Quality of Word Embeddings on Sentiment Analysis Tasks

Quality of Word Embeddings on Sentiment Analysis Tasks

Erion Çano and Maurizio Morisio

Politecnico di Torino,
Duca degli Abruzzi, 24, 10129 Torino, Italy
erion.cano@polito.it
maurizio.morisio@polito.it

Abstract. Word embeddings or distributed representations of words are being used in various applications like machine translation, sentiment analysis, topic identification etc. Quality of word embeddings and performance of their applications depends on several factors like training method, corpus size and relevance etc. In this study we compare performance of a dozen of pretrained word embedding models on lyrics sentiment analysis and movie review polarity tasks. According to our results, Twitter Tweets is the best on lyrics sentiment analysis, whereas Google News and Common Crawl are the top performers on movie polarity analysis. Glove trained models slightly outrun those trained with Skip-gram. Also, factors like topic relevance and size of corpus significantly impact the quality of the models. When medium or large-sized text sets are available, obtaining word embeddings from same training dataset is usually the best choice.

Keywords: Word Embeddings, Lyrics Mood Analysis, Movie Review Polarity

1 Introduction

Semantic vector space models of language were developed in the 90s to predict joint probabilities of words that appear together in a sequence. A particular upturn was proposed by Bengio et al. in [1], replacing sparse n-gram models with word embeddings which are more compact representations obtained using feed-forward or more advanced neural networks. Recently, high quality and easy to train Skip-gram shallow architectures were presented in [10]and considerably improved in [11] with the introduction of negative sampling and subsampling of frequent words. The "magical" ability of word embeddings to capture syntactic and semantic regularities on text words is applicable in various applications like machine translations, error correcting systems, sentiment analyzers etc. This ability has been tested in [12] and other studies with analogy question tests of the form "A is to B as C is to __" or male/female relations. A recent improved method for generating word embeddings is Glove [15] which makes efficient use of global statistics of text words and preserves the linear substructure of Skip-gram word2vec, the other popular method. Authors report that Glove outperforms other methods such as Skip-gram in several tasks like word similarity, word analogy etc. In this paper we examine the quality of word embeddings on 2 sentiment analysis tasks: Lyrics mood recognition and movie review polarity analysis. We compare various models pre-trained with Glove and Skip-gram, together with corpora we train ourself. Our goal is to report the best performing models as well as to observe the impact that certain factors like training method, corpus size and thematic relevance of texts might have on model quality. According to the results, Common Crawl, Twitter Tweets and Google News are the best performing models. Corpus size and thematic relevance have a significant role on the performance of the generated word vectors. We noticed that models trained with Glove slightly outperform those trained with Skip-gram in most of experiments.

2 Word Embedding Corpora and Models

In this section we present the different word embedding models that we compare. Most of them are pretrained and publicly available. Two of them (Text8Corpus and Moody-Corpus) were trained by us. The full list with some basic characteristics is presented in Table 1. Wikipedia Gigaword is a combination of Wikipedia 2014 dump and Gigaword

Table 1: List of word embedding corpora

Corpus Name	Training	Dim	Size	Voc	URL
Wiki Gigaword 300	Glove	300	6B	400000	link
Wiki Gigaword 200	Glove	200	6B	400000	link
Wiki Gigaword 100	Glove	100	6B	400000	link
Wiki Gigaword 50	Glove	50	6B	400000	link
Wiki Dependency	word2vec	300	1B	174000	link
Google News	word2vec	300	100B	3M	link
Common Crawl 840	Glove	300	840B	2.2M	link
Common Crawl 42	Glove	300	42B	1.9M	link
Twitter Tweets 200	Glove	200	27B	1.2M	link
Twitter Tweets 100	Glove	100	27B	1.2M	link
Twitter Tweets 50	Glove	50	27B	1.2M	link
Twitter Tweets 25	Glove	25	27B	1.2M	link
Text8Corpus	word2vec	200	17M	25000	link
MoodyCorpus	word2vec	200	90M	43000	link

5 with about 6 billion tokens in total. It was created by authors of [15] to evaluate Glove performance. Wikipedia Dependency corpus is a collection of 1 billion tokens from Wikipedia. The method used for training it is a modified version of Skip-gram word2vec described in [7]. Google News is one of the biggest and richest text sets with 100 billion tokens and a vocabulary of 3 million words and phrases [10]. It was trained using Skip-gram word2vec with negative sampling, windows size 5 and 300 dimensions. Even bigger is Common Crawl 840, a huge corpus of 840 billion tokens and 2.2 million word vectors also used at [15]. It contains data of Common Crawl (http://commoncrawl.org), a nonprofit organization that creates and maintains public datasets by crawling the web. Common Crawl 42 is a reduced version made up of 42 billion tokens and a vocabulary of 1.9 million words. Common Crawl 840 and Common Crawl 42 were trained with Glove method producing vectors of 300 dimensions for each word. The last Glove corpus is the collection of Twitter Tweets. It consists of 2 billion tweets, 27 billion tokens and 1.2 million words. To observe the role of corpus size in quality of generated

embeddings, we train and use Text8Corpus, a smaller corpus consisting of 17 million tokens and 25,000 words. The last model we use is MoodyCorpus, a collection of lyrics that followed our work in [3] where we build and evaluate MoodyLyrics, a sentiment annotated dataset of songs. The biggest part of MoodyCorpus was built using lyrics of Million Song Dataset (MSD) songs (https://labrosa.ee.columbia.edu/millionsong/). As music tastes and characteristics change over time (http://kaylinwalker.com/50-years-of-pop-music), it is better to have diversified sources of songs in terms of epoch, genre etc. Thereby we added songs of different genres and epochs that we found in two subsets of MSD, Cal500 and TheBeatles. The resulting corpus of 90 million tokens and 43,000 words can be downloaded from http://softeng.polito.it/erion. Further information about public music datasets can be found at [2].

3 Sentiment Analysis Tasks

The problem of music mood recognition is about utilizing machine learning, data mining and other techniques to automatically classify songs in 2 or more emotion categories with highest possible accuracy. Different combinations of features such as audio or lyrics are involved in the process. In this study we make use of song lyrics exploiting the dataset described in [9] (here AM628). The original dataset contains 771 song texts collected from AllMusic portal. AllMusic tags and 3 human experts were used for the annotation of songs. We balanced the dataset obtaining 314 positive and 314 negative lyrics. We also utilize MoodyLyrics (here ML3K), a dataset of 3,000 mood labeled songs from different genres and epochs described in [3]. Pioneering work in movie review polarity analysis has been conducted by Pang and Lee in [14] and [13]. The authors released sentiment polarity dataset, a collection of 2,000 movie reviews categorized as positive or negative. Deep learning techniques and distributed word representations appeared on recent studies like [17] where the role of RNNs (Recurrent Neural Networks), and CNNs (Convolutional Neural Networks) is explored. The author reports that CNNs perform best. An important work that has relevance here is [8] where authors present an even larger movie review dataset of 50,000 movie reviews from IMBD. This dataset has been used in various works such as [5], [16] etc. For our experiments we used a chunk of 10K (MR10K) as well as the full set (MR50K). We first cleaned and tokenized texts of the datasets. The dataset of the current run is loaded and a set of unique text words is created. All 14 models are also loaded in the script. We train a 15th (self_w2v) model using the corpus of the current run and Skip-gram method. The script iterates in every line of the pretrained models splitting apart the words and the float vectors and building {word: vec} dictionaries later used as classification feature sets. Next we prepare the classification models using tf-idf vectorizer which has been successfully applied in similar studies like [4]. Instead of applying tf-idf in words only as in other text classifiers, we vectorize both word (for semantic relevance) and corresponding vector (for syntactic and contextual relevance). Random forest was used as classifier and 5-fold cross-validation accuracy is computed for each of the models.

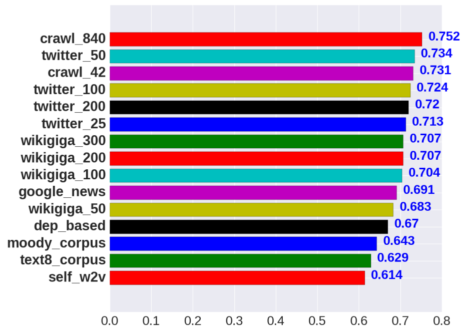

Fig. 1: Lyric accuracies on AM628

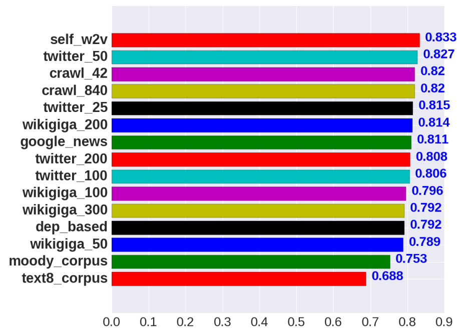

Fig. 2: Lyric accuracies on ML3K

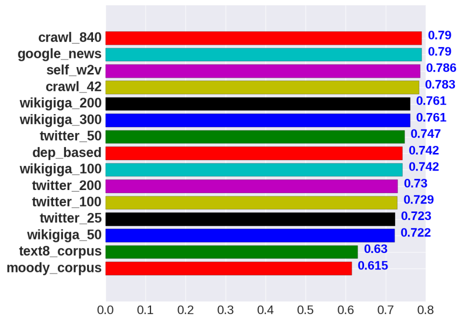

Fig. 3: Review accuracies on MR10K

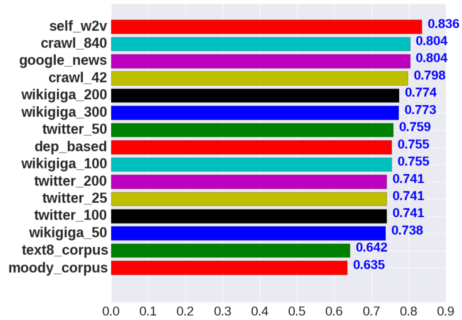

Fig. 4: Review accuracies on MR50K

4 Results

In figures 1 and 2 we see results of 5-fold cross-validation on the 2 lyrics datasets. Top three models are crawl_840, twitter_50 and self_w2v. On AM628 (very smallest dataset), it is crawl_840 (the biggest model) that leads, followed by twitter_50. Self_w2v is severely penalized by its size and thus is at the bottom. On ML3K (large dataset) self_w2v reaches the top of the list, leaving behind twitter_50. Wikigiga, google_news and dep_based are positioned in the middle whereas MoodyCorpus and Text8Corpus end the list. Their accuracy scores drift from 0.62 to 0.75. It is interesting to see how self_w2v goes up from the last to the top, with scores edging between 0.61 and 0.83. This model is trained with the data of each experiment and depends on the size of that dataset which grows significantly (see Table 2). We see that accuracy values we got here are in line with reports from other similar works such as [6] where they use a dataset of 1032 lyrics from AllMusic to perform content analysis with text features. Accuracy scores for movie review polarity prediction are presented in figures 3 and 4. Again we see that crawl_840 performs very well. Google_news is also among the top whereas Twitter models are positioned in the middle of the list. Once again self_w2v grows considerably, this time from the 3rd place to the top. On MR50K it has a discrete margin of more than 0.03 from the 2nd position. Again wikigiga models are positioned in the middle of the list and the worst performing models are MoodyCorpus and Text8Corpus. Our scores on this task are somehow lower than those reported from various studies that explore advanced deep learning constructs on same dataset. In [8] for example, authors who created movie review dataset try on it their probabilistic model that is able to capture semantic similarities between words. They report a maximal accuracy of 0.88. A study that uses a very similar method is [16] where authors combine random forest with word vector average values. On movie review dataset they achieve an accuracy of 0.84 which is about what we got here.

Table 2: Properties of self_w2v

Trial	Dataset	Dim	Size	Voc	Score
1	AM628	200	156699	8756	0.614
2	ML3K	200	1028891	17890	0.833
3	MR10K	300	2343641	53437	0.786
4	MR50K	300	11772959	104203	0.836

5 Discussion

In this paper we examined the quality of different word embedding models on two sentiment analysis tasks: Lyrics mood recognition and movie review polarity. We observed the role of factors like training method, vocabulary and corpus size and thematic relevance of texts. According to our results, the best performing models are Common Crawl, Twitter Tweets and Google News. In general, models trained with Glove slightly outperform those trained using Skip-gram, especially on lyrics sentiment analysis (Twitter and Crawl). We also notice that vocabulary richness and corpus size have a significant influence on model quality. The biggest models like crawl_840 are always among the best. Likewise self_w2v performs very well on both tasks when trained with medium or large data sizes (see Table 2). Being the smallest in sizes, MoodyCorpus and Text8Corpus are always the worst. Regarding thematic relevance, Twitter corpora perform better on lyrics sentiment analysis. They are large and rich in vocabulary with texts of an informal and sentimental language. This language is very similar to the one of song lyrics, with ***love*** being the predominant word (see word cloud in [3]). Movie review results on the other hand, are headed by Common Crawl and Google News which are the largest, both in size and vocabulary. These models are trained with diverse and informative texts that cover every possible subject or topic. Having a look on some movie reviews we also see a similar language with comments about the movies of different categories. Furthermore, we saw that when training set is big enough, obtaining word embeddings from it (self_w2v) is the best option.

References

1. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, Mar. 2003.

2. E. Çano and M. Morisio. Characterization of public datasets for recommender systems. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on*, pages 249–257, Sept 2015.
3. E. Çano and M. Morisio. Moodylyrics: A sentiment annotated lyrics dataset. In *2017 International Conference on Intelligent Systems, Metaheuristics and Swarm Intelligence*, Hong Kong, March 2017.
4. X. Hu, J. S. Downie, and A. F. Ehmann. Lyric text mining in music mood classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, pages 411–416, 2009.
5. R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 103–112, 2015.
6. W.-S. Lee and D. Yang. Music emotion identification from lyrics. *2013 IEEE International Symposium on Multimedia*, 00:624–629, 2009.
7. O. Levy and Y. Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308, 2014.
8. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
9. R. Malheiro, R. Panda, P. Gomes, and R. P. Paiva. Emotionally-relevant features for classification and regression of music lyrics. *IEEE Transactions on Affective Computing*, PP(99):1–1, 2016.
10. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
11. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.
12. T. Mikolov, S. W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013.
13. B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278, 2004.
14. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.
15. J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. ACL, 2014.
16. H. Pouransari and S. Ghili. Deep learning for sentiment analysis of movie reviews. Technical report, Stanford University, 2014.
17. H. Shirani-Mehr. Applications of deep learning to sentiment analysis of movie reviews. Technical report, Stanford University, 2014.