# "Is This A Joke?": A Large Humor Classification Dataset

**Faraz Faruqi**
Manipal Institute of Technology
Manipal
faruqi.faraz@gmail.com

**Manish Shrivastava**
IIIT Hyderabad
Hyderabad
m.shrivastava@iiit.ac.in

## Abstract

Humor is an essential characteristic of language. It has been a topic of research in linguistics and philosophy from historical times. In computer science, computational humor, as a part of Natural Language Processing, is a growing area of research. Social Media is rapidly growing as a platform for communication but processing of social media, owing to its semantic perplexity, is still a challenge. These two facts lead us to present a novel dataset for humor classification which captures diversity in humor on web resources. The large size of this dataset is to meet the data requirements for modern machine learning algorithms. This paper also deals with creating a model for detecting and analyzing humor in social media text extracted from eclectic sources on the Internet.

## 1 Introduction

Humor is one of the most interesting aspects of human language and behavior. Humans have an innate sense of humor. They can understand, interpret and create humor almost effortlessly. But even though it is a part of our daily conversations, computationally detecting and analyzing humor remains a challenge. In recent years, the study of humor has also developed into an area of computational research under computational linguistics (Friedland and Allan, 2008; Mihalcea and Pulman, 2007; Kiddon and Brun, 2011).

Among all the theories of humor (Attardo, 2010), one of the most widely accepted is the *'Incongruity Theory'* (Morreall, 1986). It suggests that humor is due to the mixing of two disparate interpretation frames in one statement.It has recently been formalized as a necessary condition for humor and used as a basis for the Semantic Script-based Theory of Humor (SSTH) (Raskin, 2012).

*Two fish in a tank. One turns to the other and says: "Do you know how to drive this?"*

The incongruity theory can also be explained as a theory of comprehension. As the joke gradually evolves, two linear train of thoughts emerge, leading to the obvious and *latent* meanings respectively. As the joke nears its end, with a clever play of words, the latent meaning becomes the dominant one and ends up being the punch line of the joke. In the preceding example, the first sentence has two connotations. "*Fish* in a Tank" and "Fish in a *Tank*". By taking the first interpretation, reader assumes the tank in question to be a fish tank. But the statement - *"Do you know how to drive this?"*, suddenly converts the connotation to one where the 'tank' in question is an 'armored car'. This creates a sense of surprise and makes the sentence humorous. Jokes based on the incongruity theory use clever wordplay to elucidate humor in a sentence. They are also short in length, making them particularly suitable for an automatic learning setting. Such kind of humor is popular among the ubiquitous social media websites, and dedicated webpages. But it is essential to standardize this data before using it in an automatic setting.

## 2 Related work

Previous work in computational humor had focused mainly on the task of humor generation (Binsted and Ritchie, 1997; Stock and Strapparava, 2003), and there has been a relatively recent paradigm shift towards humor detection. Previous researchers (Purandare and Litman, 2006; Mihalcea et al., 2010) have used a set of linguistic features to detect them (polysemy, alliteration, antonyms and adult slang etc). Kiddon and Brun



*Proc. of ICON-2018, Patiala, India. December 2018* ©*2018 NLPAI*, pages 109–114

(2011) recognized a subproblem (*Double Entendre Identification*) and constructed models to detect sexual euphemisms or wordplay in sentences. Similarly, Purandare and Litman (2006) analyzed humorous spoken conversations from a classic comedy television show - *"FRIENDS"*, by examining acoustic-prosodic and linguistic features. Taylor and Mazlack (2004a,b) considered a restricted set of all possible jokes that had wordplay as a component and examined the limited domain of *"Knock Knock"* jokes. Also, there have been other interesting researches, such as by Yang et al. (2015) where the authors developed models to extract humor 'anchors' from the sentences.

In the domain of humor-analysis on social media, Barbieri and Saggion (2014) developed automatic models to detect irony in sentences from Twitter; models developed by Davidov et al. (2010) did a semi-supervised recognition of sarcasm on Twitter and Amazon reviews. By taking the context of the tweet into account while classifying, Bamman and Smith (2015) obtained higher accuracies in detection of sarcasm in tweets.

There have not been many efforts to use deep learning methods in humor detection, owing to the subjectivity of the task and requirement of huge amount of data. de Oliveira and Rodrigo (2015) used RNN and CNN models to detect humor. But this work was in the limited domain of Yelp Reviews, and we have tried to extend this problem to the language used in social media.

## 3 Dataset

As rightly specified by de Oliveira and Rodrigo (2015), there is no large body of work on so-called *"computational humor"*. Work that exists is largely in the pure NLP domain and uses hand-written features and simplistic tree methods or SVMs (Mihalcea and Strapparava, 2006; Yang et al., 2015). This is because there is no such labeled corpus of funny texts available for a detailed semantic analysis. This problem can also be attributed to the subjectivity of assigning a binary outcome onto something as complex as humor (Bamman and Smith, 2015).

We also felt that a dataset was needed that contained balanced counts of positive and negative samples of humor. The previous dataset created by Mihalcea and Strapparava (2005) contains 16000 one-liners, obtained through bootstrapping method. Their work on this dataset in (Mihalcea and Strapparava, 2006) obtained the state-of-the-art accuracy in humor detection. de Oliveira and Rodrigo (2015), have extracted sentences from *Yelp Review* dataset. The 16000 one-liner dataset, although containing balanced samples, was too small to train a large network. The *Yelp Review* was also inapplicable in this case as it contained very long samples, extending for more than one sentence. Hence, the authors created this new dataset containing huge number of one-liner jokes, containing 400,000 sentences extracted from dedicated humor pages on various social media websites.

### 3.1 Our Dataset

This dataset contains 400,000 sentences extracted from social media and humor-dedicated websites. We used Reddit's PRAW API[1] Twitter's REST API[2] (Makice, 2009) to extract samples from various humor-dedicated pages from these sources. We also used Web Scraping (Munzert et al., 2014) method to extract a large amount of sentences from various dedicated websites like *jokeoftheday.com*, *wocka.com*, *short-funny.com*, *onelinefun.com*. Permission for scraping were taken from website owners and maintainers whenever necessary.

The negative samples were carefully chosen for testing the model on multiple settings. The hypothesis was that as the semantic similarity between the negative and positive samples increases, the accuracy of classification decreases. The sources for negative samples are:

1. News headlines from Reuters' News Agency website spanning a period of 5 years [3].
2. British National Corpus (Consortium et al., 2012)
3. Proverbs (Extracted from an online proverb collection)

## 4 Preprocessing and Balance

The sentences found on different sources were slightly different in terms of style and content. Thus, for the proper training of models, a standardization procedure was followed, which has summarized below.

The sentences containing any sort of code-mixing, image or hyperlink were removed.For

[1] https://praw.readthedocs.io/en/latest/
[2] https://developer.twitter.com/en/docs
[3] https://www.thomsonreuters.com/en.html

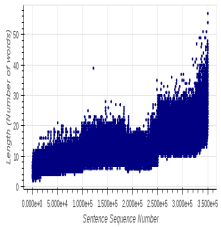
standardizing the usage of punctuation, we limited the number of punctuation marks to a max of three repetitions concurrently. CamelCase words were separated into distinct words (Friedl, 2002). Digits were separated from the alphabets using regular expressions. The non-humor samples were normalized similarly. In the Reuters dataset, highly repetitive phrases like *'Stock Market value'* and *Breaking News* were removed.

In order to cross-verify the integrity of the dataset, we randomly sampled 100 instances of humor from each source, and confirmed whether they were indeed humorous or not.

The following table contains statistical information about the Humor, Non-humor corpus and the 16000 sentences dataset (Mihalcea et al., 2010). The statistics are in terms of **Sentence Sequence number** (x-axis) vs **Number of words** (y-axis) of each sentence. It can be inferred from the data presented that the distribution of the previous dataset was much more balanced in terms of lengths of jokes. We did not maintain such a balance in our dataset as it would inhibit the model's capacity of capturing the kind of humor encountered in the real world and the social media. We intentionally included sentences of different lengths to aid diversity in the dataset.

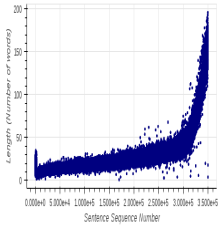
***HUMOR-***

**Minimum Length**: 3
**Maximum Length**: 250
**Mean**: 30
**Median**: 22.16
**Mode**: 16
**Population Variance**: 698.28



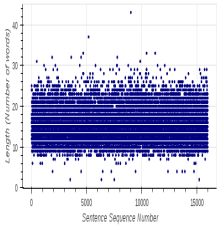
***NON HUMOR-***

**Minimum Length**: 2
**Maximum Length**: 257
**Mean**: 11
**Median**: 10.69
**Mode**: 9
**Population Variance**: 22.92



***16000 One-liners dataset***

**Minimum Length**: 2
**Maximum Length**: 43
**Mean**: 15
**Median**: 14.94
**Mode**: 14
**Population Variance**: 14.76



## 5 Experiments

Humor detection has been recognized as a binary text classification problem. But humor detection is a difficult task. Detecting incongruity in humor means that the model has to predict when a word is being used for multiple meanings (polysemy), or has to detect the change of focus in the sentence.

In order to classify the sentence based on the presence of incongruity, we try and detect the wordplay in the sentence by analyzing the relationships between the words. Word vector representations of(Mikolov et al., 2013a,b) the words were used to evaluate such relationships. In order to find the sentence-embedding from the words, two methods were used-

1. Unweighted averaging of the word vectors.
2. An RNN based language model (Cho et al., 2014).

In order to optimize the results, we performed experiments with both Word2Vec Skip Gram model (Mikolov et al., 2013b) and a count based *GloVe* Vector representation (Pennington et al., 2014) and the results have been shown below.

A minimum frequency of five has been used for creating these vectors. The window size for the embeddings was set to be 10.

### 5.1 Classification

Our sentence embedding is a Bag-of-Words (Zhang et al., 2010) vector averaging. We compare this with a neural language modeling based sentence embedding in the following sections.

#### 5.1.1 Method of Averaging

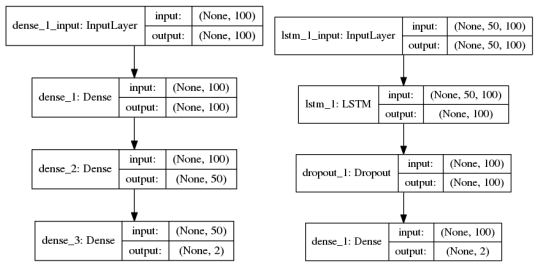
Let $\vec{x_i} \in \Re^k$ be a k-dimensional ($k$ = 100) vector corresponding to the $i$-th word in the sentence. Then the sentence of length $N$ can be represented as

$$\vec{X_i} = \frac{1}{N} \sum_{n=1}^{n=N} \vec{x_i} \quad (1)$$

The two classifiers used for the task are: Artificial Neural Networks (MLP) and Support Vector Machines (Tong and Koller, 2001). For the MLP classifier, as shown in Figure 1a, we used a 3 layered, densely connected network. The activation function used for the hidden layers was 'reLU' (Rectified Linear Unit) (Equation 2) (Dahl et al., 2013) (commonly known as reLU)

$$f(x) = max(x, 0) \quad (2)$$

and a softmax function at the last layer. In the SVM classifier, we used the *rbf* (Radial Basis Function) as the kernel (Hsu et al., 2003) and found the optimum parameters using an exhaustive grid search.




(a) MLP (b) LSTM

Figure 1: Network Architectures

#### 5.1.2 Neural Language Model

The Recurrent Neural Network (RNN) (Cho et al., 2014; Bengio et al., 2013) is a natural generalization of feedforward neural networks to sequences (Sutskever et al., 2014). Given a sequence of inputs $(x^1, ..., x^T)$, an RNN encoder encodes it into a single vector $c$ by iterating over the following equation:

$$h_t = f(x_t, h_{t-1})$$

We use an LSTM (Hochreiter and Schmidhuber, 1997) layer instead of RNN to capture long-term dependencies (Bengio et al., 1994).

The final encoded vector $c$ results from the equation $c = q(\{h_1, ...h_{T_x}\})$, (Cho et al., 2014) where $q(\{h_1, ..., h_T\}) = h_T$ for the case of LSTMs , as presented by (Sutskever et al., 2014). This encoded vector is used as the sentence embedding for classification purposes. The model was created using APIs by Tensorflow (Abadi et al., 2015) and has been presented in Figure 1b.

The sentences were pre-padded up to the length of 50 steps. We used stochastic gradient descent (SGD) algorithm together with Adadelta (Zeiler, 2012) to train the model. In order to tackle the difficulty of different lengths of sentences, we also used a variable-length vector (Dynamic Length RNN) (Cho et al., 2014) and found better results.

### 5.2 Results

| Technique | Reuters | BNC | Proverbs |
|---|---|---|---|
| MLP | 99.46% | 75.6% | 81.4% |
| SVM | 96.2% | 71.5% | 78.6% |
| LSTM | 74.20% | 52.2% | 55.6% |

Table 1: Comparison of results obtained using different datasets as negative samples.

The dataset was split into 80%-10%-10% (training - validation - testing) sets to train and test the model. Cross-validation was done through the dataset and the results were averaged. The final accuracy achieved was 99.46%.Empirically, it was observed that pre-trained vectors performed much better than the other freshly trained GloVe vectors and Word2Vec (Rehurek and Sojka, 2011) vectors. This can be attributed to the fact that they were trained on a much larger dataset, hence they better captured the substructures of the vector space.

| Technique | Accuracy |
|---|---|
| Our Approach | 95.8% |
| (Mihalcea et al., 2010) | **96.89%** |
| (Yang et al., 2015) | 80.5% |

Table 2: Comparison with other results from the literature on 16000 One-Liners Dataset.

As presented in table 1, LSTMs were not able to classify the sentences as accurately as MLP and SVM. We postulate that in order to perform well at humor classification, a much larger dataset is required for training the LSTM model. In Table 2, we report the efficacy of using different datasets as negative samples. Since News' headlines are semantically most dissimilar from humorous sentences, and the sentences from the BNC corpus are most similar, the accuracy is the highest in the former case, and the lowest in the latter.

We also tested our model on the 16000 One-Liners (Mihalcea and Strapparava, 2006) dataset, and got comparable results. It should be noted that since the humorous samples found in this dataset were extracted from a different source. This sort of cross-domain classification experiment proves that this approach can be generalized.

## 6 Conclusion and Future work

We have presented a methodology for detecting humor in social media text. A new dataset has been created and used to train machine learning models that can detect humor in English sentences. We are releasing this comprehensive dataset that has been standardized to be used in an NLP setting. We present our approach towards humor detection, along with the results achieved. Experimental results display the applicability of the model. Since LSTMs did not give acceptable results in classification, it would be interesting to use *'Attention'* (Bahdanau et al., 2014), and increase the dataset size to train such a model. In the future, we would also like to extend our work in computational humor to humor generation.

## References


Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Salvatore Attardo. 2010. *Linguistic theories of humor*, volume 1. Walter de Gruyter.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *ICWSM*, pages 574–577.

Francesco Barbieri and Horacio Saggion. 2014. Automatic detection of irony and humour in twitter. In *ICCC*, pages 155–162.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. 2013. Advances in optimizing recurrent networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8624–8628. IEEE.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Kim Binsted and Graeme Ritchie. 1997. Computational rules for generating punning riddles. *HUMOR-International Journal of Humor Research*, 10(1):25–76.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

BNC Consortium et al. 2012. The british national corpus, version 3 (bnc xml edition). 2007. *Distributed by Oxford University Computing Services on behalf of the BNC Consortium. URL: http://www. natcorp. ox. ac. uk (last accessed 25th May 2012)*.

George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

Jeffrey EF Friedl. 2002. *Mastering regular expressions*. " O'Reilly Media, Inc.".

Lisa Friedland and James Allan. 2008. Joke retrieval: recognizing the same joke told differently. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 883–892. ACM.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification.

Chloe Kiddon and Yuriy Brun. 2011. That's what she said: Double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 89–94. Association for Computational Linguistics.

Kevin Makice. 2009. *Twitter API: Up and running: Learn how to build applications with the Twitter API*. " O'Reilly Media, Inc.".

Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. *Computational Linguistics and Intelligent Text Processing*, pages 337–347.

Rada Mihalcea and Carlo Strapparava. 2005. Computational laughing: Automatic recognition of humorous one-liners. In *Proceedings of Cognitive Science Conference*, pages 1513–1518.

Rada Mihalcea and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2):126–142.

Rada Mihalcea, Carlo Strapparava, and Stephen Pulman. 2010. Computational models for incongruity detection in humour. *Computational linguistics and intelligent text processing*, pages 364–374.

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, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

John Morreall. 1986. The philosophy of laughter and humor.

Simon Munzert, Christian Rubba, Peter Meißner, and Dominic Nyhuis. 2014. *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons.

Luke de Oliveira and Alfredo Láinez Rodrigo. 2015. Humor detection in yelp reviews.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for f* r* i* e* n* d* s. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 208–215. Association for Computational Linguistics.

Victor Raskin. 2012. *Semantic mechanisms of humor*, volume 24. Springer Science & Business Media.

R Rehurek and P Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Oliviero Stock and Carlo Strapparava. 2003. Getting serious about the development of computational humor. In *IJCAI*, volume 3, pages 59–64.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Julia M Taylor and Lawrence J Mazlack. 2004a. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Julia M Taylor and Lawrence J Mazlack. 2004b. Humorous wordplay recognition. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3306–3311. IEEE.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard H Hovy. 2015. Humor recognition and humor anchor extraction. In *EMNLP*, pages 2367–2376.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52.