Sentence Embedding Evaluation Using Pyramid Annotation

Tal Baumel
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
talbau@cs.bgu.ac.il

Raphael Cohen
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
cohenrap@cs.bgu.ac.il

Michael Elhadad
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
elhadad@cs.bgu.ac.il

Abstract

Word embedding vectors are used as input for a variety of tasks. Choosing the right model and features for producing such vectors is not a trivial task and different embedding methods can greatly affect results. In this paper we repurpose the "Pyramid Method" annotations used for evaluating automatic summarization to create a benchmark for comparing embedding models when identifying paraphrases of text snippets containing a single clause. We present a method of converting pyramid annotation files into two distinct sentence embedding tests. We show that our method can produce a good amount of testing data, analyze the quality of the testing data, perform test on several leading embedding methods, and finally explain the downstream usages of our task and its significance.

1 Introduction

Word vector embeddings [Mikolov *et al*. 2013] have become a standard building block for NLP applications. By representing words using continuous multi-dimensional vectors, applications take advantage of the natural associations among words to improve task performance. For example, POS tagging [Al Rfou *et al.* 2014], NER [Passos *et al.* 2014], parsing [Bansal *et al*. 2014], Semantic Role Labeling [Herman *et al*. 2014] or sentiment analysis [Socher *et al*. 2011] - have all been shown to benefit from word embeddings, either as additional features in existing supervised machine learning architectures, or as exclusive word representation features. In deep learning applications, word embeddings are typically used as pre-trained initial layers in deep architectures, and have been shown to improve performance on a wide range of tasks as well (see for example, [Cho *et al*., 2014; Karpathy and Fei-Fei 2015; Erhan *et al*,. 2010]).

One of the key benefits of word embeddings is that they can bring to tasks with small annotated datasets and small observed vocabulary, the capacity to generalize to large vocabularies and to smoothly handle unseen words, trained on massive scale datasets in an unsupervised manner.

Training word embedding models is still an art with various embedding algorithms possible and many parameters that can greatly affect the results of each algorithm. It remains difficult to predict which word embeddings are most appropriate to a given task, whether fine tuning of the embeddings is required, and which parameters perform best for a given application.

We introduce a novel dataset for comparing embedding algorithms and their settings on the specific task of comparing short clauses. The current state-of-the-art paraphrase dataset [Dolan and Brockett, 2005] is quite small with 4,076 sentence pairs (2,753 positive). The Stanford Natural Language Inference (SNLI) (Bowman *et al*., 2015) corpus contains 570k sentences pairs labeled with one of the tags: entailment, contradiction, and neutral. SNLI improves on previous paraphrase datasets by eliminating indeterminacy

Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pages 145–149,
Berlin, Germany, August 12, 2016.

of event and entity coreference which make human entailment judgment difficult. Such indeterminacies are avoided by eliciting descriptions of the same images by different annotators.

We repurpose manually created data sets from automatic summarization to create a new paraphrase dataset with 197,619 pairs (8,390 positive and challenging distractors in the negative pairs). Like SNLI, our dataset avoids semantic indeterminacy because the texts are generated from the same news reports – we thus obtain definite entailment judgments but in the richer domain of news report as opposed to image descriptions. The propositions in our dataset are on average 12.1 words long (as opposed to about 8 words for the SNLI hypotheses).

In addition to paraphrase, our dataset captures a notion of centrality - the clause elements captured are Summary Content Units (SCU) which are typically shorter than full sentences and intended to capture proposition-level facts. As such, the new dataset is relevant for exercising the large family of "Sequence to Sequence" (seq2seq) tasks involving the generation of short text clauses [Sutskever *et al*. 2014].

The paper is structured as follows: §2 describes the pyramid method; §3 describes the process for generating a paraphrase dataset from a pyramid dataset; in §4, we evaluate a number of algorithms on the new benchmark and in §5, we explain the importance of the task.

2 The Pyramid Method

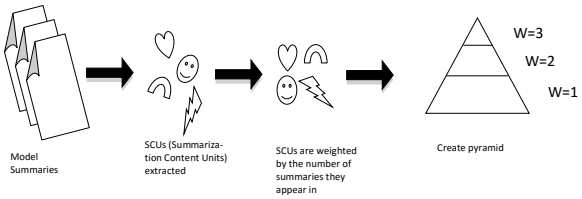

Figure 1: Pyramid Method Illustration

The Pyramid Method (Nenkova and Passonneau, 2004) is a summarization evaluation scheme designed to achieve consistent score while taking into account human variation in content selection and formulation. This evaluation method is manual and can be applied to both manual and automatic summarization. It has been included as a main evaluation technique in all DUC datasets since 2005 (Passonneau *et al*., 2006).

In order to use the method, a pyramid file must first be created manually (Fig. 1):

- Create a set of model (gold) summaries
- Divide each summary into Summary Content Units (SCUs) – SCUs are key facts extracted from the manual summarizations, they are no longer than a single clause
- A pyramid file is created where each SCU is given a score by the number of summaries in which it is mentioned (*i.e.*, SCUs mentioned in 3 summaries will obtain a score of 3)

After the pyramid is created, it can be used to evaluate a new summary:

- Find all the SCUs in the summary
- Sum the score of all the found SCUs and divide it by the maximum score that the same amount of SCUs can achieve

SCUs are extracted from different source summaries, written by different authors. When counting the number of occurrences of an SCU, annotators effectively create clusters of text snippets that are judged semantically equivalent in the context of the source summaries. SCUs actually refer to clusters of text fragments from the summaries and a label written by the pyramid annotator describing the meaning of the SCU.

In our evaluation, we divert the pyramid file from its original intention of summarization evaluation, and propose to use it as a proposition paraphrase dataset.

3 Repurposing Pyramid Annotations

We define two types of tests that can be produced from a pyramid file: a binary decision test and a ranking test. For the binary decision test, we collect pairs of different SCUs from manual summaries and the label given to the SCU by annotators. The binary decision consists of deciding whether the pair is taken from the same SCU. In order to make the test challenging and

still achievable, we add the following constraints on pair selection:

- Both items must contain at least 3 words;
- For non-paraphrase pairs, both items must match on more than 3 words;
- Both items must not include any pronouns;
- The pair must be lexically varied (at least one content word must be different across the items)

Non-paraphrase pair: '*Countries worldwide sent Equipment', 'Countries worldwide sent Relief Workers*'	**Paraphrase pair**: '*countries worldwide sent money equipment', 'rescue equipment poured in from around the world'*

Figure 2: Binary test pairs example

For the ranking test, we generate a set of multiple choice questions by taking as a question an SCU appearance in the text and the correct answer is another appearance of the same SCU in the test. To create synthetic distractors, we use the 3 most lexically similar text segments from distinct SCUs:

Morris Dees co-founded the SPLC:

1. **Morris Dees was co-founder of the Southern Poverty Law Center (SPLC) in 1971 and has served as its Chief Trial Counsel and Executive Director**
2. Dees and the SPLC seek to destroy hate groups through multi-million dollar civil suits that go after assets of groups and their leaders
3. Dees and the SPLC have fought to break the organizations by legal action resulting in severe financial penalties
4. The SPLC participates in tracking down hate groups and publicizing their activities in its Intelligence Report

Figure 3: Ranking test example question

Using DUC-2007, 2006 and 2005 pyramid files (all contain news stories), we created 8,755 questions for the ranking test and for the binary test we generated 8,390 positive pairs, 189,229 negative pairs for a total 197,619 pairs. The propositions in the dataset contain 95,286 words (6,882 unique).

4 Baseline Embeddings Evaluation

In order to verify that this task indeed is sensitive to differences in word embeddings, we evaluated 8 different word embeddings on the task as a baseline: Random, None (One-Hot embedding), word2vec (Mikolov *et al*., 2013) trained on Google News and two models trained on Wikipedia with different window sizes (Levy and Goldberg 2014), word2vec trained with Wikipedia dependencies (Levy and Goldberg 2014), GloVe (Pennington *et al*., 2014) and Open IE based embeddings (Stanovsky *et al*., 2015). For all of the embeddings, we measured sentence similarity as the cosine similarity[1] of the normalized sum of all the words in the sentences.

For the binary decision test, we evaluated the embedding by finding a threshold for answering where a pair is a paraphrase that maximizes the F-measure (trained over 10% the dataset and tested on the rest) of the embedding decision. For the rank test, we computed the percentage of questions where the correct answer achieved the highest similarity score and the MRR measure (Craswell, 2009).

Results are summarized in Table 1.

	Binary Test (F-measure)	Ranking Test (Success Rate)	Ranking Test (Mean reciprocal rank)
Random-Baseline	0.04059	24.662%	0.52223
One-Hot	0.26324	63.973%	0.77202
word2vec-BOW (google-news)	0.42337	**66.960%**	**0.78933**
word2vec-BOW2 (Wikipedia)	0.39450	61.684%	0.75274
word2vec-BOW5 (Wikipedia)	0.40387	62.886%	0.76292
word2vec-Dep	0.39097	60.025%	0.74003
GloVe	0.37870	63.000%	0.76389
Open IE Embedding	**0.42516**	65.667%	0.77847

Table 1: Different embedding performance on binary and ranking tests.

The OpenIE Embedding model scored the highest for the binary test (0.42 F). Word2vec model trained on google news achieved the best success rate in the ranking test (precision@1 of 66.9%),

[1] Using spaCy for tokenization

significantly better than the word2vec model trained on Wikipedia (62.8%). MRR for ranking was dominated by word2vec with 0.41.

5 Task Significance

The task of identifying paraphrases specifically extracted from pyramids can aid NLP sub-fields such as:

- **Automatic Summarization**: Identifying paraphrases can both help identifying salient information in multi-document summarization and evaluation by recreating pyramid files and applying them on automatic summaries;
- **Textual Entailment**: Paraphrases are bi-directional entailments;
- **Sentence Simplification**: SCUs capture the central elements of meaning in observable long sentences.
- **Expansion of Annotated Datasets**: Given an annotated dataset (*e.g.*, aligned translations), unannotated sentences could be annotated the same as their paraphrases

6 Conclusion

We presented a method of using pyramid files to generate paraphrase detection tasks. The suggested task has proven challenging for the tested methods, as indicated by the relatively low F-measures reported in Table 1 on most models. Our method can be applied on any pyramid annotated dataset so the reported numbers could increase by using other datasets such as TAC 2008, 2009, 2010, 2011 and 2014[2]. We believe that the improvement that this task can provide to downstream applications is a good incentive for further research.

[2] http://www.nist.gov/tac/tracks/index.html

Acknowledgments

This work was supported by the Lynn and William Frankel Center for Computer Sciences, Ben-Gurion University. We thank the reviewers for extremely helpful advice. We would also like to thanks the reviewers for their insight.

References

Al-Rfou, R., Perozzi, B. and Skiena, S., 2013. Polyglot: Distributed word representations for multilingual nlp. arXiv preprint arXiv:1307.1662.

Bansal, M., Gimpel, K. and Livescu, K., 2014. Tailoring Continuous Word Representations for Dependency Parsing. In ACL (2) (pp. 809-815).

Bowman, S.R., Angeli, G., Potts, C. and Manning, C.D., 2015. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326.

Craswell, N., 2009. Mean reciprocal rank. In Encyclopedia of Database Systems (pp. 1703-1703). Springer US

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Dolan, W.B. and Brockett, C., 2005, October. Automatically constructing a corpus of sentential paraphrases. In Proc. of IWP.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P. and Bengio, S., 2010. Why does unsupervised pre-training help deep learning?. The Journal of Machine Learning Research, 11, pp.625-660.

Goldberg, Y. and Levy, O., 2014. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

Hermann, K.M., Das, D., Weston, J. and Ganchev, K., 2014, June. Semantic Frame Identification with Distributed Word Representations. In ACL (1) (pp. 1448-1458).

Levy, O. and Goldberg, Y., 2014. Dependency-Based Word Embeddings. In ACL (2) (pp. 302-308).

Mikolov, T., Yih, W.T. and Zweig, G., 2013, June. Linguistic Regularities in Continuous Space Word Representations. In HLT-NAACL (pp. 746-751). Vancouver

Nenkova, A. and Passonneau, R., 2004. Evaluating content selection in summarization: The pyramid method.

Passonneau, R., McKeown, K., Sigelman, S. and Goodkind, A., 2006. Applying the pyramid

method in the 2006 Document Understanding Conference.

Passos, A., Kumar, V. and McCallum, A., 2014. Lexicon infused phrase embeddings for named entity resolution. arXiv preprint arXiv:1404.5367.

Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global Vectors for Word Representation. In EMNLP (Vol. 14, pp. 1532-1543).

Karpathy, A. and Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3128-3137).

Socher, R., Pennington, J., Huang, E.H., Ng, A.Y. and Manning, C.D., 2011, July. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 151-161). Association for Computational Linguistics.

Stanovsky, G., Dagan, I, and Mausam, 2015. Open IE as an Intermediate Structure for Semantic Tasks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)