Role of Data Properties on Sentiment Analysis of Texts via Convolutions

Role of Data Properties on Sentiment Analysis of Texts via Convolutions

Erion Çano and Maurizio Morisio

Politecnico di Torino,
Duca degli Abruzzi, 24, 10129 Torino, Italy

Abstract. Dense and low dimensional word embeddings opened up the possibility to analyze text polarity with highly successful deep learning techniques like Convolution Neural Networks. In this paper we utilize pretrained word vectors in combination with simple neural networks of stacked convolution and max-pooling layers, to explore the role of dataset size and document length in sentiment polarity prediction. We experiment with song lyrics and reviews of products or movies and see that convolution-pooling combination is very fast and yet quiet effective. We also find interesting relations between dataset size, text length and length of feature maps with classification accuracy. Our next goal is the design of a generic neural architecture for analyzing polarity of various text types, with high accuracy and few hyper-parameter changes.

Keywords: Textual Sentiment Analysis, Convolution Neural Networks, Text Dataset Properties

1 Introduction

Deep learning techniques today excel in lots of complicated tasks like object detection, machine translation, speech recognition or sentiment analysis, becoming a hype in computing industry. Self-driving cars, intelligent personal assistants and chat bots passing Turing test, are some examples of today's deep learning revolution. In particular, Convolution Neural Networks (CNNs) that try to mimic the structure of visual cortex, have become highly effective in image analysis applications. The basic structure of a CNN was proposed by LeCun *et al.* in [12] for recognizing images of handwritten digits. After a decade of lethargy, CNNs exploded in late 2000s and are now part of many advanced image recognition architectures like Inception [15], ResNet [5] or others. Applicability in text analysis was initially hindered by problems like small datasets, data sparsity and very high dimensionality. Explosion of user generated texts in social media increased text set sizes available for research in areas like sentiment analysis, topic recognition, machine translation etc. Furthermore, the introduction of word embeddings by Bengio *et al.* in [1], alleviated both data sparsity and high dimensionality problems. One of the first studies using CNNs for sentiment analysis was conducted by Kim in [9]. He used basic CNNs on short sentences, reporting excellent results with little computation load. Other studies like [18] used deeper CNN architectures that start from characters and build up word patterns that are used as classification features. Instead, in [17] or [10] authors use gated networks or combinations of CNNs with RNNs (Recurrent Neural

Networks) for sentiment analysis of short texts or sentences. All above studies have largely increased awareness about effectiveness of CNNs in various text analysis applications like sentiment analysis or topic modeling. However, they are usually limited in using datasets of short text documents or sentences of a certain length. In this paper instead, we experimented with text datasets of different size and document lengths, from short sentences to documents of more than 500 words. We used texts of product reviews, movie reviews and song lyrics trying to observe possible tendencies relating properties like size of datasets and length of text documents with sentiment polarity classification accuracy. To this end, we made use of simple neural networks based on convolution and pooling layers that are very fast to train. Obtained accuracy scores were slightly lower than the state of the art using same datasets. This is something we expected, given the elementary models we experimented with. We also observed that bigger datasets are usually better interpreted using several levels of concatenated convolution and pooling layers stacked upon each other. Regarding text lengths, we saw that the longer texts require larger pooling. Optimal feature maps are usually from 6 to 18. We cannot tell if these results are specific to the datasets and/or neural networks we used, or general tendencies useful in a broader perspective. We will thus experiment with a higher variety of document types, lengths and contents for building a neural network architecture for sentiment polarity analysis of different text types with few changes in hyper-parameters.

2 Data Preparation

We experiment with datasets of various sizes and different text document lengths or content. Prior to using the documents, we applied basic text cleaning and preprocessing in each of them. The step by step process is explained in the following subsections.

2.1 Experimental Datasets

Mlpn MoodyLyrics is a dataset of 2,596 songs labeled as 'happy', 'angry', 'sad' or 'relaxed' [3]. It was created to experiment on music emotion recognition based on lyrics. Here we use Mlpn, the version with 2 categories that contains 2,500 positive and 2,500 negative songs, labeled using Last.fm tags as described in [4].

Phon The dataset of unlocked smartphone reviews contains user descriptions about phones sold in Amazon. Users have provided a text description and a 1-5 star rating for each phone. Entries without review or star rating were cleared out. Furthermore, 3-star reviews were removed as they contain both positive and negative (ambiguous) descriptions. We reached to a total of 232,546 reviews. Finally, 1-star and 2-star reviews were labeled as negative whereas 4-star and 5-star reviews were labeled as positive.

Imdb IMDB movie review dataset [13] is a ground-truth text bundle of 50K movie reviews, frequently used in text analysis studies. Text documents are of different lengths in the original version and 400 words long in the popular preprocessed one. The goal is to determine if each movie review is positive or negative.

Sent Sentence polarity dataset is one of the first sentiment analysis datasets created by Pang and Lee in 2005. It consists of 5331 positive and 5331 negative texts extracted from IMDB archive of movie reviews and categorized as positive or negative.

Yelp Yelp Review Polarity is one of the various big datasets created by Zhang *et al.* in [19]. It consists of 598K yelp reviews about businesses (restaurants, hotels, etc.), balanced and labeled as positive or negative.

Table 1: Summary of dataset statistics

Dataset	#Docs	MinL	AvgL	MaxL	UsedL
Song Lyrics	5K	23	227	2733	450
Sentence Polarity	10K	1	17	46	30
Movie Reviews	50K	5	204	2174	400
Phone Reviews	232K	3	47	4607	100
Yelp Reviews	598K	1	122	963	270

2.2 Document Preprocessing and Statistics

Basic text preprocessing on the text documents of each dataset was performed, removing remaining html tags. We kept smileys like :(, :-(, :P, :D, :-), :) that are usually helpful, as they correlate well with the emotion categories. Stopwords on the other hand, appear very frequently but carry little or no meaning at all. We removed {'the', 'these', 'those', 'this', 'of', 'at', 'that', 'a', 'for', 'an', 'as', 'by'} only. Residues of short forms (e.g., 'll', 'd', 's', 't', 'm') and negation forms (e.g., couldn', don, 'hadn, 'didn') were kept in, as their presence or absence can change the emotional polarity of the sentence and thus hurt prediction performance. Finally, we cleared out any remaining odd patterns and lowercased everything. After preprocessing, we observed length distribution of documents and their statistics summarized in Table 1. Song lyrics lengths range between 23 and 2733 words, with an average of 227. Smartphone and movie reviews are highly dispersed, with lengths ranging from 3 to 4607 and average 47 in the former and range 5 to 2174 averaging to 204 in the later. Yelp reviews were less dispersed, with length spanning between 1 and 963 words. Sentence polarity datasets is the most uniform set of texts, with sentence lengths from a single word to 46. We also observed that in any dataset, most documents are quiet short. In review datasets for example, very few documents are longer than 500 words. As a result, we decided to clip the few long documents and pad the shorter ones to uniform lengths we picked considering the distributions. When deciding about the length of each dataset, we took care to clip less than 10 % of texts. Experimentation lengths are shown in the last column of Table 1. This process greatly reduced the computation requirements of each experiment with no loss in quality of data.

3 Experimental Setup

3.1 Word Representation

Traditionally, text features have been represented using bag-of-words (BOW) model which is simple and intuitive. Each document is considered as an unordered set of V

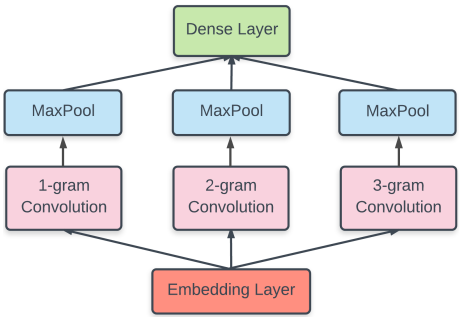

Fig. 1: Basic neural network structure

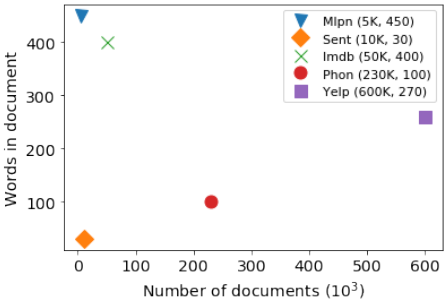

Fig. 2: Size-length distribution of datasets

(size of vocabulary) words. Those words are then vectorized and scored with different functions like binary vectorizer, count vectorizer or tf-idf vectorizer based on their presence or absence in the document. BOW and tf-idf have proved to be very effective on several text classification tasks, especially when support vector machine is used as classifier [6]. There are however various problems like data sparsity and high dimensionality that make it incompatible with the recent neural network classifiers. Feature matrix is a function of V and in high vocabulary scenarios (as in our case here) it becomes very big and sparse. Word embeddings provide a much denser representation of lower dimensionality that is independent from V. They are generated using semi-supervised methods trained on big text corpora [14]. Vectors of each word are very good in capturing syntactic and especially semantic relations of that word with the other vocabulary words appearing in same contexts. As suggested in [2], when relatively small text sets are available, obtaining word vectors pretrained from big text corpora gives better results. They work like general feature extractors applicable across different text sets. As a result, we chose to utilize static vectors of 300 dimensions obtained from GoogleNews[1] collection. They were generated from a 100-billion words bundle and their power and effectiveness has been reported in other works like [9] or [11].

3.2 Neural Network Design Alternatives

The basic network structure we experiment with is presented in Fig. 1. Embedding layer is not trainable and uses the static vectors of GoogleNews for each word appearing in the document. It is followed by the convolution layers that are responsible for feature extraction. Different studies like [16] have pointed out the performance gains of adding 2-gram and 3-gram features in sentiment classification of texts. For this reason we use three convolution layers in parallel, with kernel sizes 1, 2 and 3 and 70 feature maps to catch samples of words (e.g., awesome, "awful", etc.), 2-grams (e.g., "so great", "really bad" etc.) or 3-grams (e.g., "that was great", "not that good", etc.) and their relation with sentiment classes We also explored $relu(x) = max(0, x)$, $tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ and $softsign(x) = \frac{x}{1+|x|}$ activation function for convolution layers and used $relu$ (the best

[1] https://code.google.com/p/word2vec/

in most trials) for the rest of experiments. Max-pooling layers that follow, downsample data by effectively selecting the most salient features. Obtained feature maps are finally flattened, merged and pushed forward into the dense classification layer. A simple classifier of one dense layer with 80 nodes was used on all experiments. Overfitting was considerably reduced using 0.1 L2 regularization and 0.35 (or sometimes 0.5) dropout. Fig. 2 shows the experimental datasets with their different sizes and text lengths. To find the optimal network structure for each dataset, we considered alternatives derived from the basic version of Fig. 1 (Conv-Pool4) which has three layers of convolutions followed by three regional max-pooling layers with region size 4. A deeper alternative is obtained by duplicating the two stacks of convolutions and max-pools. The rest of the networks are similar and change only in max-pool region size (5, 16 or 25) to adapt to the different document lengths of the datasets. We used a 70-10-20 % data split for training, development and testing respectively. Obtained results are presented and discussed in the following section.

4 Results and Discussion

Table 2 presents accuracy scores of different neural networks on each dataset. We exercised network versions with 1 and 2 levels of convolution-pooling layers and different pool sizes. Top scores on smartphone reviews (Phon) and song lyrics (Mlpn) seem satisfactory. Unfortunately we have no basis of comparison for these datasets. On sentiment polarity dataset (Sent) we reached 79.89 % which is rather good. This dataset is frequently used in various studies where they usually get accuracy scores of 76 - 82 %. Best result we found in the literature is 83.1 %. It was reached by Zhao *et al.* in [20] using a self-adaptive hierarchical sentence model implemented with a gating network. On IMDB movie reviews (Imdb) and Yelp polarity reviews (Yelp) we got 90.68 % and 94.86 %. Johnson and Zhang claim to have scored 92.23 and 97.36 % in [7] and [8] respectively, using deeper and more complex networks. Despite the lowers scores, it is important to note that unlike the top-performing deep and complex architectures, our neural networks here are very simple and fast, with no more than 200,000 trainable parameters. On the other hand, carefully observing results of Table 2, we spot some interesting insights about relation between document length, max-pooling region size and accuracy. We see that datasets of longer documents (Mlpn and Imdb) perform better on networks with longer pooling regions (25 and again $5x5 = 25$). The opposite is true about Sent and Phon datasets that contain shorter documents. In fact, region size is the parameter that regulates length of the feature maps that are generated ($map\_length \approx document\_length/pool\_size$). We see that in all cases, top accuracy is achieved on feature maps with length $6 - 18$. Furthermore, we can see that bigger datasets (Imdb, Phon and Yelp) work better with the deeper network versions whereas smaller datasets (Mlpn and Sent) work better with the basic network versions. This is something we expected, since complex networks are usually more data hungry than simple ones. It is still worthy to mention that size-length surface of Fig. 2 remains mostly uncovered. Obviously, extensive experimentation with different networks and more datasets of various document lengths is required to further affirm the insights reported above.

Table 2: Accuracies of 5 network structures

Network	Mlpn	Sent	Imdb	Phon	Yelp
Conv-Pool4	72.24	**79.89**	87.98	95.31	92.32
Conv-Pool5	72.75	77.68	88.56	95.44	92.55
Conv-Pool16	73.17	75.62	89.62	96.06	92.73
Conv-Pool25	**75.63**	74.46	90.12	95.15	93.51
2 x Conv-Pool4	73.34	75.08	89.87	**96.57**	**94.86**
2 x Conv-Pool5	75.44	74.22	**90.68**	95.64	93.84

5 Conclusions and Future Work

In this work we experimented with different convolution and max-pooling neural networks for sentiment polarity analysis of different types of texts like song lyrics, movie reviews, item reviews etc. We utilized GoogleNews pretrained word embeddings that work as generic text feature extractors and a simple dense layer as classifier. The goal of our experiments was to observe performance of convolution networks on text sets of various document lengths and sizes. To this end, we selected text datasets of short sentences, brief product reviews and long movie reviews or song lyrics. Accuracy score was lower than state of art on each dataset, given that we used very fast and simple neural networks with few parameters. Our preliminary results also indicate that there is a solid relation between length of documents and length of generated feature maps with respect to sentiment classification accuracy. Best results are usually achieved when texts are reduced to feature maps of lengths 6 to 18. Regarding size of datasets, best results are achieved when using deeper networks on the bigger ones. In the near future, we aim to explore performance of convolution-based neural architectures on text collections of a higher variety of size-length combinations. The final goal is to design an architecture that yields competitive sentiment prediction accuracy in reasonable time, by adopting to text datasets of diverse sizes, lengths and contents with little change in hyper-parameters.

Acknowledgments

This work was supported by a fellowship from TIM.[2] Part of computational resources was provided by HPC@POLITO,[3] a project of Academic Computing within the Department of Control and Computer Engineering at Politecnico di Torino.

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] `https://www.tim.it`
[3] `http://hpc.polito.it`

2. E. Çano and M. Morisio. *Quality of Word Embeddings on Sentiment Analysis Tasks*. Springer International Publishing, 22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, pp. 332-338, Liège, Belgium, June 21-23, 2017.
3. E. Çano and M. Morisio. Moodylyrics: A sentiment annotated lyrics dataset. In *2017 International Conference on Intelligent Systems, Metaheuristics and Swarm Intelligence*. ACM, pp. 118–124, Hong Kong, March 2017.
4. E. Çano and M. Morisio. Music mood dataset creation based on last.fm tags. In *Computer Science & Information Technology (CS & IT)*, volume 7, pages 15–26. AIRCC Publishing Corporation, Vienna, Austria, May 2017.
5. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
6. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142, 1998.
7. R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. *CoRR*, abs/1412.1058, 2014.
8. R. Johnson and T. Zhang. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 562–570, 2017.
9. Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.
10. S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. AAAI, vol. 333, pp. 2267-2273, 2015.
11. J. H. Lau and T. Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *CoRR*, abs/1607.05368, 2016.
12. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
13. 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 - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
14. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119, Dec 2013.
15. C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.
16. S. Wang and C. D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
17. M. Zhang, Y. Zhang, and D.-T. Vo. Gated neural networks for targeted sentiment analysis. AAAI, pp. 3087-3093, 2016.
18. X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA, 2015. MIT Press.
19. X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626, 2015.
20. H. Zhao, Z. Lu, and P. Poupart. Self-adaptive hierarchical sentence model. *CoRR*, abs/1504.05070, 2015.