosslingual and monolingual STS tasks, we also discuss some possible shortcomings of our approach, and opportunities for improvement.

The rest of the paper is organized as follows: section 2 discusses the main novelties of our submissions, and presents the task-internal and external datasets we leverage for training our systems, section 3 gives a detailed discussion of each of our submitted systems, including information on hyperparameters and training configuration, section 4 gives a summary of experimental results, and section 5 discusses the advantages and disadvantages of our approach, and proposes avenues for future work.

2 Methodology

2.1 Paragram Vectors and PPDB

Wieting (2015) introduced Paragram-phrase embeddings, which use a novel training objective designed to learn robust sentence-level embeddings which are simple bag-of-words averages of the embeddings in each sequence. Wieting discusses several possible means of encoding a sequence into a vector using shallow and deep feedforward and recurrent networks. Surprisingly, the best performing model is a single-layer word embedding matrix, where sentence vectors are constructed by taking the mean of the token embeddings (equation 1).

In our preliminary experiments, we also experimented with deeper feedforward models, as well as mono-directional and bi-directional Long Short-Term Memory (LSTM) recurrent models (Hochreiter and Schmidhuber, 1997) in place of the simple averaging approach; however, we did not observe an improvement in performance, which supports the results presented by Wieting (2015).

We also experimented with objective functions that are more representative of the task objective, such as Kullback-Leibler Divergence (Tai et al., 2015; Wieting et al., 2015); however, we found that the simple margin-based training objective outperforms cost-functions which take the score into account. We hypothesize that this is because the notion of "partial-similarity" is mostly captured by the bag-of-words averaging of all token embeddings to compose the vector representation of a sequence, and because the semantic similarity scores for the STS task are sufficiently coarse that the bulk of their semantic content can be efficiently captured even when all structural information has been discarded.

Equation 2 shows the objective function for the Paragram-phrase model. This function pushes similar examples together, and dissimilar examples apart, driven by the margin δ. $\mathbf{g}(\mathbf{x})$ is some differentiable function which transforms a sequence of tokens into a fixed-size vector. This model is simple to implement, and very fast to train[1]. An additional advantage of the margin-based objective function is that the model can learn from any dataset containing pairs of phrases which are semantically equivalent, enabling the use of unsupervised paraphrase data during training. We exploit this flexibility to greatly improve the performance of our models by tuning the Paragram vectors with new data (section 3.2).

$$embedding(x) = \frac{1}{n}\sum_{i}^{n} W_{word}^{x^i} \tag{1}$$

The word embeddings are the only parameters of this network. Intuitively, tokens whose embeddings have a high L2 norm in this space contribute more to the semantics of a sentence than those whose norm is low. This simple parameterization has the added advantage that it is very fast to train, relative to other possible architectures, such as mono- or bi-directional LSTMs or Gated Recurrent Units (GRUs) (Chung et al., 2015). However, the main

[1] our implementations will be made available at https://www.github.com/chrishokamp/synthetic-embeddings

$$\min_{W_w} \frac{1}{|X|} \sum_{\langle x_1, x_2 \rangle \in X} max(0, \delta - cos(g(x_1), g(x_2)) + cos(g(x_1), g(t_1))) \\ + max(0, \delta - cos(g(x_1), g(x_2)) + cos(g(x_2), g(t_2))) \\ + \lambda_w ||W_{w_{initial}} - W_w||^2 \tag{2}$$

Dataset	DCU 2015 model	Paragram Raw	Paragram + DCU 2015
forums	**.673**	.647	.672
students	.682	**.773**	.732
belief	.708	**.774**	.762
headlines	.810	.748	**.816**
images	.840	.826	**.850**
ALL	.743	.754	**.766**

Table 1: Using the 2015 STS data as development data, a comparison of our 2015 model with raw results from paragram vectors trained on PPDB, and an ensemble system of our 2015 model with the paragram similarity included as a feature.

advantage of this objective function for our work is that models can now be trained with any dataset consisting of pairs of sequences which are semantically equivalent. Thus datasets such as PPDB can be used to train high-quality embeddings.

We start by using the 300-dimensional vectors provided by Wieting (2015). These vectors were trained using the XXL version of PPDB (Ganitkevitch et al., 2013). The sentence embeddings obtained by averaging the raw paragram vectors are used as the development baseline for our systems, and we look for ways to tune the model for the STS task without changing the training objective.

This training objective assumes that each pair is "sufficiently similar" – there is no explicit way to represent partial similarity, since the δ margin dictates that the model should predict a score of at least δ for positive training examples. Therefore, we filter the Semeval STS 2012-2014 training data to contain only those pairs whose similarity is $>= 3.8$[2].

3 System Descriptions

3.1 Generating Negative Examples

Wieting (2015) discusses two ways of selecting negative examples for the paragram vector training objective. The first is to compute the similarity of x_1 and x_2 with every segment in the training minibatch, choosing the most similar segment t_1 to x_1 and the most similar segment t_2 to x_2 that are *not* members of the current pair (x_1, x_2), and to use these as the negative training examples for the pair. The second is to alternate between choosing a random negative example and choosing the most similar phrase. Although the approach of choosing the most similar negative example is theoretically satisfying, there are heavy computational costs: this requires at least N vector comparisons for each example in each pair, where N is the size of the minibatch, and the comparisons must be repeated for each training epoch, since the most similar segment may have changed since the previous epoch. Due to the computational overhead associated with computing the most similar example for each example in each minibatch, we opt instead to use randomly chosen segments as the negative examples. Because the random negative examples are re-selected for each epoch, the model also views more data – each time a training pair is seen, the negative examples t_1 and t_2 selected for x_1 and x_2 are different. Intuitively, this should positively contribute to desirable invariance in the learned semantic embeddings; however, we did not validate this empirically.

3.2 Synthetic Data Generation

We believe that the requirement for human annotation is the major bottleneck for producing more training data for the STS task. Inspired by the

[2] This cutoff parameter was tuned between 3-4.5 with increments of 0.1, note that lowering this threshold results in more training data, but also lowers the quality of the paraphrases, since more partially-similar pairs are included

Dataset	**task-internal**	**synthetic**	**fusion**	*Median*	*Best System*
answer-answer	.627	**.688**	.583	.480	.692
headlines	.719	.687	**.764**	.764	.827
plagiarism	.808	**.819**	.814	.789	.841
postediting	.809	.809	**.847**	.812	.867
question-question	.516	.506	**.566**	.571	.747
ALL	.699	.7133	**.717**	0.689	.778

Table 2: Monolingual STS results by run, with median scores and best scores for reference. "fusion" indicates the ensemble system. Our best performing systems are bolded.

Dataset	**task-internal**	**synthetic**	*Best System*
News	.894	**.897**	.912
Multi-source	.769	**.793**	.819
Mean*	.832	**.846**	.863

Table 3: Crosslingual STS results by run, with best scores for reference. Our best performing systems are bolded.

methodology used to create PPDB (Ganitkevitch et al., 2013), we propose a novel means of producing paraphrases which combines domain-adaptation with paraphrase generation using two or more MT systems. For the experiments presented here, we translate every test sentence into Spanish, then back into English, and add the resulting "pseudo-instance" to the data provided by the task organizers. This method has the additional advantage that we can generate new paraphrases targeted at the sentences in the test data, allowing unsupervised domain adaptation of the model to the test datasets.

This approach is obviously dependent upon the quality of the machine translation output; however, if translation from $e \rightarrow f \rightarrow \hat{e}$ outputs exactly the input e, the new synthetic training example would be of little use. Therefore, the MT systems used for synthetic generation should ideally produce fluent output $\hat{e}$ which paraphrases the original input e, but is diverse with respect to the gold-standard reference translations.

In order to validate that adding synthetic data actually improves performance, we generated synthetic paraphrases for the 2015 Images dataset, and compared performance with respect to the Paragram baseline, and with respect to a model trained with only the task-internal data. These experiments confirmed that synthetic data generation can significantly improve performance. During development, we did not test performance on all of the 2015 data because the process of generating paraphrases is time-consuming, and because we wanted to keep our usage of the Google Translate API within the free credit allocated for test usage of the API, to ensure that our results can be easily replicated.

3.3 Semantic Textual Similarity

We submit three systems to the monolingual STS task. The first system is an ensemble of features from our 2015 submission, together with two features produced by the embedding systems. The second system uses the task-internal data from Semeval 2012-2015 to tune the Paragram embeddings for the STS task. The third system includes one synthetic paraphrase for each sentence in each test dataset, generated by first translating the sentence into Spanish, then back into English. Note that, due to time constraints we did not tune a separate model for each test dataset, instead we used one model trained with all synthetic paraphrases from all test datasets. We believe that training a separate model for each test dataset with synthetic data for only that dataset would improve performance somewhat. Because the scores of the embedding models are in the range 0–1, we scale the outputs by a factor of 5 to match the Semeval scoring system.

3.4 Ensemble Model

In order to test the use of embedding model similarity scores as downstream features, we train an ensemble system with all features from our 2015 submission along with the similarity scores generated

Feature	Gini Coefficient	Description
paraphrase1	0.337	Vector similarity
paraphrase2	0.073	Vector similarity with synthetic data
cosine icf	0.054	Normalized cosine similarity
f15	0.049	Weighted Word Match
f14	0.042	WordNet match
f20	0.038	Relative length difference
product	0.036	Word2Vec based sentence similarity score
f2	0.033	F1-score of number match
nn1	0.028	Comparing only nouns using Word2Vec

Table 4: Ensemble model top 10 features in decreasing order

by both the task-internal and synthetic models. This ensemble, which we call "fusion" was our best system overall (see table 2). We used gradient boosting regressor[3] model trained over combined set of previous year's Semeval STS data sets from 2012-2015. The details of this system are described in (Arora et al., 2015).

3.5 Cross-lingual

For our submission to the cross-lingual STS task, we leverage the Google translate API[4] in three ways: the language identification API is used to detect which segments are in Spanish, the translation API is used to translate Spanish sentences into English, and the pivoting method for generating synthetic paraphrases discussed in section 3.2 is used to generate one new paraphrase for each segment in each test instance. We then apply our monolingual embedding methodology to the translated text with no modification.

3.6 Training Configuration

For the final systems, we use all existing task-internal training data from the Semeval STS task from 2012-2015. The 2015 datasets were used as validation data to find the best system settings, and then included in the training data for the final systems. δ from equation 2 is set to 0.8 for all of our experiments. Embedding dimensionality is 300. We use a minibatch size of 100, and use AdaDelta (Zeiler, 2012) as the gradient update method. The regularization weight λ is set to 10^{-5} for all models.

Dataset	% Unknown
question-question	3.2
plagiarism	4.9
post-editing	4.2
headlines	4.86
answer-answer	0.81

Table 5: Total % unknown for each 2016 dataset

The pretrained paragram vectors have a size of 42091; we do not add any new tokens to the index. Table 5 gives the total percentage of each 2016 test dataset which is unknown with respect to our index. Because the baseline paragram vector index does not contain a special "UNKNOWN" token, we randomly choose a low-frequency token to assign as unknown. Some experimentation showed that using a rare token instead of a stopword results in a small performance improvement.

4 Results

Our monolingual STS systems all performed better than the median system, with the fusion system slightly outperforming the embedding model trained with synthetic data (see tables 2 and 3).

For all systems in both the monolingual and crosslingual STS tasks, we observe an overall improvement over the Paragram baseline when using task-internal training data, and a further improvement when we incorporate synthetic training examples. This result validates the utility of synthetic paraphrases generated by machine translation, and encourages us to explore this avenue further.

[3] `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html`

[4] `https://cloud.google.com/translate/`

For our ensemble based approach, we analyzed the features using Gini importance[5] (Singh et al., 2010). Table 4 shows the importance of the top 10 features in our ensemble model. The relative impact of our two paraphrase features is very high, confirming the utility of the paragram embedding model for the STS task.

Our Features
task-internal paraphrases, synthetic paraphrases, cosine_icf, product_w2v, nn_w2v, vb_w2v, cosine, sum_w2v, det_1, det_2
TakeLab (Šarić et al., 2012) features
weighted_word_match, wn_sim_match, relative_len_difference, number_features, weighted_dist_sim, case_matches, relative_ic_difference

Table 6: Important features.

5 Conclusions

We have presented a method of fine-tuning Paragram-phrase vectors for the STS task using both task-internal and synthetic paraphrases. Our embedding models achieve surprisingly good performance on the STS task without directly taking advantage of the gold-standard scores during training. We have also introduced a novel method of generating synthetic paraphrases for test instances using machine translation. Finally we have shown that a combination of traditional features with the similarity score learned by our embedding models outperforms each individual system.

Future work will focus on increasing the diversity of the synthetic data, and on incorporating multiple objective functions into different stages of the training process.

Acknowledgments

This research was supported by the EXPERT (EU Marie Curie ITN No. 317471) project, and Science Foundation Ireland (SFI) as a part of the ADAPT Centre at Dublin City University (Grant No: 12/CE/I2267),

[5] http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

References

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: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA.

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)*, pages 81–91, Dublin, Ireland.

Piyush Arora, Chris Hokamp, Jennifer Foster, and Gareth Jones. 2015. Dcu: Using distributional semantics and domain adaptation for the semantic textual similarity semeval-2015 task 2. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 143–147, Denver, Colorado, June. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Preslav Nakov, Torsten Zesch, Daniel Cer, and David Jurgens, editors. 2015. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, June.

Sanasam Ranbir Singh, Hema A. Murthy, and Timothy A. Gonsalves. 2010. Feature selection for text classification based on gini coefficient of inequality. In *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, FSDM, held at PAKDD 2010, Hyderabad, India, June 21st, 2010*, pages 76–85.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th*

International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 1556–1566.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 441–448, Montréal, Canada.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701.