LT3 at SemEval-2020 Task 7: Comparing Feature-Based and Transformer-Based Approaches to Detect Funny Headlines

Bram Vanroy, Sofie Labat, Olha Kaminska, Els Lefever and Véronique Hoste
LT3, Language and Translation Technology Team
Department of Translation, Interpreting and Communication – Ghent University
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
firstname.lastname@ugent.be

Abstract

This paper presents two different systems for the SemEval shared task 7 on Assessing Humor in Edited News Headlines, sub-task 1, where the aim was to estimate the intensity of humor generated in edited headlines. Our first system is a feature-based machine learning system that combines different types of information (e.g. word embeddings, string similarity, part-of-speech tags, perplexity scores, named entity recognition) in a Nu Support Vector Regressor (NuSVR). The second system is a deep learning-based approach that uses the pre-trained language model RoBERTa to learn latent features in the news headlines that are useful to predict the funniness of each headline. The latter system was also our final submission to the competition and is ranked seventh among the 49 participating teams, with a root-mean-square error (RMSE) of 0.5253.

1 Introduction

Research in the field of computational humor can be divided into three distinct tasks: detecting, assessing and generating humor. Humor detection is performed by either distinguishing humorous items such as one-liners, jokes or texts from their non-humorous counterparts (Chen and Soo, 2018; Fan et al., 2020), or by ranking these items according to their respective funniness (Potash et al., 2017; Hossain et al., 2019). Moreover, researchers have tried to gain insights in the techniques and mechanisms that underlie humor production (Yang et al., 2015; Cattle and Ma, 2018; Hossain et al., 2019; West and Horvitz, 2019). Such insights can be used for automatic humor generation (Shahaf et al., 2015; He et al., 2019; Winters et al., 2019). Due to the subjective nature of humor and the deep conceptual common-sense knowledge that it requires, tasks involving humor remain challenging AI problems. Part of the challenge also lies in the scarcity of publicly available datasets.

For the SemEval shared task on Assessing the Funninness of Edited News Headlines (Hossain et al., 2020), Hossain et al. (2019) introduce a novel dataset, named "Humicroedit", that can be used for all three tasks described above (see Section 2). The shared task is split into two sub-tasks, namely (i) a regression task to predict the mean grade of all annotators for a given edited sentence, and (ii) a binary classification task to select the funnier of two edited headlines. We only participated in the regression sub-task.

The remainder of this paper is structured as follows. In Section 2, we provide an overview of the dataset that was used for the competition. Section 3 introduces our two machine learning systems, namely (i) a feature-based regression model (NuSVR), and (ii) a state-of-the-art deep learning system. Next, in Section 4, our results for the two systems are compared and discussed. Finally, Section 5 concludes this paper.

2 Data Analysis

While most existing datasets in the field of computational humor are constructed for binary classification purposes (Miller et al., 2017; Khodak et al., 2018) (funny or not), the organizers of the competition provided a novel resource ("Humicroedit") that contains humor intensity labels for edited news headlines. The dataset contains around 15,000 headlines (Hossain et al., 2019), divided into train (9,653 headlines;

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1033–1040
Barcelona, Spain (Online), December 12, 2020.

64%), development (2,420; 16%) and test (3,025; 20%) data. The original headlines were collected from Reddit, and experts from Amazon Mechanical Turk were asked to create and to estimate the funniness of minor edits to the headlines. The suggested changes were applied to one word or collocation while respecting the following rule: entities or nouns are replaced by nouns only, verbs are replaced with verbs. The scores from annotators ranged from 0 (not funny at all) to 3 (very funny). All instances were rated by 5, 10, or 15 annotators. For each instance, all provided grades and their average were given, the average being the value that models must learn to predict correctly. The distribution of the average grades is visualised in Figure 1, which shows that the dataset contains less highly ranked "funny" samples than lowly ranked "not funny" instances.

We examined the headlines and their grades to find any patterns that could prove useful. The original tokens before replacement are mainly single words (94.9%), rarely two (4.9%) and sporadically three words (0.2%). The replacement tokens, on the other hand, are always a single word. We attempted to identify common topics in the original headlines by using Latent Dirichlet Allocation (LDA) and keyword extraction based on term frequency-inverse document frequency (tf-idf) scores. The first approach did not provide reasonable results as it was hard to separate the particular independent topics because most of them contain a lot of common terms. With the second approach, we found that the keywords in the headlines are mainly Named Entities (tags 'Person', 'Location' and 'Organization'). For this reason, we extracted only nouns with those NER tags for all the funniest (with a mean grade higher or equal to 2.0) and the least funny headlines (with a mean grade less or equal to 1.0) and used them to calculate tf-idf scores. These scores were in turn used to select keywords. Surprisingly, we did not discover any particular patterns or specific topics for those groups because they mostly had common keywords. We also took into account topics proposed by Hossain et al. (2019) for edited words, which the authors ranked according to how they influence the funniness of a headline. They also proposed a set of keywords for each headline. We compared the proposed keywords of the funniest topics with those obtained from the original headlines, but did not find any useful overlap.

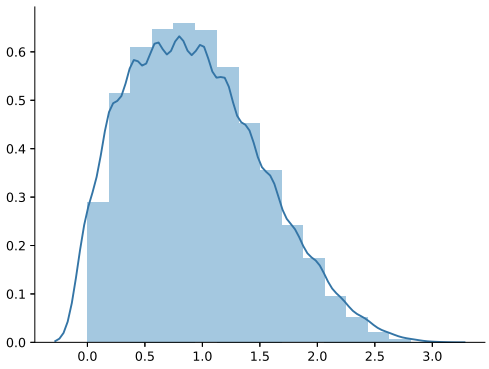

Figure 1: Density plot of the mean grades.

3 System Descriptions

In this section we will start by introducing the features that we engineered for a NuSVR system (Section 3.1), followed by a description of our best performing system, which is a fine-tuned transformer-based language model (Section 3.2).

3.1 Feature-Based Machine-Learning System

For our first machine learning system, we extracted various features that either capture information about the original headline or about the relationship between the original and edited headline. By creating features that model different types of humor generation techniques used by the editors, we want to learn how successful each technique is. In this section, we describe the different types of features that were created and motivate why these features would be useful for the task of humor detection according to existing humor theories. Secondly, we discuss the experimental setup and the machine learning model that was used to perform regression.

3.1.1 Feature engineering

Some of our features were inspired by Chhaya et al. (2018), who introduced a supervised learning method to model frustration. Their feature set covers a wide range of features for emotion detection, which intuitively could be useful for our task as well. We implemented three groups of their features, leaving out

those features that are either not relevant for this task or underrepresented in the dataset. The following features were used for the original headline: lexical features (number of uppercase words, number of non-alphanumeric characters, number of punctuation marks, average word length), NER-based features (presence of NER tags 'person', 'location', 'organization'), and so-called *derived* features quantifying the readability of headlines (the number of contractions, presence of hedges, subjectivity and sentiment scores that were generated with the `TextBlob` Python library (Loria et al., 2014)).

The other features are based on initial findings of Hossain et al. (2019), who performed a qualitative analysis of their Humicroedit dataset to gain insights in the types of humor generation techniques used by the editors. Our first group of features deals with the length of jokes. Ritchie (2004) claims that jokes often contain additional information to make them funnier. Hossain et al. (2019) confirmed that in general longer headlines have more humor generation potential. The authors also found that headlines that contain a higher number and proportion of replaceable words (i.e. nouns or verbs) are usually funnier. Therefore, we decided to create the following features for the original headline: absolute length (in words), relative length (in words), the absolute number of replaceable part-of-speech (POS) tags, and the proportion of replaceable POS tags.

Hossain et al. (2019) also found evidence that the funniest headlines had their replacement word often positioned towards the end of the headline. This finding corresponds to the "setup and punchline" approach (Rochmawati, 2017): while a joke or headline builds up towards a certain expectation, the ending is suddenly changed to something unexpected that is still coherent, thus producing a humorous effect. In light of these findings, we also added the position of the edited word in the sentence as a feature.

The remaining features focus on modeling information about the relation between the original and the edited headline. Hossain et al. (2019) found that editors sometimes chose replacements that are similar in pronunciation to the replaced words. Hence, we implemented a range of string similarity metrics between the lower-cased replacement and original words. We generated features with the following metrics: Dice, Dice with 3-grams, Jaccard distance, XDice, XXDice, Longest Common Subsequence Ratio (LCSR), Normalized Levenshtein Distance (NLD) and Jaro-Winkler Similarity. As for the semantic features, Hossain et al. (2019) noticed that the editors frequently opted for replacements that are semantically distant from their replaced counterparts. To capture this humor generation technique, we calculated the cosine similarity between the vector of the original token and its replacement, and between the original headline and its replacement. For this purpose, after removing numbers and punctuation marks and lower-casing, word vectors were extracted with fastText (Bojanowski et al., 2017). The vector for a given headline was calculated as the mean of its word vectors. Formula 1 illustrates how we modified cosine similarity to calculate the distance between two vectors A and B as a feature with a positive value to match the other features:

$$cosine\_similarity(A, B) = \frac{1 + \frac{A \cdot B}{||A|| \times ||B||}}{2} \quad (1)$$

In our modification of cosine similarity, two vectors (word or headline representations) can be exactly the opposite (0), exactly the same (1), or any value in-between.

Our final group of features deals with perplexity. Intuitively, an utterance might be considered funny when there is a high *surprisal* factor involved. This corresponds to the incongruity theory of humor (Morreall, 2016): jokes that violate an expectation generate a surprising and humorous effect. In terms of running text, this means that a given word with high perplexity (thus being an unexpected word) could be conceived as humorous. The intuitive importance of perplexity makes language models perfect candidates for this task. Hence, we used the SRILM toolkit (Stolcke, 2002) to build a statistical n-gram language model on the News on the Web (NOW) corpus (Davies, 2017) (lower-cased), which contains a large collection of textual data from web-based newspapers and magazines. During training, we applied the following parameters: `order` $= 5$, `kndiscount` and `interpolate`. Besides building one language model with these parameters, we trained an additional language model for which n-gram probabilities are pruned (with a threshold of $1E{-}8$). The two resulting models were used to calculate perplexity scores for the lower-cased original and edited headlines. Features were then obtained by dividing the scores of the

original headlines by the scores of the edited headlines for the two respective language models.

3.1.2 Machine-learning system

We combined the different groups of features in different machine learning systems to get a sense of which model could obtain the lowest RMSE for this regression problem. We used systems available in `sklearn` (Pedregosa et al., 2011), particularly `KNeighborsRegressor`, `RandomForestRegressor`, `SVR` (Support Vector Regression) and `NuSVR`. After some rudimental hyperparameter tuning, we found that NuSVR outperformed the other machine learners so we further optimized its hyperparameters by performing 5-fold cross-validation grid search on 5 fixed subsamples of the train set. This resulted in the following, best-performing hyperparameters: `kernel` = `RBF`, `nu` = 0.25, `C` = 10, `gamma` = `auto`. We also investigated the performance of individual feature groups (see Table 1), but to our surprise, there was no feature group clearly outperforming the others. Hence, we decided to simply combine all feature groups in the system. The results of the final system will be discussed in Section 4.

Feature group	List of features	RMSE
Lexical	# Upper-case words, # non-alphanum. char., # punctuation marks, average word length	0.576
NER-based	NER tags PERSON, LOCATION, ORGANIZATION	0.579
Readability	# Contractions, presence of hedges, subjectivity scores, sentiment scores	0.578
Length	Absolute length, relative length, absolute # replaceable POS tags, proportion of replaceable POS tags	0.578
Positional	Position of the replaced word in the sentence	0.577
Orthographic	Dice, Dice 3-grams, Jaccard distance, XDice, XXDice, LCSR, NLD, Jaro-Winkler Similarity	0.575
Semantic	Cosine sim. original token and its replacement, cosine sim. original headline and its replacement	0.575
Perplexity	Standard perplexity feature, pruned perplexity feature	0.576

Table 1: An overview of the different groups of features and their performance in the NuSVR algorithm (without hyperparameter tuning) when trained on train + dev data, and evaluated on test data.

3.2 Transformer-Based System

Humor often expects world knowledge and awareness of the context to drive its intent home. Language models are capable of some language understanding (Wang et al., 2018, GLUE, and its successors), and on the popular SQUAD 2.0 question-answering benchmark (Rajpurkar et al., 2018) single (non-ensemble) language models even outperform human participants (Yang et al., 2019; Lan et al., 2019; Clark et al., 2020). The question is, however, if language models can use this information to figure out whether some given input is not only (un)true but also whether it is funny.

Even though language models are nothing new, the last few years have seen great improvements in their architecture, most notably the attention mechanism and, building on that, the transformer (Vaswani et al., 2017). Language models are typically trained with unsupervised training objectives which allows them to learn general language patterns from a given dataset. These patterns are encoded in the millions or even billions of parameters of the language model. Transfer learning allows us to *transfer* these parameters to other down-stream tasks, which is exactly what we did for this task.

In our experiments we used RoBERTa (Liu et al., 2019), which is an improvement of the most well-known currently used language model, BERT (Devlin et al., 2019). RoBERTa was trained on a dataset that is much larger than BERT. Whereas BERT was originally trained on 16 GB of data, RoBERTa utilises 160 GB of corpora. Most notable for our purposes is that a large part of that dataset (76 GB) is their newly created CC-news corpus, containing 63 million English news articles. Even though the current task involves news *headlines*, we would assume that during pretraining, RoBERTa at least learnt some hidden features that could relate to news-specific text.

In our best performing model, we finetuned RoBERTa-base (125M parameters) using a custom head configuration (Appendix A). On top of the base RoBERTa model we added a pre-classifier ($in\_features = 6144$, $out\_features = 2048$) and a classifier layer ($in\_features = 2048$, $out\_features = 1$). We concatenate the last four layers of the base model (inspired by the original BERT paper (Devlin et al., 2019, Table 7) and then concatenate the output of the classification token $\langle s \rangle$ and the output of the edited token ($768 * 4 * 2 = 6144$). We also tried concatenating the output of $\langle s \rangle$, the edited token and the original

token, or any combination of those, but none were as successful as combining $\langle s \rangle$ and the edited token. Other configurations for the top layers were tested but were found to be less effective. For instance, we tried adding an activation function between the linear layers (ReLU, GELU), tried different intermediate dimensions, and we used dropout to prevent overfitting. However, the previously discussed settings performed best. We also experimented with finetuning the language model on the NOW corpus (Davies, 2017) prior to finetuning it on this task but found no improved results, perhaps because the pretraining data already contained a lot of news text. Finally, we also tried out the large RoBERTa model (355M parameters) and other architectures (ALBERT (Lan et al., 2019), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019)), but none of these lead to improved performance. Different sets of hyperparameters were tested as well, but the best configuration consists of AdamW as an optimizer (Loshchilov and Hutter, 2017) with lr $5E{-}5$, a maximal sequence length of 64, batch size of 64 and early stopping if the validation did not improve for five epochs. The dataset was not preprocessed for the transformer-based system.

4 Experimental Results

In this section we present the results of our two systems and compare them to each other and to a baseline that predicts the average `meanGrade` value of the whole training set (0.9356) for all instances in the test set. The RoBERTa system performs better than the feature-based system in terms of loss (RMSE 0.5253 and 0.5720 respectively). In addition, the Pearson correlations between the two systems and the correct labels vary significantly (Table 2): whereas the transformer-based system correlates moderately with the gold labels ($r = .41$), the correlation of the feature-based system is poor ($r = .11$). The difference in performance between the baseline and the feature-based system, on the other hand, is very small (0.5747 and 0.5720 respectively).

	MSE	**RMSE**	**Pearson r***
roberta	0.2760	0.5253	.4127
feature-based	0.3272	0.5720	.1135
baseline	0.3302	0.5747	NA

Table 2: Results of the two systems and a baseline. $^*p < .01$

	entropy*	**stdev***
roberta	$-.1300$	$-.1241$
feature-based	$-.2822$	$-.2645$
baseline	$-.2758$	$-.2636$

Table 3: Correlations between squared error on the one hand and entropy and standard deviation on the other. $^*p < .01$

We hypothesised that our systems would perform better on instances where annotators agreed more on the funniness score than on the instances where they disagreed. To verify this, we calculated the correlations between the squared error and the annotation entropy and between the squared error and the standard deviation of the ratings. Entropy, in this context, can be seen as the amount of uncertainty among or agreement between the annotators to give the same score to a specific instance.[1]

The correlations given in Table 3 show unexpected results. We anticipated positive correlations between the loss of the systems (squared error) and the entropy and standard deviation of the grades. That would indicate that low inter-annotator agreement adds a layer of noise to the predictability of a grade. However, the correlations show that the opposite is true: the larger the disagreement or variance between annotators, the smaller the squared loss, which means better performance. Even though unexpected, the correlation is not strong.

5 Conclusion

In this research, we propose two machine learning systems to tackle the regression task of predicting the funniness of edited news headlines. While our feature-based machine learning system obtains a RMSE of 0.5720, it is outperformed by our deep learning system which achieves a RMSE of 0.5253. In addition, the Pearson correlation between the predicted scores and the given mean scores is better for the latter

[1]For simplicity's sake, we consider the given grades (0, 1, 2 or 3) as nominal variables even though they are in fact ordinal.

system. For future work, it would be interesting to test the performance of our best-performing pipeline on other humor detection datasets in order to validate its portability and robustness.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Andrew Cattle and Xiaojuan Ma. 2018. Recognizing Humour using Word Associations and Humour Anchor Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1849–1858.

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.

Niyati Chhaya, Kushal Chawla, Tanya Goyal, Projjal Chanda, and Jaya Singh. 2018. Frustrated, Polite, or Formal: Quantifying Feelings and Tone in Email. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 76–86.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Mark Davies. 2017. The new 4.3 billion word NOW corpus, with 4–5 million words of data added every day. In *Proceedings of the 9th International Corpus Linguistics Conference*, pages 523–524.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Yanbo Zou. 2020. Humor detection via an internal and external neural network. *Neurocomputing*.

He He, Nanyun Peng, and Percy Liang. 2019. Pun Generation with Surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1734–1744.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "president vows to cut <taxes> hair": Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 133–142, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A Large Self-Annotated Corpus for Sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 641–646.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv:1907.11692 [cs]*, July.

Steven Loria, P Keen, M Honnibal, R Yankovsky, D Karesh, E Dempsey, et al. 2014. Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*, 3.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 Task 7: Detection and Interpretation of English Puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68.

John Morreall. 2016. Philosophy of humor. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Mueller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July. Association for Computational Linguistics.

Graeme Ritchie. 2004. *The Linguistic Analysis of Jokes*. Routledge.

Dyah Rochmawati. 2017. Pragmatic and rhetorical strategies in the English-written jokes. *Indonesian Journal of Applied Linguistics*, 7(1):149–159.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside Jokes: Identifying Humorous Cartoon Captions. In *KDD'15 : Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1065–1074.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS 2017*, pages 1–15, Long Beach, CA, USA.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Robert West and Eric Horvitz. 2019. "Reverse-Engineering Satire, or "Paper on Computational Humor Accepted Despite Making Serious Advances". In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pages 7265–7272.

Thomas Winters, Vincent Nys, and Danny De Schreye. 2019. Towards a General Framework for Humor Generation from Rated Examples. In *Proceedings of the 10th International Conference on Computational Creativity*, pages 274–281.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

A Transformer-based system visualization