A Framework for Humor Recognition from Social Media using Word Embeddings

Sai Krishna R[1], Brij Mohan Lal Srivastava[1], Sai Sirisha Rallabandi[1], Ayushi Pandey[2] and Suryakanth V gangashetty[1]

{*saikrishna.r,brijmohanlal.s,sirisha.rallabandi*}*@research.iiit.ac.in*, ayuship.09@gmail.com, svg@iiit.ac.in

[1]International Institute of Information Technology, Hyderabad

[2]English and Foreign Languages University, Hyderabad

ABSTRACT

We propose a word embedding approach for the recognition of humour in social media data. Conventional methods aim at modeling the text structure and hand crafting features surgically designed to emulate linguistic tendencies in specific types of jokes. We try to take a more hands off approach and use a data driven approach based on the word embeddings to design a complete framework. The experimental results show substantial improvements and outperform the baseline.

General Terms

Computational Humor, Recursive Neural Networks, Word Embedding

1. INTRODUCTION

Automatic humour recognition is a small but growing area in the field of Information Retrieval.The complexity of the problem lies in identifying the constituting element of the joke that delivers humour. Reyes et al. [9] prove that features like ambiguity (morphosyntactic and semantic), perplexity and sentence complexity can be used to distinguish between humorous and non-humorous one-liners. Their results show that humorous one-liners show a higher degree of ambiguity for all these features. Strapparava et al.[11] study automatic classification of humour of one-liners using stylistic features, content-based features or both in a learning framework. Stuart et al. [12] proposes a template-match based recognition system using Ontological System Technology and constructionist approaches. Zhang et al. [15] propose a systematic method to derive the correlation between humor-related aspects and features from humor-related theories.

Although each of these techniques work quite well on a limited domain, they rely heavily on hand crafted features. Recently, vector space models of word embeddings have been very successful at learning both the lexical as well as semantic information and are applied to a variety of useful natural language processing applications such as search query expansions [2], quick extraction for IR [6], automatic annotation [8]. In all these models, the meaning of a word is encoded as a vector computed from the co-occurence statistics of a word and its neighboring words.They are shown to capture the syntax [4],[1],[13]. These models can be extended to automatic humor recogniton.

In this paper, we propose a framework that uses word embeddings learnt from a recursive neural network for automatic recognition of humor from social media data. In Section 2, we briefly describe the word embedding scheme and in Section 3 we describe the model design, followed by experimental results and conclusion.

2. EMBEDDING

Embedding refers to the process where words from the vocabulary (and possibly phrases thereof) are mapped to vectors of real numbers in a low dimensional space, relative to the vocabulary size (”continuous space”).

2.1 Word Vector Representations

The following are the various publicly available pretrained word vectors which can be used to test the applicability of any task which uses embedding. These representations exhibit a balanced mix between large and small amounts of unlabeled text as well as between the neural and spectral methods of training word vectors.

2.1.1 Glove Vectors

Global vectors for word representations [7] are trained on aggregated global word-word co-occurrence statistics. These vectors were trained on 6 billion words from Wikipedia and English Gigaword, and are of length 300.

2.1.2 SkipGram Vectors

Mikolov et al. [5] proposed the Word2Vec tool. In this model, Huffman code of each word is used as an input to a log-linear classifier with a continuous projection layer and words within a given context window are predicted. The available vectors are trained on 100 billion words of Google news dataset and are of length 300.

2.1.3 Global Context Vectors

These vectors are learned using a recursive neural network [10] that incorporates both local and global (document-level) context features [3]. These vectors were trained on the first 1 billion words of English Wikipedia and are of length 50.

2.2 Semantic Lexicons

A lot of optimization techniques have been proposed to improve the representation of the vectors based on the task at hand. As the humor related tasks involve common sense knowledge in addition to mere representation, we use the following semantic lexicon to improve the representation of word vectors.

2.2.1 Concept Net

ConceptNet is a semantic network containing lots of things computers should know about the world, especially when understanding text written by people. It is built from nodes representing words or short phrases of natural language, and labeled relationships between them. (We call the nodes ”concepts” for tradition, but they'd be better known as ”terms”.) These are the kinds of relationships computers need to know to search for information better, answer questions, and understand people's goals.

3. MODEL

As we grow old, our brain collects memories and trains itself to predict the outcome of events beforehand. For example, if a person is moving towards the door, we expect him to open/close the door. We do not realize but the cognitive model of brain prepares itself with a few choices, and that forms the obvious or expected turn of events. With this prior information, we can define ”humor” as the turn of events which are unexpected by this cognitive model. As soon as there is an event which do not follow the predictions of brain, we experience a peculiar sort of muscle movement which can be characterized by laughter, giggle or smile. We are trying to capture the ambiguity just mentioned by suitable modeling the words as vectors in a higher dimensional space and thus come up with a classifier which can distinguish between humorous and non-humorous sentences based on the distribution of the word vectors.

In order to identify if the word embeddings in isolation can be used to predict the humorous nature of the sentences, we conducted initial experiments using the embeddings and tried to classify a segment of the data using Support Vector Machines and Multi-layer Perceptrons.We collected one-liners from twitter using Twitter API and sentences from the book ”History of Julius Caeser” as humorous and non humorous data respectively. We obtained the word vectors for each word that correspond to humor as well as non-humor and tried to classify them. We've used three word and 5 word contexts as representations but there was no significant improvement. We present the best obtained results in Table 1.

3.1 Results of Initial Experiment and Inference

From the table 1, it is apparent that mere word representations do not serve as useful tools for discrimination of humorous and non humorous data. In order to gain insight, we clustered the humor and non-humor words separately and observed that although these two classes have center of densities towards opposite ends, on an average, they form a non-separable cluster. This may be due to high volume of common data across the classes. Though we have huge amount of common words across these classes, we cannot remove them as they form an integral part to generate humor and unexpectation.

Table 1: Classification Accuracy of one-liners vs Biography

Exp	SVM	MLP
Accuracy	27.15	29.16

Also, single word vector models are severely limited since they do not capture compositionality. The dominant approach for building representations of multi-word units from single word vector representations has been to form a linear combination of the single word representations, such as a sum or weighted average. These approaches can work well when the meaning of a text is literally ”the sum of its parts”, but fails when words function as operators that modify the meaning of another word which usually happens in the case of humor. This inference leads to the clarity that we cannot use a linear classifier or word-level models for this task and is proved by the experiments we conducted using SVMs and Multi-layered Perceptrons. Hence we investigate the performance of deep neural networks like RNN(Recursive neural Networks) to achieve a non-linear separation.

3.2 Recursive Neural Network

In this section, we provide a brief intorduction to Recursive Neural Networks. Unlike standard neural networks, recursive neural networks (RNNs) are able to process structured inputs by repeatedly applying the same neural network at each node of a directed acyclic graph(DAG). In the past they have only been used in settings where another (often symbolic) component was first used to create directed acyclic graphs. These DAGs were subsequently given as input to the RNN. In such a setting, each non-leaf node of the DAG is associated with the same neural network. In other words, all network replications share the same weights. The inputs to all these replicated feed-forward networks are either given by using the child's labels to look up the associated representation or by their previously computed representation.

The models alternate between two stages:

- (1) Forward pass – recursively construct morpheme trees (cimRNN, csmRNN) and language model structures (csmRNN) to derive scores for training examples, and
- (2) Back-Propagation pass – compute the gradient of the corresponding objective function with respect to the model parameters.

3.3 Semantic Compositionality using Matrix Vector Recursive Neural Network (MV-RNN)

In the curent context , compositionality is the ability to learn compositional vector representations for various types of phrases and sentences of arbitrary length. The vector captures the meaning of that constituent. The matrix captures how the model modifies the meaning of the other word that it combines with. A representation for a longer phrase is computed bottom-up by recursively combining the words according to the syntactic structure of a parse tree. This model provides a new possibility for moving beyond a linear combination through use of a matrix W that multiplied the word vectors (a, b), and a nonlinearity function $g(.)$ (such as a sigmoid or tanh). This function is recursively applied inside a binarized parse tree so that it can compute vectors for multi-word sequences.

The advantages of MV-RNN are:

- Assigns a vector and a matrix to every word
- Learns an input-specific, nonlinear, compositional function for computing vector and matrix representations for multi-word sequences of any syntactic type.

Assigning vector-matrix representations to all words instead of only to words of one part of speech category allows for greater flexibility which benefits performance. If a word lacks operator semantics, it's matrix can be an identity matrix. However, if a word acts mainly as an operator, such as ”sadly”, it's vector can become close to zero, while its matrix gains a clear operator meaning.

3.4 Improving Lexical Embeddings with Semantic Knowledge

The word embeddings obtained may not capture the desired semantics required and hence it makes sense to impart prior knowledge from semantic resources to learn improved lexical semantic embeddings. Suppose we have a resource that indicates relations between words. In the case of semantics, we could have a resource that encodes semantic similarity between words. Based on this resource, we learn embeddings that predict one word from another related word. We use the Relation Constraint model based joint training approach proposed in [14] with ConceptNet and WordNet as resources to improve the obtained embeddings.

4. DATA

As there is no gold standard in computational humor recognition, we performed our experiments on the corpus used by previous works in the domain, i.e [9] and [15]. In [9], the authors use 3.8 million comments retrieved from SlashDot news Website. Comments on SlashDot are categorized in a community-driven process. The comment categories include the following tags: funny, informative, insightful, interesting, off-topic, flamebait, and troll. In [15], the authors have used tweets from social media Twitter. They segregate the data as humorous tweets, humorous non tweets (from textfiles.com) and non-humorous tweets.

5. EXPERIMENTS

We've analysed the performance of the proposed framework against both the previous mentioned frameworks, using the same data as mentioned. We've used word representations derived from the Neural Network language model for the experiments although the representations from the other two mentioned models could be considered as well.

Table 2: Classification Accuracy of Funny vs Informative

Exp	Bayes	SVM	REPTree	MVRNN
s1	57.15	57.16	57.16	78.55
s2	57.35	57.38	57.36	82.46
s3	58.03	57.38	57.29	79.23
s4	58.26	57.94	58.31	78.10

Table 3: Classification Accuracy of Funny vs Insightful

Exp	Bayes	SVM	REPTree	MVRNN
s1	62.19	62.25	62.25	90.14
s2	62.66	62.43	62.74	88.78
s3	62.39	62.52	62.92	81.55
s4	63.08	62.97	63.52	87.24

5.1 Experiment 1: Web Comments

We've used the same evaluation criterion and experimental design as that of [9] and compared the quality of our results against the reported results in each. We mention the experiments and the procedure here. The training sets contain 100,000 comments per class, the test sets contain 50,000 comments per class. Each classifier is evaluated using different sets of features. The following schema summarizes the features and the order in which they are assessed:

- s1 sexual-content and semantic ambiguity
- s2 sexual-content, semantic ambiguity, and polarity
- s3 sexual-content, semantic ambiguity, polarity and emotions
- s4 all features

All classifications experiments consider the classes funny versus informative, insightful, and negative respectively. The Tables 2-4 comprise the results.

5.2 Experiment 2: Twitter Data

The authors of [15] have made the data publicly available and hence we could run the evaluation on the same data. The authors have used Gradient Boosted Regression Trees for classification.

We can clearly see from the results in Tables 2-4 and 5 that Matrix-Vector Recursive Neural Networks using embeddings outperfrom baseline. The reason for this maybe due to the nature of sentences. The temporal information is highly vital to represent the semantics of a sentence and it is well captured and modeled by MV-RNN. This also can be attributed to the sematic relations captured inherently in the continuous space distribution of the Matrix Vector model of Recursive Neural Network. As humor is a highly cognitive aspect,it may not depend not just on a mere set of

Table 4: Classification Accuracy of Funny vs Negative

Exp	Bayes	SVM	REPTree	MVRNN
s1	60.37	60.36	60.37	92.54
s2	60.54	60.41	60.54	88.14
s3	60.13	60.37	60.54	91.44
s4	60.48	60.89	60.89	88.25

Table 5: Classification Accuracy of GBRT vs MVRNN

Exp	GBRT	MVRNN
Accuracy	0.817	0.824
F1	0.812	0.810

features/word representations, but rather on the relation between the features/words. Therefore, models like MV-RNN which capture the latent relationship between a pair of words intuitively perform better compared to a set of features. However, the features mentioned in [15] can be used as constraints while training the model itself to improve the performance.

6. CONCLUSION

This paper presents a framework to model humorous content on social media into a coherent data-driven architecture using various classifiers. We tabulated a series of experiments performed with a wide range of input data and variety of network parameters for RNN. We tried to find the best classifier for this task which happens to be MV-RNN. This work shall be further extended to recognize similar humorous sentences using paraphrasing techniques and cluster them under well-defined topics using topic modeling.

7. REFERENCES

[1] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[2] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396. ACM, 2006.

[3] 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*, pages 142–150. Association for Computational Linguistics, 2011.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[5] 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*, pages 3111–3119, 2013.

[6] M. Paşca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics, 2006.

[7] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.

[8] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011.

[9] A. Reyes, M. Potthast, P. Rosso, and B. Stein. Evaluating humour features on web comments. In *LREC*, 2010.

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

[11] O. Stock and C. Strapparava. Automatic production of humorous expressions for catching the attention and remembering, 2006.

[12] L. M. Stuart. Constructions for joke recognition. In *2012 AAAI Fall Symposium Series*, 2012.

[13] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[14] M. Yu and M. Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 545–550, 2014.

[15] R. Zhang and N. Liu. Recognizing humor on twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 889–898. ACM, 2014.