The STAVICTA Group Report for RepLab 2014 Reputation Dimensions Task

Afshin Rahimi[1, 5], Magnus Sahlgren[2, 5], Andreas Kerren[3, 5], Carita Paradis[4, 5]

[1] Computer Science Department, Linnaeus University,Växjö, Sweden
Afshin.rahimi@lnu.se
[2]Gavagai AB, Stockholm, Stockholm, Sweden
mange@gavagai.se
[3] Computer Science Department, Linnaeus University, , Växjö, Sweden
andreas.kerren@lnu.se
[4] Center for Languages and Linguistics, Lund University, Lund, Sweden
Carita.paradis@englund.lu.se
[5]StaViCTA Project Group

Abstract. In this paper we present our experiments on the RepLab 2014 Reputation Dimension task. RepLab is a competitive challenge for Reputation Management Systems. RepLab 2014's reputation dimensions task focuses on categorization of Twitter messages with regard to standard reputation dimensions (such as performance, leadership, or innovation). Our approach only relies on the textual content of tweets and ignores both metadata and the content of URLs within tweets. We carried out several experiments focusing on different feature sets including bag of n-grams, distributional semantics features, and deep neural network representations. The results show that bag of bigram features with minimum frequency thresholding work quite well in reputation dimension task especially with regards to average F1 measure over all dimensions where two of our four submitted runs achieve highest and second highest scores. Our experiments also show that semi-supervised recursive autoencoders outperform other feature sets used in our experiments with regards to accuracy measure and is a promising subject of future research for improvements.

Keywords: short text categorization, sentiment analysis, reputation monitoring

1 Introduction

Twitter has become a good source of data for opinion mining systems. Not only does the length restriction of tweets (140 characters) encourage users to keep their messages concise (this is of course not always the case), the characteristics of the medium itself promote opinionated content; its simplicity, brevity, and velocity makes Twitter an ideal channel for users to express opinions about current events. Using the vast amount of data Twitter provides, there has been several attempts to apply machine learning on various applications including, but not limited to, predicting election results [20], [22], monitoring brands' reputation [7], [18] and forecasting stock prices [4] Many of these attempts rely on sentiment analysis (or opinion mining), which is usually cast as a classification problem over the categories positive, negative, and neutral [13]. However, for many applications such as Reputation Classification [1, 2] positive/negative categories are too simplistic and current interest has drifted towards more complex sentiment palletes like that of the RepTrak® model [15] that is adopted in the RepLab reputation dimensions task.

RepLab 2014 [2] is an evaluation campaign addressing the challenge of categorizing tweets related to several brands with regards to standard reputation dimensions introduced by the RepTrak® model. These dimensions/categories are:

- Products/Services
  There's a nice BMW in front of my window.... I think I'm gonna steal it.
- Innovation
  Wait! They're integrating Siri into cars. Mercedes, Honda, GM, Toyota etc.
- Workplace
  What's going on at the Nissan plant?
- Citizenship
  Ireland Tours and http://Travel.com shared Volvo Ocean Race Galway's photo. http://fb.me/1KEVvWrnt
- Governance
  Accounting experts join RBS board http://bit.ly/pHHg5Z accounting
- Leadership
  Panic at the White House? Gloomy Goldman Sachs sees high unemployment ... http://bit.ly/rbJIdI
- Performance
  Chris Whalen's Inst Risk Analytics Downgrades outlook on Goldman and Morgan Stanley
- Undefined, which covers tweets not relating to any of the other 7 categories.
  Ford music. In my car!

The rest of this article is organized as follows. Section 2 presents the dataset; section 3 summarizes our experiments with regards to the reputation dimension task; section 4 provides both submitted and unsubmitted results, and section 5 briefly concludes our work and discusses future improvements.

2 Dataset

RepLab 2014 uses Twitter data in English and Spanish. For the reputation dimensions task the dataset is the same as in RepLab 2013 and consists of a collection of tweets related to 61 entities/brands in four different industries. The RepLab 2014 dataset only uses tweets in the automotive and banking subsets. For each entity at least 2200 tweets are downloaded and annotated from which 700 tweets are used for the training set and the last 1500 tweets are reserved for the test set. As Twitter terms of service does not permit distribution of tweet contents, the id of tweets are provided to be used in retrieving tweets directly from Twitter. However, since some tweets may have been deleted or changed to private by users, the actual number of retrieved tweets will possibly be lower than the initial number of annotated tweets. Training tweets are categorized with regards to the 8 mentioned categories.

For each entity there are a number of uncategorized background tweets that can be used in different ways (e.g. for unsupervised feature learning).

3 Approach

We performed several experiments to evaluate the performance of various feature sets and various classification algorithms for the reputation dimension task. The feature sets we used in these experiments can be roughly categorized into the following 3 groups with regards to representation type: bag of words representations, distributional representations and deep neural network representations.

3.1 Bag of Words Representations

Bag of words is arguably the most common form of representation of textual content; each text is represented as a feature vector where the elements record (some function of) the frequencies of the words in the text. Although there have been many attempts to devise more sophisticated forms of text representations, bag of words representations have remained the standard form of text representation for classification purposes. The main reason for this is their simplicity, coupled with the fact that they produce competitive results not only in text classification, but also in many other tasks such as information retrieval, clustering, question answering, etc. The main drawback of these models is the very assumption that makes them so simple: Assuming that we can have a representation of a piece of text by considering it as a bag of words and that we can completely ignore the sequence and the structure of the words in a text in favor of the simplicity of representation. To relax this overly naïve assumption, we used bag of n-gram models to incorporate local sequential and structural information up into the representation.

Unigrams, bigrams, trigrams and 4-grams were used in different experiments. Previous research has shown that in some tasks unigrams perform better than higher order n-grams [13]. Using bigrams and higher order n-grams as features in text classification tasks introduces a lot more new rare features many of which occur in just one

or two documents especially when the training data is not very big. These rare features have very high Inverse Document Frequency (IDF) because they occur in few documents which means that they will get high scores using TF-IDF weighting:

$$TFIDF(t,d,D) = \frac{n(t,d)}{\sum_{w \epsilon d} n(w,d)} \times \frac{\log |D|}{|d : t \in d|}$$

Where t is a term, d is a document and D is the document collection. The first product term is Term Frequency (TF) and the second product term is Inverse Document Frequency (IDF). Here n(t, d) is the number of times term t occurs in document d, |D| is the total number of documents/tweets and |d| is the number of documents/tweets that a term is occurred in. As is shown in the formula rare terms have low d and so result in a big IDF. These new features introduce a lot of noise to the classification task and consequently decrease accuracy. In order to alleviate this problem one working solution is to use a minimum document frequency with which the features that occur in just few documents are removed before a TF-IDF transformation. We found that removing the n-grams that occur in just one document improves accuracy in the RepLab dataset. To prevent over-fitting a 10-fold cross-validation was used and the resulting accuracies were averaged into an overall accuracy for each setting.

In addition to using words, we tried to enrich our feature sets with named entities. As an example, in the sentence *John left Ford* was converted to *Person left Organization* in order to get more generalized features. We used the Stanford named entity recognizer tool [5] to tag both training and test set to be used later as features. We used named entities both as extra features and as a replacement for the named entities of tweets.

3.2 Distributional Representations

Distributional Semantic Models (DSM) are word representations that try to capture the semantic similarity of words using their distribution in language. The idea which is known as the Distributional Hypothesis is that words with similar distribution have similar meaning [17]. Here the word 'distribution' means the collection of occurrences of a word within a context where context can be a very narrow window of size 1 around that word or a large textbook the words occur in.

As input to the DSM, we concatenated English Wikipedia, Spanish Wikipedia, the RepLab training and background sets. It should be noted that the test set was not included in the corpus. we used the Random Indexing framework, which is an efficient method for building DSM models for big data, since it uses fixed-dimensional vectors whose dimensionality is much lower than the representational dimensionality of the data [8]. We used Random Indexing with 2048-dimensional vectors, and documents (Wikipedia articles or tweets) as word contexts. After building the model we used Positive PMI to normalize weights in order to disfavor highly frequent words.

To come up with a vector-based representation for each tweet out of the word representations two different approaches were applied: summing word vectors and concatenating word vectors. In the summing approach the representations of words of a tweet were fetched from the DSM model and summed to form a 2048-dimensional compositional vector representing the semantic content of that tweet. Vector addition is a very simple but comparatively effective approach to form compositional DSM representations [6]. In the concatenation approach we concatenated the first 20 word representations of each tweet. If a tweet had less than 20 words zero valued vectors were concatenated at the end the vector, resulting in 40960-dimensional vector representing each tweet. The tweet vectors were then used as features of the training and test sets. We also carried out an experiment with a combination of both bag of words feature set and DSM feature set. In our second approach

3.3 Deep Neural Network Representation

Deep Neural Networks are producing state of the art results in many Machine Learning fields including Computer Vision, Speech Recognition, Natural Language Processing and Music Recognition. Recursive Autoencoders have been shown to produce good results in sentiment analysis tasks [21]. We reproduced Socher et al.'s experiment with the reputation dimensions dataset. We also used Theano's [3] implementation of Deep Belief Networks in order to compare the abstract feature sets provided by these deep representations to bag of words.

4 Results

We used scikit-learn [14], a collection of simple and efficient tools for machine learning in Python, for doing feature extraction, weight normalization, and classification. The deep learning experiments are evaluated by partitioning the training set into two random train and test sets by ratio of 9 to 1. Other experiments have been evaluated by 10-fold cross validation. The gold standard final test set consisted of 7 unbalanced categories (excluding undefined category). The distribution of tweets in these 7 categories is shown in table 1.

Table 1. The distribution of classes in the gold test set

Category	Percent
Products & Services	56.60034879
Citizenship	17.89158985
Governance	12.08314055
Performance	5.68743994

Workplace	4.000427092
Leadership	2.647969534
Innovation	1.089084244

Table 2 summarizes the main results of our experiments. As can be seen in table 2 the bag of bigram model outperforms the DSM model and LinearSVC outperforms other classifiers. The only classifier that works better than LinearSVC with bigram features is socher-recursive-autoencoders which achieved a high accuracy of 0.83 but because we did not evaluate the model by cross-validation we did not submit that run for the task. Final results shown in table 3 indicate that our models outperform the baseline model with regards to both accuracy and f measure and also perform close to the best results from other participants (uofTr_RD_4, DAE_RD_1 and LyS_RD_1). Some runs including uofTr_RD_4, DAE_RD_1 and LyS_RD_1 perform better than our runs with regards to accuracy but our runs perform better with regards to macro averaged f measure. Given the skewed distribution of categories in table 1 it is important for a classifier to perform well with regards to f measure too because if someone just classified all tweets in Products & Services class it would achieve about 56% accuracy. The final results in table 3 show that all our runs which use bag of bigram models perform quite well with regards to F measure and in the same time achieve reasonable accuracies too.

Table 2. Experimental results for RepLab 2014 reputation dimension classification

Method	**Accuracy**	**F1**
BoW-unigram-RidgeClassifier	0.758	0.742
BoW-unigram-LinearSVC	0.760	0.750
BoW-bigram-RidgeClassifier	0.766	0.750
BoW-bigram-LinearSVC	**0.770**	**0.759**
BoW-bigram-PassiveAggressive	0.755	0.749
BoW-bigram-MultinomialNB	0.750	0.742
MultinomialNB-bigram-NER	0.740	0.728
BoW-trigram-RidgeClassifier	0.764	0.748
BoW-trigram-LinearSVC	0.767	0.756

Socher-recursive-autoencoder	**0.83**	-
Theano-DBN-3layer-1000node	0.49	-
DSM-sum	0.675	0.638

Table 3. Final evaluation of submitted runs over test data

Method	**Accuracy**	**Macro Averaged F1**
baseline-dimensions-bow-presence-SVM	0.622	0.380
uogTr_RD_4*	0.731	0.473
DAE_RD_1*	0.723	0.390
LyS_RD_1*	0.716	0.477
run1: BoW-bigram-LinearSVC	0.695	**0.489**
run2: BoW-bigram-MultinomialNB	0.685	0.475
run3: BoW-bigram-PassiveAggressive	0.661	0.482
run4: BoW-bigram-RidgeClassifier	**0.703**	0.469

*The best results from other participants with regards to accuracy

5 Conclusion

Our goal in these experiments was to evaluate different feature sets with regards to the reputation dimensions task. We carried out several experiments with bag of word representations, DSM representations and deep learning representations. Our results show that higher order n-gram features such as trigrams do not perform better than bigrams. We assume the reason for this is data sparseness; higher-order n-grams pro-

vide more specific features, but if the data is not big enough (i.e. if the occurrence counts of the n-grams are uncertain) they will only introduce noise to the representations. They also show that in order to reduce noise introduced by bigrams, minimum frequency thresholding should be applied. Removing bigrams that occur just once in the corpus is the best minimum threshold on the RepLab dataset and this improvement resulted in highest and second highest scores in the RepLab reputation dimensions challenge with regards to average F1 over all dimensions. We also used named entity features in several experiments but the resulting accuracy was lower than not using them at all. In [19] similar results are reported both for replacing NER features with real names and for adding them to bag of word models. Although named entity features resulted in lower accuracy the generalized features they provide is a good subject of future research in domain adaptation tasks.

As our results show the DSM representations do not perform better than bag of word models. Although such models can encode semantic content, summing or concatenating them is shown here not to perform well in the reputation dimensions task. However, recent works [10, 11, 12] indicate that word vectors produced by neural network-based models can be used to improve text representations for classification results. The composition of word vectors into sentence/document vectors is another subject of future research.

While Deep Belief Networks did not produce good results, semi-supervised recursive autoencoders [21] performed quite well according to accuracy measure. We did not submit deep learning results in the RepLab challenge but as the results show they can produce promising representations and consequently are a subject of future research.

Acknowledgement

This work has been funded through the project StaViCTA by the framework grant "the Digitized Society – Past, Present, and Future" with No. 2012-5659 from the Swedish Research Council (Vetenskapsrådet).

References

1. Amigó, Enrique, et al. "Overview of replab 2013: Evaluating online reputation monitoring systems." *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*. Springer Berlin Heidelberg, 2013. 333-352.
2. Amigó, Enrique, et al. "Overview of replab 2014: Author profiling and reputation dimensions for Online Reputation Management." *Proceedings of the Fifth International Conference of the CLEF Initiative*. Sep. 2014.
3. Bergstra, James et al. "Theano: a CPU and GPU math expression compiler." *Proceedings of the Python for scientific computing conference (SciPy)* Jun. 2010.
4. Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." *Journal of Computational Science* 2.1 (2011): 1-8.
5. Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* 25 Jun. 2005: 363-370.

6. Guevara, Emiliano. "Computing semantic compositionality in distributional semantics." *Proceedings of the Ninth International Conference on Computational Semantics*. Association for Computational Linguistics, 2011.
7. Jansen, Bernard J., et al. "Twitter power: Tweets as electronic word of mouth." *Journal of the American society for information science and technology* 60.11 (2009): 2169-2188.
8. Kanerva, Pentti, Jan Kristofersson, and Anders Holst. "Random indexing of text samples for latent semantic analysis." *Proceedings of the 22nd annual conference of the cognitive science society* Aug. 2000: 1036-7.
9. Karlgren, Jussi, and Magnus Sahlgren. "26 From Words to Understanding." (2001).
10. Le, Quoc V., and Tomas Mikolov. "Distributed Representations of Sentences and Documents." *arXiv preprint arXiv:1405.4053* (2014).
11. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
12. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in Neural Information Processing Systems*. 2013.
13. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002.
14. Pedregosa, Fabian et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12 (2011): 2825-2830.
15. Ponzi, Leonard J., Charles J. Fombrun, and Naomi A. Gardberg. "RepTrak™ pulse: Conceptualizing and validating a short-form measure of corporate reputation." *Corporate Reputation Review* 14.1 (2011): 15-35.
16. Sahlgren, Magnus. "An introduction to random indexing." Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 16 Aug. 2005.
17. Sahlgren, Magnus. "The distributional hypothesis." *Italian Journal of Linguistics*20.1 (2008): 33-54.
18. Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." *The Semantic Web–ISWC 2012* (2012): 508-524.
19. Saif, Hassan, Yulan He, and Harith Alani. "Alleviating data sparsity for twitter sentiment analysis." CEUR Workshop Proceedings (CEUR-WS. org), 2012.
20. Sang, Erik Tjong Kim, and Johan Bos. "Predicting the 2011 dutch senate election results with twitter." *Proceedings of the Workshop on Semantic Analysis in Social Media* 23 Apr. 2012: 53-60.
21. Socher, Richard et al. "Semi-supervised recursive autoencoders for predicting sentiment distributions." *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 27 Jul. 2011: 151-161.
22. Tumasjan, Andranik et al. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment." *ICWSM* 10 (2010): 178-185.