SemEval2020 Task 7:
the computational humour of single-word edits

Gavin Leech

May 2020

1 Task

We are to estimate the funniness of news headlines edited by humans in an attempt to make them funny. We use the *Humicroedit* dataset [1]. The intervention is a "micro-edit", single-word replacements of *noun-noun* or *verb-verb*. The edited sentence was labelled with the mean of 5 humans assigning a funniness score $\in [0, 1, 2, 3]$.

Original Headline	Substitute	Grade
Kushner to visit **Mexico** following latest Trump tirades	**therapist**	2.8
Hilllary Clinton Staffers Considered Campaign Slogan `Because It's Her **Turn**'	**fault**	2.8
The Latest: BBC cuts **ties** with Myanmar TV station	**pies**	1.8
Oklahoma isn't **working**. Can anyone fix this failing American state?	**okay**	0.0
4 **soldiers** killed in Nagorno-Karabakh fighting: Officials	**rabbits**	0.0

Figure 1: Example rows in Humicroedit [2]

Our task is to predict the mean funniness y from the original - microedit pairs X: $y \sim X$. (This was 'Subtask 1' in the original competition.) [3]

As usual in NLP, the raw text is hopelessly high-dimensional. So we need a representation which compresses the humour-relevant parts of it.

Consider the abstract requirements of solving such a task. There are various accounts of humour: as making the audience feel superior, as the surprise of incongruity, or as testing social boundaries [4]. These all seem incomplete, or perhaps representing different types of humour. But we can imagine what a humour representation would need for each:

1. Superiority: needs to capture some notion of relative status or of vice. Could well be audience-specific.

2. Incongruity: needs to have a good word embedding and then to detect large semantic distances resulting from the edit.

3. Play: needs to capture norms and norm violations.

(The learned representation need have no high-level feature corresponding to these social abstractions, and there may well be more to humour than this handful of philosophical views can cover, but it seems fair to say that a good representation would simulate something like them.)

1.1 Hypotheses

Punchline: late word index as funniness predictor

Many jokes take the form of a setup and a punchline, where the humour is located later in the joke. (This fits the Incongruity account.) Does the lateness of the word edit predict the humour score?

$$y \propto i/n$$

for $i = 1, ...n$ where each headline is a word sequence $h = w_1, w_2, ..., w_n$.

(However, the correlation between edit lateness and mean funniness is low: Pearson's $r = 0.09$, $p < 0.001$. Figure 4 shows how this cashes out over a positive but narrow band of funniness.)

A General factor of funniness?

We can use the atomic changes in each microedit to try and find edits which are generally funny, or the centroids of different funny clusters. (e.g. Slapstick, toilet humour, the sacred, sex, death.)

A basic test of this is unigram representation: does the mere presence of the word increase funniness? There were 7,004 unique tokens in the data set.

If humour is about contrasts (incongruity), we should expect the unigrams to be worse overall than bigrams: however, if we use both unigrams and bigrams, then use feature selection to reduce redundancy, then perhaps some of the unigrams are themselves helpful.

2 Data: the Humicroedit dataset

The original *Humicroedit* data is all from January 2017 to May 2018. Insofar as the dataset relies on being topical for its humour (for instance, knowledge of campaign slogans from the 2016 US election), this could limit the generalisation of our learned humour representation. Similarly, the labels are all by Americans.

2.1 Schema

1. id: unique int
2. original: headline as on Reddit with word to replace denoted < word_to_be_replaced/ >
3. edit: word chosen to replace tagged word
4. grades: list of 5 funniness labels from 5 judges; 0 (Not Funny); 1 (Slightly Funny); 2 (Moderately Funny); 3 (Funny).
5. meanGrade: average grade

The edits are by Amazon Mechanical Turk workers; other AMT workers supplied the funniness labels. Both the edit index and the edit word was chosen by them.

A better measure, controlling for the ambient humour of the strange fragment of English used in headlines, would be to score both the original headline and the edited headline and then use the difference as a label.

2.2 Exploration

See Figures 2-4.

Humour variance Humour differs greatly even within a culture. The mean funniness variance within judgments of a given headline is 0.62, one-fifth of the scale. There was extreme disagreement (at least one '0' score and one '3' score) in 24% of headlines.

There are 2,283 instances of headlines that someone rates as both funny (3) and another person rated as not funny (0). This could prove predicting headlines to be tough.

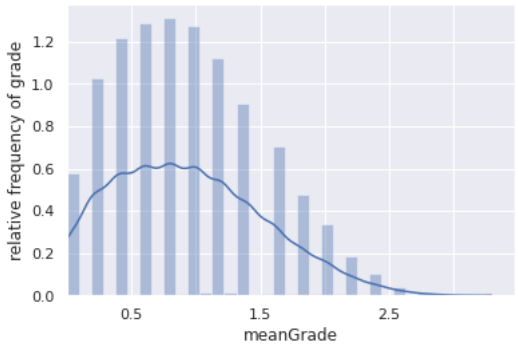

Figure 2: Distribution of edited headline funniness

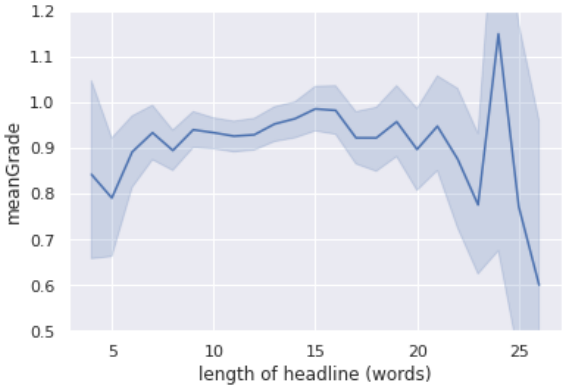

Figure 3: Funniness by headline length

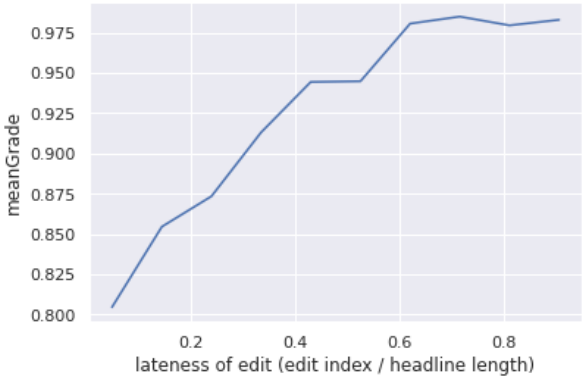

Figure 4: Funniness by lateness of edit

3 Methods

My code (end-to-end including download, preprocessing, and evaluation) can be found here with helpers here.

3.1 Preprocessing

1. Create full edited headline
2. Remove 'stop words' using the standard NLTK corpus.
3. Also remove numbers
4. Remove rare words, less than 2 occurrences (min_df)
5. Vectorise both headlines
   (a) Do bigram bag of words (BOW)
   (b) Do TDF-IF vectors.
   (c) Do word2vec embeddings and take the average for the headline embedding.

Bigrams yielded 27,020 distinct pairs for features.

3.2 Modelling

I tried four regressors for each representation: stochastic gradient descent regression, support vector regression, random forest, and gradient boosted machines. These represent a good coverage of linear and nonlinear methods. During development I worked with a random 30% sample.

4 Results

Baseline

In the official competition, the best recorded submission had RMSE 0.513 [2]. But a better baseline for our purposes, a naive constant approach predicting the mean funniness for all inputs, gave RMSE = 0.578.

Representation	Best model	RMSE
None	Baseline (mean)	0.578
Unigram BOW	GBM	0.563
Bigram BOW	GBM	0.563
Bigram TF-IDF	GBM	0.567
Google News Sentence2vec	SGD	0.566
Punchline index only	GBM	0.573

Table 1: Test-set performance with each representation

5 Postmortem

Hyperparameter search and cross-validation over massive bags-of-words and TF-IDFs was the limiting step. The randomised search was very easy to develop, though. It does however introduce a lot of instability into my results.

The unigrams were slightly better than the naive baseline, which is weak evidence for the general factor of humour idea: the mere presence of individual words is enough to somewhat predict the funniness (or unfunniness) of a sentence.

Automatic feature selection is likely to explain the good performance of GBM tree ensembles in the high-dimensional representations.

A regression using only the index of the edited word performed marginally better than the baseline, very weak evidence for the punchline hypothesis.

The unigrams and bigrams performed very similarly, which was unexpected. The word embedding representation - the only one making use of actual semantics - gave no improvement over presence and frequency representations, which was extremely surprising. This is possibly down the naive way in which the headlines were embedded (as the mean of its word embeddings).

Open questions after the short exercise above:

1. The dataset isn't huge: only 12,000 examples. There actually are additional data available, from the FunLines online game [5][6]. I didn't have time to make use of it.
2. The strength of Humicroedit is the atomic, easy credit assignment. But humour is a large phenomenon. A more natural signal could be obtained from contrasting a serious new outlet (e.g. The New York Times) with one of the many dedicated joke news outlets (e.g. The Onion).
3. We removed numerals along with the stop words. Are numbers funny?
4. It would have been cleaner to organise all this as a (sklearn) Pipeline.

5. The task is probably well suited to complex neural networks, but we used simpler regressors. This actually backfired, since sklearn has no GPU acceleration and the tens of thousands of features slowed things down considerably.

6. Semantic clustering would be a much better way of testing the general factor idea.

7. The hyperparameter search was pretty primitive; no Bayesian optimisation for instance, just randomised. I had no compute to speak of. I manually picked the search ranges for hyperparameters based on past experience and could easily have excluded good solutions. The random cross-validation may have also missed things.

8. Humour has some universal and some culture-specific components. How much less predictive is the present model when applied on non-US data?

References

[1] Nabil Hossain, John Krumm, and Michael Gamon. "President vows to cut <taxes> hair": Dataset and analysis of creative text editing for humorous headlines. *arXiv preprint arXiv:1906.00274*, 2019.

[2] Nabil Hossain. Assessing the funniness of edited news headlines (SemEval-2020). `https://competitions.codalab.org/competitions/20970#results`.

[3] Nabil Hossain, John Krumm, Lucy Vanderwende, Eric Horvitz, and Henry Kautz. 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, 2017.

[4] John Morreall. Philosophy of humor. *Stanford Encyclopedia of Philosophy*, 2012.

[5] Tanvir Sajed, John Krumm, Henry Kautz, and Nabil Hossain. Funlines. `https://funlines.co/humor/`, 2019.

[6] Nabil Hossain, John Krumm, Tanvir Sajed, and Henry Kautz. Stimulating creativity with funlines: A case study of humor generation in headlines. *arXiv preprint arXiv:2002.02031*, 2020.