# Surely you're joking!
# Effect of multi-domain training on joke-recognition performance

**Ksenia Sokolova**
Computer Science Department
Princeton University
sokolova@princeton.edu

## Abstract

A lot of work has been done in the area of joke recognition. The problem is complicated by the intricacy of dataset formation: not only humor is subjective, but it also comes in different forms. Two main forms of data collection are either a costly production of small curated datasets, or scraping data from the themed web pages. The goal of this paper is to study the effect of using datasets from different joke domains and collection strategies on the performance, as well as explore the option of dataset selection using a quicker training option. Base model used is BERT-sentence encoding with a classifier. We demonstrate that training on high quality datasets from different domains results in a model that performs as well as training separate model on each dataset separately, and that freezing BERT to select the best data combination would not work.

## 1 Credits

This is a final project for COS 598C, "Deep Learning for Natural Language Processing", Spring'20, taught by Prof. Danqi Chen.

## 2 Introduction

> *Humor can be dissected as a frog can, but the thing dies in the process and the innards are discouraging to any but the pure scientific.* (White and White, 1941)

Humor is a serious business. Understating what exactly makes people laugh is an evasive task, shared across such fields as psychology, neuroscience and philosophy. Multiple theories exist on the scientific constructs of humor, and only certain commonalites are accepted among the most. One of them is "perception of incongruity", or juxtaposition of incompatible concepts (Sabato, 2019). Such definition by nature relies on the persons understanding of the world and previous experiences.

Joke is one form of humor expression. It is generally characterized as a short, amusing story ending in a punchline. Another form of humor expression is conversational humor, which can be further broadly subdivided into groups based on the linguistic properties (anecdote, wordplay, irony) or on the intention of using humor (satire, sarcasm, self-deprecation, etc) (Martin, 2007). These groups are not mutually exclusive.

All of the above makes automated humor understanding a very challenging task. Not only it requires outside knowledge, but also (1) is subjective to the person's opinion, making ground truth complicated to produce and (2) involves different overlapping types of humor, in effect including different data domains in one dataset. The former is problem is especially important in the light of the way the data is collected: it is either web-scraping or costly

mechanical-turk data collection.

## 3 Related Work

Significant effort has been applied in the area of computational humor recognition and generation. Some of the creative tasks include funny caption generation using CNN+LSTM (Yoshida et al., 2018) or computer-aided creation of funny Mad Lib stories (Hossain et al., 2017). In 2017 SemEval-2017 Task 6 was #HashtagWars, with the goal of comparative humor ranking of short tweets with a theme (Potash et al., 2017).

A more general problem, on the other hand, is identifying the joke. Significant amount of work has been done in that area, with recent shift towards using deep learning to tackle the problem.

(Taylor and Mazlack, 2004) used Raskin's *Semantic Theory of Verbal Humor* (Raskin, 1984) and statistical patterns of N-grams to create and recognize wordplay "knock-knock" jokes. (Chen and Soo, 2018) use CNN model with GloVe embeddings to identify jokes. (Weller and Seppi, 2019) used BERT encoder architecture and dataset scraped from Reddit to provide a measure for the joke humor. The Reddit dataset contains jokes formatted as "body" containing joke context and punchline, paired with a score - number of upvotes of the post. (Annamoradnejad, 2020) also uses BERT as an encoder, and create their own dataset entitled "ColBERT", with 200k equally split between short text jokes and neutral text to identify a joke. Note that BERT (Devlin et al., 2019) is a deeply bidirectional encoder, trained on masked language modeling and next sentence prediction tasks.

## 4 Data

There is a variety of datasets available online that are related to the task of humor classification, with a lot of papers (as noted in previous section) creating and publishing their own datasets.

Since the goal of this paper is exploring how different types of data affect training in the context of inherent multi-domain humor classification, we use different datasets and combinations of thereof.

For the purpose of the paper, we will fix definitions of the following:

**Pun**: form of word play exploiting multiple-meanings or similar sounding words (Merriam-Webster)

**One-liner joke**: a succinct joke delivered in a single line.

**Short joke**: a more general group of jokes, limited only by the length. Does not limit the type of humor contained.

Note, that the groups are like russian-dolls, could be contained within each other: pun can also be considered a one-liner joke, and all puns and one-liners fall into short joke category.

| Dataset | Positive | Negative |
|---|---|---|
| Humicroedit/FunLines | 4720 | 24601/13180 edited |
| Pun of the Day | 2423 | 2403 |
| Crowdtruth oneliners | 5250 | 11513 |
| Short Jokes | 231645 | 0 |
| Million News | 0 | 2M/used 200K |

Table 1: Number of positive and negative samples in each of the datsets used. Note that for Humicroedits, there are both original titles, and not-funny edits in the negative samples.

**Humicroedit and FunLines:** (Hossain et al., 2019, 2020). Both of the datasets have an original news title and "microedited" one, where one word was replaced to make is funny (example replacements and scores in Table 2). The readers then assigned a score from 0 to 3, with each title getting reviews from multiple people. The scores roughly correspond to: 0 - Not funny, 1 - slighlty funny, 2 - funny, 3 - very funny. Special effort was devoted to

ensure high quality labels, with consistent scoring and good distribution of labels. Still, there was disagreement over slightly-funny and funny label (Figure 1). In the future in the task of binary classification, samples with mean score below 1.5 will be considered neutral and score above - as positive example.

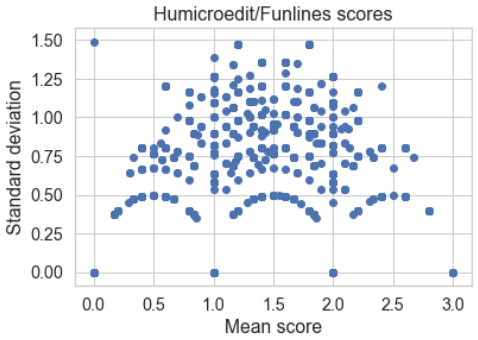


Figure 1: The most disagreement between labels happens over the 'medium' funny edits, in agreement with the subjectivity of humor. Specifically in this dataset, it seems that while "very funny" jokes were funny for everybody, the middle-ground ones were more debatable.

**Pun of the Day:** (Yang et al., 2015), made available by (Weller and Seppi, 2019). The dataset is evenly split between puns and neutral sentences.

**Crowdtruth oneliners:** (CrowdTruth, 2016) Dataset available on Github. It was scraped from Twitter, contains posts from "funny" accounts, as well as Reuters headlines, English proverbs and Wiki sentences. Basic cleaning was performed, where outliers by joke length we removed (which comprised one joke over 120 words).

**Short Jokes:** (Moudgil, 2017) Dataset contains 231657 jokes, in range of length from 10 to 200 characters, and was scraped primarily from Reddit. There are no negative (neutral) examples. Upon manual inspection, this dataset is generally noisy, with not funny jokes (per authors' judgement), and also contains dirty, highly offensive jokes. Based on the distribution of lengths of jokes, in total 12 jokes were 50 words and were removed from the dataset.

**Balancing dataset:** While some of the datasets listed are balanced between jokes and neutral, Short Jokes is not. This creates a severely unbalanced data. To mitigate this effect, we use *A Million News Headlines* (Rohk, 2020), and use the most recent 200K news titles, which roughly corresponds to the time period between October 2015 and December 2019.

The summary of the sizes of the datasets and number of positive/negative labels is available in Table 1. Distribution of length of jokes from each dataset is shown in Figure 2.

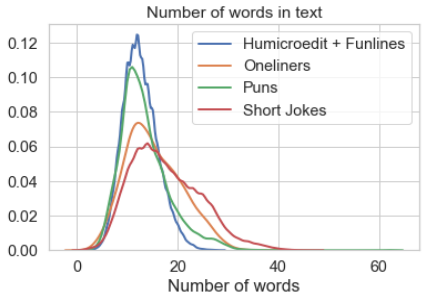


Figure 2: Distribution of the number of words in each joke by dataset. As evident from the plot, puns and short jokes are in general longer compared to Humicroedit/Funlines and puns.

## 5 Experiments

### 5.1 Models

Pre-trained *BERT-base-uncased* was obtained from (Wolf et al., 2019), and training was done using PyTorch. All of the models have the same basic structure, where BERT is used as an encoder and CLS head is fed into 2 fully-connected layers with 50 hidden neurons and 0.3 dropout in the middle. Adam optimizer was

| Original | Replacement | Scores | Mean score |
|---|---|---|---|
| Recent Scandals Highlight Trump 's Chaotic <Management/>Style | Fashion | 33333 | 3 |
| Chile election ends era of female presidents in Latin <America/> | Dance | 21110 | 1 |
| Merkel reassures EU over lack of Berlin coalition <deal/> | Catering | 0 | 0 |

Table 2: Examples of "very funny", "medium funny" and neutral Humicroedit/Funline edits

used, with a step scheduler. Separate models were ran, when BERT layers were frozen and when the model was fully trainable.

While training, model was evaluated on the validation dataset after every epoch. If performance on the validation improved, the model is saved. If not, the training continues and the best model from previous runs is not overwritten. Such method prevents using the model that is over-fitting on the training data.

### 5.2 Hyperparameter selection

For unfrozen BERT, hyperparameters used were: batch size 32, maximum length of input 30, learning rate $5 * 10^{-5}$, no step changes (batch size and learning rate taken from (google research, 2016)). The models were ran for 5 epochs, although the best iteration was the 1st or 2nd epoch in all the cases, aftern which the model started to over-train severely. The only exception was "Short Jokes" separate dataset, which did not converge for the given parameters, and instead was ran with learning rate $5 * 10^{-6}$ for 7 epochs.

For frozen BERT, an exhaustive grid search was used to select the best hyperparameters. In total 50 models were run to ensure that the model is stable under different conditions. Starting learning rate had the most influence on the convergence properties among the training parameters checked (batch size, step size and maximum length of the input). Learning rate of $0.001$ was the best option, and the other parameters: batch size $128$, maximum length $30$, and step size of learning rate decay by $0.1$ was $5$ epochs.

Maximum length for both of the sets was fixed at 30 words for both types of models based on the distribution of length of the jokes (Figure 2) and the general goal of prediction for the short jokes.

### 5.3 Datasets

Datasets from Table 1 were used individually (short jokes merged with Million News to produce a balanced dataset) and combined together to create 8 datasets in total. Description on the combined datasets, as well as resulting sizes are available in Table 3. Additionally, all "expanded" datasets were trained multiple times with different weight (1, 0.7, 0.5) assigned to the loss from short jokes partition during backpropagation to alleviate the potential low quality of the samples.

## 6 Results

### 6.1 Test hold-out performance

Full results are available in Table 4. AUC score was chosen as a primary comparison measure as it does not depend on the threshold for a binary classification task. When fixing the training dataset, training with unfrozen BERT produced better results, although it is worth to mention that it took longer to train.

Short jokes and Oneliner performance was very similar, and most of the times good performance on one meant good performance on the other. Moreover, these datasets had high AUC scores (above 0.95) for all models that used either of the training sets for both frozen and unfrozen BERT. For example, training just on Puns dataset produced poor results on Short jokes and Oneliners. Similarly, training just

| Dataset name | Description | Pos labels | Neg labels |
|---|---|---|---|
| Basic all | Humicroedit/Funlines, Puns, Onliners | 10839 | 35752 |
| Basic | Puns, Oneliners | 6119 | 11151 |
| Expanded all | Humicroedit/Funlines, Puns, Onliners, Short | 196029 | 195878 |
| Expanded 50 | Humicroedit/Funlines, Puns, Onliners, 50K Short | 37542 | 59049 |
| Expanded 100 | Humicroedit/Funlines, Puns, Onliners, 100K Short | 64553 | 82038 |

Table 3: Different combinations of the datasets used and resulting number of labels

on Humicroedit/Funlines gave 0.82 AUC for both of Short Jokes and Oneliner datasets. But if either Short Jokes or Onelines was in the training data, performance was very good.

Puns testing dataset had equally good score when training on Puns separately, and when mixing it with the other 3 main datasets (Basic all) when working with unfrozen BERT. For frozen BERT this was not true, as the gap between models trained on Puns vs Basic all dataset was much larger.

Humicroedit/Funlines dataset had the worst results overall among all options, with no model that would breach 0.7 AUC score. The best result was obtained when training on the Basic all dataset with unfrozen BERT, and not when training on just humicroedit/funlines, which is different from the other datasets.

Adding weight to the loss function did have an effect on the performance, although quantifying the exact results would be hard, as it depended on other parameters. Specifically, when using "Expanded all" dataset, adding a 0.7 weight led to a slight improvement in both frozen and unfrozen BERT. This was not true for either Expanded 50 or Expanded 100. For Expanded 100 adding 0.7 weight only improved unfrozen BERT, and for Expanded 50 using data without weights was the best.

### 6.2 Misclassified examples

Based on factors listed above, it seems that the model that performed best on all the tasks combined would be "Basic all". It had score very close to the best for Puns, strong performance on Short Jokes and Oneliners, and the best score on Humicroedit, which proved to be the hardest dataset to achieve good results on.

Consider some of the misclassified examples from Puns dataset (original spelling maintained):

> *The hairless goat wished that it had mohair.* Prediction: 0.28. True: 1

This is an example of a homophonic and homographic joke, as mohair is a yarn made from the fur of Angora goat, and also sounds like "more hair". Note that "mohair" is not in the BERT-base vocabulary.

> *Is the training given to expectant parents apparenticeship.* Prediction: 0.3. True: 1

Here wrong spelling coupled with homophonic features is what makes the pun.

Next, consider some misclassified Humicroedit/FunLines samples:

> *Sen. Rand Paul assaulted at his Kentucky home, squirrel arrested.* Prediction: 0.02. True: 1

> *Delaware state trooper dances after being shot in convenience store parking lot.* Prediction: 0.24. True: 1

> *Trump Org told to frame phony Time magazine issues.* Prediction: 0.04. True: 1

> *Thanks to Trump, recovery from baldness finally starting.* Prediction: 0.66. True: 0

| **Unfrozen BERT** | | | | |
|---|---|---|---|---|
| Training dataset | Test dataset | | | |
| | Puns | Short Jokes | Oneliners | Humicroedit/Funlines |
| Humicroedit/Funlines | 0.65 | 0.82 | 0.82 | 0.66 |
| Puns | **0.96** | 0.36 | 0.54 | 0.49 |
| Short Jokes | 0.55 | 0.999 | 0.74 | 0.54 |
| Oneliners | 0.53 | 0.99 | 0.99 | 0.54 |
| Basic all | 0.95 | 0.98 | 0.99 | **0.67** |
| Basic | **0.96** | 0.96 | 0.99 | 0.51 |
| Expanded all | 0.84 | 0.999 | 0.995 | 0.63 |
| Expanded all, 0.7 weight | 0.85 | 0.999 | 0.99 | 0.63 |
| Expanded all, 0.5 weight | 0.8 | 0.999 | 0.996 | 0.62 |
| Expanded 50 | 0.92 | 0.99 | 0.99 | 0.66 |
| Expanded 50, 0.7 weight | 0.9 | 0.99 | 0.99 | 0.66 |
| Expanded 50, 0.5 weight | 0.9 | 0.99 | 0.99 | 0.59 |
| Expanded 100 | 0.88 | 0.99 | 0.99 | 0.62 |
| Expanded 100, 0.7 weight | 0.93 | 0.99 | 0.99 | 0.64 |
| Expanded 100, 0.5 weight | 0.91 | 0.99 | 0.99 | 0.62 |
| **Frozen BERT** | | | | |
| | Puns | Short Jokes | Oneliners | Humicroedit/Funlines |
| Humicroedit/Funlines | 0.62 | 0.81 | 0.78 | 0.58 |
| Puns | 0.92 | 0.27 | 0.46 | 0.48 |
| Short Jokes | 0.53 | 0.999 | 0.91 | 0.56 |
| Oneliners | 0.5 | 0.99 | 0.99 | 0.55 |
| Basic all | 0.83 | 0.97 | 0.98 | 0.59 |
| Basic | 0.84 | 0.97 | 0.98 | 0.53 |
| Expanded all | 0.68 | 0.999 | 0.99 | 0.58 |
| Expanded all, 0.7 weight | 0.71 | 0.999 | 0.99 | 0.58 |
| Expanded 50 | 0.78 | 0.99 | 0.99 | 0.59 |
| Expanded 50, 0.7 weight | 0.76 | 0.99 | 0.99 | 0.58 |
| Expanded 50, 0.5 weight | 0.77 | 0.99 | 0.99 | 0.59 |

Table 4: AUC scores on the holdout test set for different training subsets. In bold are the best results for Puns test dataset and Humicroedit/Funlines test dataset. All of the testing datasets contain 20% of the full dataset that was obtained online. For Expanded datasets weight realtes to the weight assigned to the loss during backpropagation.

## 7 Discussion

For all the same datasets, unfreezing BERT always leads to better results, even in the absence of the hyperparameter fine-tuning. The trends between frozen and unfrozen BERT models are not consistent, and therefore selecting the best data subset using unfrozen BERT and then training on the full BERT would not be the best option.

Training on the dataset from which the testing data was taken almost always leads to better results, but the model has a worse generalization qualities when comparing with other datasets. This is to be expected, as the domain should not change between training and testing. However, mixing all the good quality training datasets together gives a model that performs better on Humicroedit/Funlines data and the same on the Puns dataset. While the improvement is small, this is an interesting fact. One possible explanation could be that Humicroedit/Funlines dataset is a mix of puns and jokes (inter-domain), or that by itself the dataset is too small.

Adding weight to the poor samples was beneficial in some cases but not the others. When using a whole Short Jokes large dataset, adding lower weights improved performance slightly compared to using no weights. This could be due to the vast majority of examples that are low quality, adding weight adds more value to the high quality samples from other sources. But since the model performance is better without this large dataset at all, it seems that the noisy samples still overpower the model.

Performance on Short jokes and Oneliners is connected probably because they were scraped from the dame website and majority of the jokes are in the same style, although Oneliners is better filtered. Thus, good performance on one dataset would be a precursor to good performance on the other. Additionally, Short Jokes could be ”easy” to get a good performance on, due to the semantic differences between jokes and news title content that was used as neutral, which is another reason for testing on different types of datasets. This is on comparison to Humicroedit/Funline jokes that are very similar to news headlines in style.

While it is hard to prove without additional annotations, it is authors' observation that homophonic puns and puns based on wrong spelling are harder to classify correctly, and represent the majority of misclassified examples in the Puns dataset. This is also not surprising in theory, as to classify the former model needs to know pronunciation and with the latter - map the misspelled words close to the intended ones.

Examples from Humicroedit/Funlines dataset highlight the problems that arises with this dataset - without context some of the jokes loose their punch. And while the humor is subjective, the authors of the paper would not call some misclassified examples from this dataset (shown in the Results section) particularly funny.

## 8 Future Work

There are limitations to this study. First, only one type of the architecture was selected. Second, the hyperparameters for frozen BERT were selected through grid search, and then fixed the same, and no hyperparamter fine-tuning happened for unfrozen BERT. While the authors do not expect this to strongly affect the comparative results, it is something that could be explored. For example, an improvement that would affect the overall performance would be to use BERT-Large instead of BERT-Base, and fine-tune the length of the sentence used.

Another interesting approach would be to consider the difference in learned features between models that are trained on mixed datasets vs the models trained on limited data.

In other words, whether there is difference in the importance of specific features that arises from different training domains, and how does it change with addition of new data.

Additionally, it would be interesting to have a testing dataset with different types of humor labeled separately, to have a more comprehensive diagnostics of the performance.

## 9 Conclusions

The goal of this paper is to analyze how including samples from different joke-domains into training affects the performance, and whether assigning weight to the low-quality samples could alleviate the problem of scraped datasets. While adding weight to poor quality samples could be useful, having cleaner dataset gives better results. Increasing the domain of training data results in better performance for some tasks. Selecting best combination of datasets based on frozen BERT with intent to be trained with unfrozen BERT would not work.

## References


Issa Annamoradnejad. 2020. Colbert: Using bert sentence embedding for humor detection. *ArXiv*, abs/2004.12765.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana. Association for Computational Linguistics.

CrowdTruth. 2016. Short text corpus with focus on humor detection. https://github.com/CrowdTruth/Short-Text-Corpus-For-Humor-Detection.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "president vows to cut hair": Dataset and analysis of creative text editing for humorous headlines. *ArXiv*, abs/1906.00274.

Nabil Hossain, John Krumm, Tanvir Sajed, and Henry Kautz. 2020. Stimulating creativity with funlines: A case study of humor generation in headlines. *ArXiv*, abs/2002.02031.

Nabil Hossain, John Krumm, Lucy Vanderwende, Eric Horvitz, and Henry Kautz. 2017. Filling the blanks (hint: plural noun) for mad Libs humor. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Copenhagen, Denmark. Association for Computational Linguistics.

Rod A. Martin. 2007. Chapter 1 - introduction to the psychology of humor. In Rod A. Martin, editor, *The Psychology of Humor*, pages 1 – 30. Academic Press, Burlington.

Merriam-Webster. Pun.

Abhinav Moudgil. 2017. Short jokes. https://www.kaggle.com/abhinavmoudgil95/short-jokes.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. SemEval-2017 task 6: #HashtagWars: Learning a sense of humor. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 49–57, Vancouver, Canada. Association for Computational Linguistics.

Victor Raskin. 1984. *Semantic Theory of Humor*, pages 99–147. Springer Netherlands, Dordrecht.

google research. 2016. Bert. https://github.com/google-research/bert.

Rohk. 2020. A million news headlines. https://www.kaggle.com/therohk/million-headlines.

Giovanni Sabato. 2019. What's so funny? the science of why we laugh.

Julia M. Taylor and Lawrence J. Mazlack. 2004. Computationally recognizing wordplay in jokes. In *In Proceedings of CogSci 2004*.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3621–3625, Hong Kong, China. Association for Computational Linguistics.

E.B. White and K.S.A. White. 1941. *A Subtreasury of American Humor*. Coward-McCann, Incorporated.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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

Kota Yoshida, Munetaka Minoguchi, Kenichiro Wani, Akio Nakamura, and Hirokatsu Kataoka. 2018. Neural joking machine: Humorous image captioning. *arXiv preprint arXiv:1805.11850*.