Sentiment Lexicon Creation using Continuous Latent Space and Neural Networks

Pedro Miguel Dias Cardoso
Synthesio / 8 rue Villedo, Paris
pedro@synthesio.com

Anindya Roy
Synthesio / 8 rue Villedo, Paris
aroy@synthesio.com

Abstract

This work presents a novel approach for automatic creation of sentiment word lists. In this approach, words are first mapped into a continuous latent space, which serves as input to a multilayer perceptron (MLP) trained using sentiment-annotated words. When evaluated using manually annotated EmoLex corpus, our approach compares favourably with SentiWordNet 3.0, another automatically generated word list.

1 Introduction

Many of the state of the art sentiment analysis systems uses input features based on sentiment word lists (Mohammad et al., 2013). While such lists may be manually curated, automatic approaches are privileged if large lists need to be generated. SentiWordNet 3.0 (Baccianella et al., 2010) is an example of such an automatically generated sentiment word list.

In this work, we present a novel approach for automatic generation of sentiment word lists. In this approach, words are mapped into a continuous latent space using two embedding methods Word2Vec (Mikolov et al., 2013a)(Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). The mappings are given as input to a MLP trained using sentiment annotated words, which outputs a sentiment class or score for each word. Results show that our method can create large sentiment lexicons with higher accuracy.

The method is based on a small annotated lexicon and large un-labeled corpora. Hence, it could be easily applied to domain-specific contexts or under-resourced languages.

2 Related Work

2.1 Word embeddings

In classical vector space representation of text, individual words directly correspond to dimensions in the feature vector. This approach does not take into account the *semantic* proximity between words. Recent advances have tried to overcome this limitation by representing words as points in continuous latent space, where proximity between points in space indicates *semantic* proximity e.g. Word2Vec. However, these systems based on word context do not necessarily encode word *sentiment* information. For example, based on word distances in a 300-dimensional latent space, the word “good” is closest to “bad” because “good” and “bad” are often found in similar contexts. In this work, we leverage the advantage of continuous latent spaces but add a MLP to infuse sentiment information.

2.2 Sentiment classification and word lists

Initially, sentiment classification was approached as a *document* classification problem. An early study of sentiment classification (Pang et al., 2002) compared Naive Bayes, Logistic Regression and Support Vector Machine (SVM) classifiers. An early example of sentiment analysis using *microblogs* is presented in (Pandey and Iyer, 2009). A comprehensive review of recent developments is presented in (Vinodhini and Chandrasekaran, 2012).

Recently, sentiment classification has seen increased use of word lists with associated sentiment

Proceedings of NAACL-HLT 2016, pages 37–42,
San Diego, California, June 12-17, 2016.

values, set either manually or automatically. Among automatic methods, (Turney and Littman, 2003) and (Esuli and Sebastiani, 2006) compute word sentiment based on context of known sentiment words using Pointwise Mutual Information. In (Mohammad et al., 2013), specific Twitter hashtags with positive or negative sentiment were exploited. In our experiments, we use SentiWordNet 3.0 (Baccianella et al., 2010). SentiWordNet was built in two steps, a first semi-supervised step and a second random walk step. Sentiment values of a set of seed words (Turney and Littman, 2003) were propagated using WordNet's binary relations.[1] Propagated labels were used to train a classifier which was applied on all words. The second step is a random walk on WordNet, where sentiment is propagated if most of the terms used to define a given term and the term itself are of a specific sentiment value. SentiWordNet is publicly available.

2.3 Word embeddings for sentiment

Works which use word embeddings for sentiment classification include (Amir et al., 2014) where the Word2Vec vector of all words present in the document are summed and used to detect sentiment. In (Irsoy and Cardie, 2014) Recursive Neural Networks are used with Word2Vec as input. In (Tang et al., 2014b) and (Tang et al., 2014a), tweets are represented using *sentiment-specific* word embeddings. Note that these approaches deal with sentiment classification of *documents*. In contrast, our work uses word embeddings with MLP for sentiment classification of *words*. Furthermore, (Tang et al., 2014b) requires *sentiment-annotated* corpora to learn latent space embeddings, while our approach uses unsupervised methods which may be trained using any suitable unannotated corpora.

3 Methodology

We created two MLP-based systems to estimate sentiment value of words. The first is a classifier system that predicts sentiment class (positive, negative, neutral). The second is a regressor system predicting a sentiment value for each word in a continuous range. The classifier and regressor systems use as inputs two different word representations in embedded continuous space: Word2Vec[2] model trained with 100 billion tokens and GloVe[3] trained with two corpora, one containing 42 billion tokens and second containing 840 billion tokens. The two represent each word in a 300-dimensional space.

Lexicon	Training Data			Evaluation Data		
	Neg	Neu	Pos	Neg	Neu	Pos
SynthesioLex	3544	2204	1500	880	552	375
EmoLex	2591	6146	1845	648	1537	462
MxDiff	260		431	62		109

Table 1: Classification lexicon size

3.1 Lexicons

For classification experiments, we use two lexicons. First one is curated by us over time and denoted in this work by SynthesioLex. It contains positive, negative or neutral sentiment class for each word. The second, EmoLex (Mohammad and Turney, 2013), is a publicly available lexicon[4]. For each word it contains a combination of sentiment tonality (positive, negative) and one of eight possible emotion classes (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) for each word. In case of neutral words, all labels for tonality and emotion are 0. We keep only words that have sentiment tonality value and neutral ones. For regression, we use publicly available MaxDiff lexicon (Kiritchenko et al., 2014)[5]. It contains words with sentiment value in a continuous range from -1 to 1 and from 0 to 1 obtained manually by crowdsourcing (Orme, 2009). Table 1 gives more details about each lexicon. Note that EmoLex is bigger, and biased towards neutral class. SynthesioLex is slightly biased towards negative class. For each lexicon, we used 80% of the data for training and 20% for evaluation, uniformly sampled.

3.2 Test Setup

At its core, the system has a MLP with 3 layers: (1) input layer with linear activation function and 300 units, the same as the word representation space. (2) hidden layer with a $\tanh$ activation function and number of units varying from 20 to 500. (3) output layer with softmax activation function with

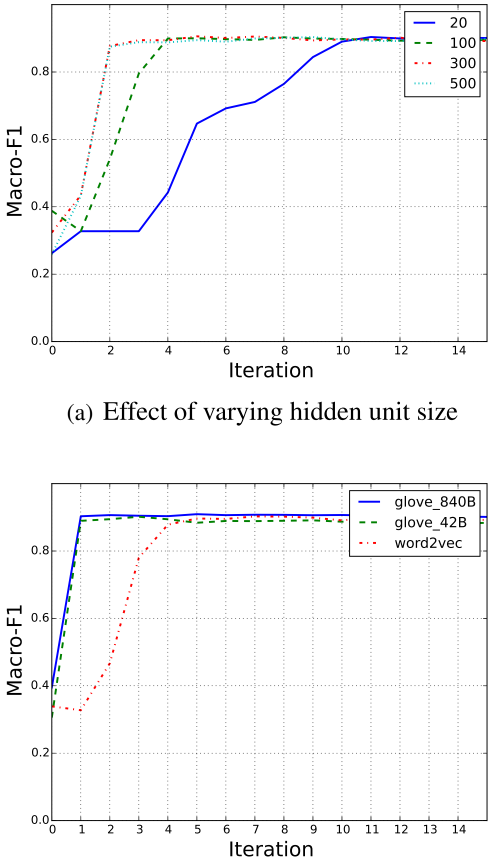
[1] http://wordnet.princeton.edu

[2] https://code.google.com/p/Word2Vec

[3] http://nlp.stanford.edu/projects/glove/

[4] http://saifmohammad.com/Lexicons/NRC-Emotion-Lexicon-v0.92.zip

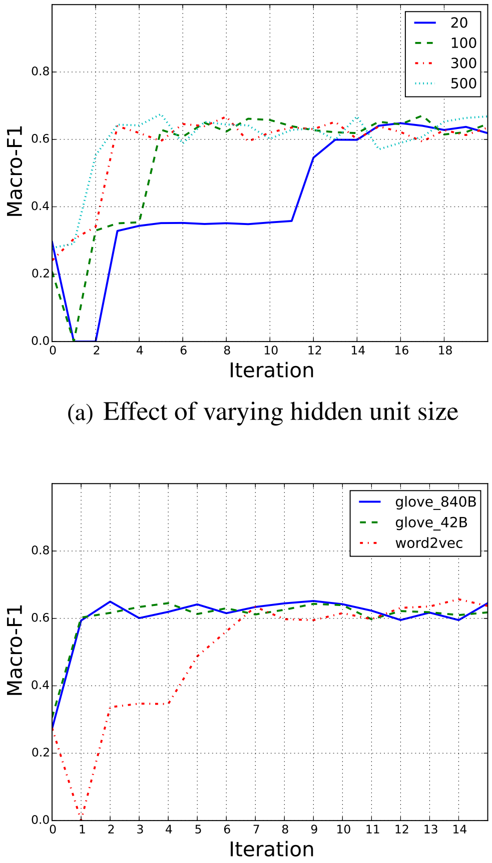
[5] http://saifmohammad.com/WebDocs/MaxDiff-Twitter-Lexicon.zip



(a) Effect of varying hidden unit size

(b) Effect of different word embeddings

Figure 1: Results on SynthesioLex



(a) Effect of varying hidden unit size

(b) Effect of different word embeddings

Figure 2: Results on EmoLex

2 or 3 units for classification, and linear function with single unit for regression. MLP training is done with mean square error as cost function using Keras library,[6] a Python package built on top of Theano[7] (Bergstra et al., 2010).

4 Results

Evaluation metric for classification is Macro-F1 between negative and positive classes. F1 measure of Neutral class is not considered, following the SemEval evaluations on twitter sentiment classification (Rosenthal et al., 2014). For regression, we use mean square error as evaluation metric.

4.1 Classification

In figure 1 and 2, we present Macro-F1 on evaluation data for each training epoch. All models converged quickly, and towards similar results. In subfigures 1(a) and 2(a) we see that convergence results are similar, independent of number of hidden units. With 20 units, model took longer to converge but it is evident that the model was able to train even with small number of units. Based on this, we used 100 hidden units for subsequent evaluations. In figures 1(b) and 2(b) we compare results for different word embedding spaces and again all converge to similar values, with Word2Vec taking longer time to stabilise. Classification results for the two lexicons are shown in Table 2. The *binary* cases represent binary classification where only positive and negative data were retained and all neutral data removed. The system performs better on SynthesioLex than on EmoLex, possibly due to higher number of neutral words in EmoLex. Results for the two lexicons are similar if we only use positive and negative sentiment classes (binary case, last two rows). This can be seen also in the confusion matrices in Tables 3

[6] http://keras.io/

[7] http://deeplearning.net/software/theano/

and 4 for SynthesioLex and EmoLex lexicons. For EmoLex, many sentiment words were classified as neutral, reducing recall. Also, when compared with SynthesioLex results, many neutral words are seen as having a sentiment, reducing precision.

Lexicon	**Word rep**	**Macro-f1**	**Best iter**
SynthesioLex	Word2Vec	90.23	6
	glove_42B	89.05	4
	glove_840B	90.96	7
EmoLex	Word2Vec	75.85	18
	glove_42B	74.33	2
	glove_840B	74.78	6
SynthesioLex Binary	Word2Vec	93.91	2
	glove_42B	93.42	2
	glove_840B	94.76	6
EmoLex Binary	Word2Vec	91.16	5
	glove_42B	90.31	4
	glove_840B	92.02	5

Table 2: Results for classification problem.

	Neg	Neu	Pos
Neg	835	29	16
Neu	29	510	13
Pos	32	23	320

Table 3: SynthesioLex confusion matrix, row: groundtruth, column: predicted.

	Neg	Neu	Pos
Neg	431	211	6
Neu	91	1317	129
Pos	14	203	245

Table 4: EmoLex confusion matrix, row: groundtruth, column: predicted

In Section 1, we mentioned that the current method can be used for automatic generation of sentiment word list. To test this proposition, we trained the MLP using SynthesioLex and compared it with another semi-supervised method, SentiWordNet 3.0. We used EmoLex lexicon for evaluation. For SentiWordNet, the sentiment value for each word was computed as average sentiment value over all possible synsets. Confusion matrix for the two methods can be seen in 5 and 6. SentiWordNet achieves very low recall for positive and negative emotion classes, predicting most words as neutral. Macro-F1 results are 44.85% for our method and 6.31% for SentiWordNet.

	Neg	Neu	Pos
Neg	94	1392	13
Neu	71	6392	28
Pos	15	1369	40

Table 5: Confusion matrix between EmoLex (row) and SentiWordNet (column)

	Neg	Neu	Pos
Neg	1120	311	68
Neu	1826	3445	1220
Pos	186	558	680

Table 6: Confusion matrix between EmoLex (row) and current method (column)

4.2 Regression

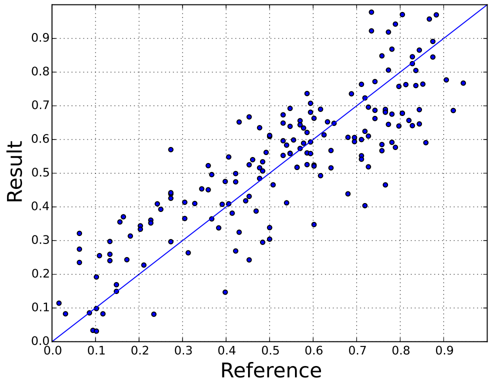


Figure 3: Regression results

For regression, we used same method for training as for classification. Optimal results were obtained with 100 hidden units. Best word representation is GloVe trained with 840 billion tokens. Lowest mean square error is 0.014. Regression results are shown in Figure 3. There is strong correlation of sentiment value between reference and result (correlation coefficient 0.85). One way to test this is to transform regressor to classifier by setting all words with sentiment value below 0.5 as negative and all with value above 0.5 as positive, keeping same training and evaluation splits. Figure 4 shows performance of such a classifier. Errors are associated with words close to decision boundary. For those whose sentiment is strongly positive or negative, there are few mistakes. This suggests setting to neutral all words whose regression values are close to decision boundary.

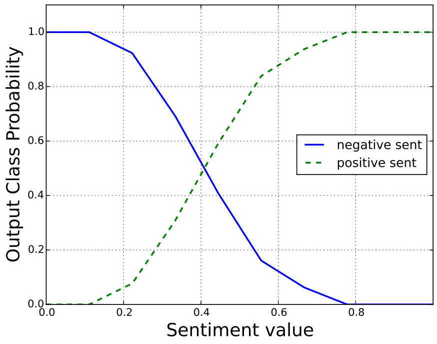


Figure 4: Performance of regressor as classifier

4.3 Discussion

Word embeddings like Word2Vec and GLoVE map "good" as closest to "bad" in latent space (ref. Section 1). MLP succeeds in mapping such opposite sentiment words far from each other at the hidden layer output space (ref. Section 3.2). In terms of cosine distance using the hidden layer of our MLP output, "good" was closest to "dignified", "compassionate", while "bad" was closest to "embarrassment", "contemptuous". "good" and "bad" were not part of the training data for this study.

5 Conclusion

Although the presented mappings do not consider sentiment information in word context, good results were obtained for word sentiment classification using these mappings as input to a MLP classifier trained and tested on two lexicons, SynthesioLex and EmoLex. When trained on SynthesioLex and tested on EmoLex, proposed approach performed better than SentiWordNet 3.0. We also studied a regression system which can be used to create features in a continuous sentiment space. This is a work in progress focusing on sentiment word list creation. In future, it is planned to integrated this approach in a complete document sentiment classification system.

References

Silvio Amir, Miguel Almeida, Bruno Martins, Joao Filgueiras, and Mário J Silva. 2014. Tugas: Exploiting unlabelled data for twitter sentiment analysis. *SemEval 2014*, page 673.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research (JAIR)*, 50:723–762.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR13)*.

Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and hb. *Sawtooth Software*.

Vipul Pandey and C Iyer. 2009. Sentiment analysis of microblogs. *CS 229: Machine learning final projects*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014a. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1555–1565.

Peter D Turney and Michael L Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

G Vinodhini and RM Chandrasekaran. 2012. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6).