

# MLEngineer at SemEval-2020 Task 7: BERT-Flair based Humor Detection Model (BFHumor)

**Farah Shatnawi**  
ffshatnawi16@cit.just.edu.jo

**Malak Abdullah**  
mabdullah@just.edu.jo

**Mahmoud Hammad**  
m-hammad@just.edu.jo

Department of Computer Science  
Jordan University of Science and Technology  
Irbid, Jordan

## Abstract

Task 7, Assessing the Funniness of Edited News Headlines, in the International Workshop SemEval2020 introduces two sub-tasks to predict the funniness values of edited news headlines from the Reddit website. This paper proposes the BFHumor model of the MLEngineer team that participates in both sub-tasks in this competition. The BFHumor's model is defined as a BERT-Flair based humor detection model that is a combination of different pre-trained models with various Natural Language Processing (NLP) techniques. The Bidirectional Encoder Representations from Transformers (BERT) regressor is considered the primary pre-trained model in our approach, whereas Flair is the main NLP library. It is worth mentioning that the BFHumor model has been ranked 4<sup>th</sup> in sub-task1 with a root mean square error (RMSE) value of 0.51966, and it is 0.02 away from the first ranked model. Also, the team is ranked 12<sup>th</sup> in the sub-task2 with an accuracy of 0.62291, which is 0.05 away from the top-ranked model. Our results indicate that the BFHumor model is one of the top models for detecting humor in the text.

## 1 Introduction

Humor is a distinctive quality of humans, which is considered a figurative language; where the person can express his feelings through social media once entertainment media have existed or if he faces a funny event. Researchers found that humor leads to enhance the human's health and mood (Hernández-Farías et al., 2015; Mao and Liu, 2019; Reyes et al., 2012; Yan and Pedersen, 2017).

In the last decades, the world has witnessed a rapid move in Artificial Intelligent (AI) field toward simulating human intelligence in machines or computer systems (Wen et al., 2019; Raja and S, 2019). One of the main branches of AI is Natural Language Processing (NLP), where the computer learns how to understand human languages regardless the diversity and complexity of human languages that are considered open challenges for NLP systems and communities (Abdullah and Shaikh, 2018; Zhou et al., 2020). Knowing that humor has many facets and can be produced through words (texts), gestures (vision), and prosodic cues (acoustic) (Hasan et al., 2019; Abdullah et al., 2018), humor detection is considered a multimodal language. The reasons behind the difficulty of detecting humor are due to (1) the nature of the context, emotion, and rhythm, which contains a large degree of complexity for the machine to understand the level of the humor. (2) the cultural differences, knowing that humor is universal, the different cultures see the humor in various ways (Martin et al., 1993).

Detecting humor from social media sites or TV dialogues using a set of algorithms had been studied thoroughly in the NLP field (Glazkova et al., 2019; Bertero and Fung, 2016). In this paper, we detect the humor through news headlines from the Reddit website<sup>1</sup> that are provided by SemEval-2020 organizers (Hossain et al., 2020). The International Workshop on Semantic Evaluation SemEval-2020 has introduced several NLP shared tasks. Task 7, Assessing the Funniness of Edited News Headlines, aims to motivate participants to build systems to predict the level of funniness in edited news headlines. The task asked participants to replace a word in the headline using a micro-edit and check if this modification makes

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

<sup>1</sup>Reddit website: <https://www.reddit.com>

the news headlines more funnier or not. The task consists of two subtasks (sub-task1 and sub-task2). In sub-task1, the participants should predict the funniness of the edited headline based on RMSE value. While sub-task2 is intended to predict which edited headline is more funnier based on the accuracy value. Further clarifications about task 7 and the dataset in section 3. Our team, MLEngineer, proposed a BFHumor model for predicting the funniness value of the edited news headlines in both subtasks. Our model is an ensembling model of different state-of-the-art pre-trained models with various NLP techniques. The MLEngineer has been ranked 4<sup>th</sup> in sub-task1 and 12<sup>th</sup> in sub-task2 out of 84 teams.

The remaining sections of this paper are organized as follows: Section 2 describes the existed work on detecting humor. Section 3 provides insights into the proposed system methodology and architecture. Section 4 presents the key experimental setup and BFHumor results. Finally, in Section 5, we conclude with the findings of our research.

## 2 Related work

A plethora of research has studied the figurative languages in various datasets to predict the funniness value. The publication race in this area is growing at a fast pace between the industry and academic researchers. This section presents some analysis of the existed work related to humor, irony, ridicule, satire detection, and assessing funniness.

The authors in (Mao and Liu, 2019; Ortega-Bueno et al., 2018; Farzin et al., 2019; Castro et al., 2018; Weller and Seppi, 2019; Garain, 2019) studied humor detection in various languages to classify each tweet into a joke or not and how they are funny. In (Mao and Liu, 2019), the authors classified the dataset into a joke or not with a dataset that was obtained from Twitter in the Spanish language. While Bueno et al. (Ortega-Bueno et al., 2018), predicted the level of funniness in tweets (score value prediction based on the average value of 5-stars).

Several machine learning techniques have been used to predict the funniness value. In (Garain, 2019), the researchers proposed a model using deep learning algorithms in the HAHA-2019 task, such as Bidirectional-LSTM (BiLSTM) and LSTM. The researchers in (Blinov et al., 2019) applied several classification algorithms for humor recognition from texts, such as the linear Support Vector Machine (SVM). Another group of researchers (Glazkova et al., 2019) built a neural network classifier and applied it to the HAHA at IberLEF 2019 dataset. Also, authors in (Farzin et al., 2019) used Universal Language Model Fine-tuning for Text Classification ULMFiT and Naive Bayes with SVM (NBSVM).

In this paper, we have built a robust model, named BFHumor, dealing with continuous labels, which aims to predict the funniness value of the news headline between zero and three. This model is a combination of the various pre-trained model and techniques that are mainly BERT regressor and the NLP Flair library. Up to our knowledge, we are the first to use these state-of-the-art techniques for humor detection.

## 3 System Methodology

In this section, we present the workflow of our proposed system to predict the funniness of edited news headlines as shown in Figure 1. More details about the workflow are in the following subsections.

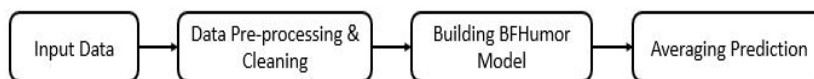


Figure 1: Work-flow of the BFHumor model.

### 3.1 Input Data

The dataset used in this paper is obtained from Task 7 of SemEval2020. The organizers of this task (Hossain et al., 2019; Hossain et al., 2020) collected the original news headlines datasets from a popular website, Reddit. This data is called Humicroedit that was reduced to 15,095 headlines for sub-task1 and 14,696 headlines for sub-task2. The dataset is divided into training, development, and testing dataset for

two sub-tasks, as shown in Table 1. Some examples of original and edited news headlines appear in Table 2.

Sub-task	Train	Dev	Test	Total
Sub-task-1	9,652	2,419	3,024	15,095
Sub-task-2	9,381	2,355	2,960	14,696

Table 1: Dataset size for the two sub-tasks of Task 7.

Tasks	Funny or Not	Score	Examples
Original Sub-Task1	Not Funny	0	Dutch minister resigns in drug baron <row/>
	Slightly Funny	1	Trump ’s 2nd Nominee for <Army/>Secretary Withdraws
Original Sub-Task2	Moderately Funny	2	#WomensMarch against <Donald Trump/>around the world
	Funny	2.8	Conservative Leaders Urge Mitch McConnell to <Resign/>
Edited Sub-Task1	Not Funny	0	Dutch minister resigns in drug baron blow
	Slightly Funny	1	Trump ’s 2nd Nominee for Class Secretary Withdraws
Edited Sub-Task2	Moderately Funny	2	#WomensMarch against kitchens around the world
	Funny	2.8	Conservative Leaders Urge Mitch McConnell to Bathe

Table 2: Examples of original & edited news headlines

### 3.2 Data Pre-processing and Cleaning

The preprocessing and cleaning procedure consists of two critical steps: (1) Word-Replacement and (2) Data Pre-processing. In the first step, we replaced the substitutional word with the word between </>. Such as replacing the **Vice** word with **school** word in (“Trump was told weeks ago that Flynn misled <Vice/>President”) that is became (“Trump was told weeks ago that Flynn misled school President”).

In the second step, we have experimented with several preprocessing packages, however, the best results of the model came with using the original data with a few amount of preprocessing. We have applied existing preprocessing packages for cleaning the data, such as ekphrasis<sup>2</sup>, spaCy<sup>3</sup>, and clean-text.<sup>4</sup> We have tokenized the headline by splitting it into chunks of words. Then, we have applied spell correction on those words, unpacked the contractions (can’t to can not), and unpacked the hashtags by applying word segmentation (#MeToo to “<hashtag>Me Too </hashtag>”). Also, we have converted the text into a lower case and deleted stop words using spaCy package. Finally, we have removed numbers, currency symbols (i.e.\$,£), punctuation marks (i.e.,?!:;()[]#@), and applied lower case in the clean-text package.

### 3.3 Building BFHumor Model

In this subsection, we describe our proposed model, the BERT-Flair-based Humor Detection Model (BFHumor). BFHumor model is built using two main components: the BERT regressor and the Flair library, as shown in Figure 2.

**Sub-model1:** we have applied a novel BERT regressor models from the bert-sklearn library with four types: (bert-base-cased, bert-base-uncased, bert-large-cased, and bert-base-cased-finetuned-mrpc)<sup>5</sup> (Wolf et al., 2019). The input of Sub-model1 is the training dataset to predict the corresponding mean grade (label) of that testing dataset. We have trained our regressor with a training batch size of 16 and an evaluation batch of 8 for 2 epochs. The learning rate is 3e-5, the validation fraction is 0.0; gradient accumulation steps are 1, and the maximum length is 64.

**Sub-model2:** we have used the Flair NLP library, which (1) applies several NLP models to text (i.e., named entity recognition) (2) supports multi-language for text (i.e., German, and French) (3) provides an interface that extracts word embeddings (Akbik et al., 2019). After that, we have used a RoBERTa type<sup>6</sup>

<sup>2</sup>ekphrasis: <https://github.com/cbaziotis/ekphrasis>

<sup>3</sup>spaCy: <https://stackoverflow.com/questions/45605946/how-to-do-text-pre-processing-using-spacy>

<sup>4</sup>clean-text: <https://github.com/jfilter/clean-text>

<sup>5</sup>Pre-trained models: [https://huggingface.co/transformers/pretrained\\_models.html](https://huggingface.co/transformers/pretrained_models.html)

<sup>6</sup>RoBERTa: <https://github.com/pytorch/fairseq/tree/master/examples/roberta>

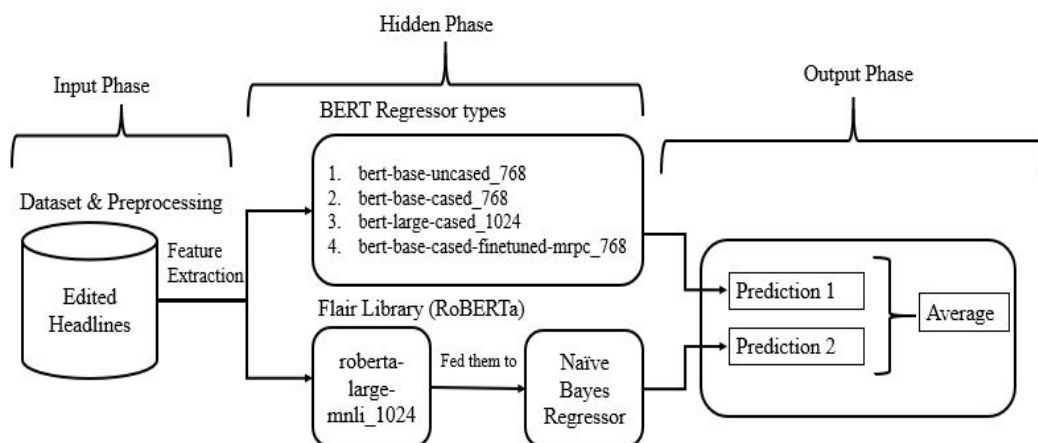


Figure 2: The architecture of BFHumor model.

as the underlying Flair to extract word embeddings from the training and testing dataset, especially from roberta-large-mnli. Then feeds them to the machine learning algorithms. Sub-model2 details: 24 layers, 1024 hidden layers, 16 heads, and 355 million parameters.

Finally, to predict the funniness values of the testing data in the two-sub-tasks, the Naïve Bayes regressor (Mayo and Frank, 2020) reads these embeddings as an input and returns the predictions. Then, we ensembled the predictions of BERT and Naïve Bayes based on different weights based on several experiments for the best results in both sub-tasks.

## 4 Experimental Setup & BFHumor Results

### 4.1 Experimental Setup

#### 4.1.1 Sub-task1

Several experiments were applied for sub-task1 with using: BERT regressor with two types bert-base-cased and bert-base-uncased, XLNet pre-trained model(Yang et al., 2019), Recurrent neural network (RNN) and NB (Mayo and Frank, 2020; De Mulder et al., 2015). More details about models information are shown in Table 3.

Firstly, we applied the bert-base-cased, bert-base-uncased as underlying Bert type (Devlin et al., 2018), and XLNet pre-trained model (Yang et al., 2019; Wolf et al., 2019) with different hyper-parameters from Anaconda software<sup>7</sup> using Google Colab.<sup>8</sup> We used the model sizes in BERT and XLNet models as following: number of layers= 12, hidden size= 768, number of self-attention heads= 12 and total parameters = 110 million. The RMSE results are as follows: 0.55974, 0.62023, and 0.57896 respectively.

Secondly, the architecture of the used RNN contains input, hidden, and output layers(De Mulder et al., 2015). The input layer consists of word2vec as a word embedding with size = 300. Then, two Long Term Short Memory layers ( 256 units, recurrent dropout is equal to 0.2, dropout is equal to 0.1, and the return\_sequences is equal to true and false in two layers). Also, two dense layers (256 units with activation function is ReLU), and two dropout layers (dropout is equal to 0.3). The output layer is Dense with one unit and the sigmoid activation function. The loss function is mean squared error; optimizer function is adam, metric used is mean squared error, the batch size is 52, and the number of the epoch is 200. Based on the above parameters, the RMSE value obtained is 0.57855.

We extracted 153 features using AffectiveTweets package (Bravo-Marquez et al., 2019) in Weka tool<sup>9</sup> plus Python codes. The Weka features are extracted by using Embedding, Input Lexicon, and Sentistrength functions. While Python features are Jaccard, cosine, and complexity. Then, we fed them into NB, which gave 0.57223 as an RMSE value.

<sup>7</sup>Anaconda site: <https://www.anaconda.com/>

<sup>8</sup>Google Colab site: <https://colab.research.google.com/notebooks/intro.ipynb>

<sup>9</sup>Weka tool: <https://www.tutorialspoint.com/weka/index.htm>

Model	Model Info		RMSE	
bert-base-cased	Parameters	Value	0.55974	
	Validation fraction	0		
	Learning rate	3e-5		
	Gradient accumulation	1		
	Max sequence length	64		
	Train & Eval batch size	16 , 8		
bert-base-uncased	Validation fraction	0	0.62023	
	Learning rate	1e-5		
	Gradient accumulation	1		
	Max sequence length	200		
	Train & Eval batch size	52 , 26		
xlnet-base-cased	Learning rate	2e-5	0.57896	
	Gradient accumulation	3		
	Max sequence length	128		
	Train batch size	8		
	Adam epsilon	1e-8		
	Number of epochs	1		
	Max grad norm	1		
	Warmup steps	0		
RNN	Input layer	Hidden layer	Output layer	0.57855
	Embedding layer	Masking layer	Dense layer	
		LSTM		
		Dense layer		
	Dropout layer			
NB	Weka Features	Lexical & Semantic		0.57223
	Embedding	Jaccard	Cosine	
	Input Lexicon			
	Lexicon			
	Sentistrength	Complexity		

Table 3: Model information and results of sub-task1

#### 4.1.2 Sub-task2

We also conducted several experiments for sub-task2, such as Bert regressor with bert-large-cased, ELMO (Peters et al., 2018), and roberta-large-mnli as a RoBERTa type from Flair library.

Firstly, we applied the bert-large-cased as underlying BERT type (Devlin et al., 2018) in Anaconda software. We used the model sizes in this type as following: number of layers= 24, the hidden size= 1024, the number of self-attention heads= 16, and total parameters = 340 million. The accuracy result is 0.55741.

Secondly, we used the roberta-large-mnli and ELMO models. Then, we fed the word embeddings of these models into an NB algorithm to obtain the accuracy values as 0.58980, and 0.54931, respectively. The hyper-parameters, number of dimensions, algorithms, accuracy values for the three above models are shown in Table 4.

## 4.2 BFHumor Results

In both sub-tasks, we use the same general structure with only one difference, which is the output phase (see Figure 2). Each sub-task passes the edited headlines through two different models, which produces two predictions and merges them.

We have noticed the following findings when we conducted several experiments:

1. Using two epochs in any of the BERT regressor model types gives better prediction results than using more than two epochs. But rarely, one epoch was giving the best prediction results.
2. Splitting the data into 0.8 and 0.2 for train and test data, respectively using the Hold-out method in NB regressor gives the best results.
3. Using the roberta-large-mnli word embedding with NB gives the best results.

Model	Model Info	Accuracy
BERT	Parameter	value
	Validation fraction	0
	Learning rate	3e-5
	Gradient accumulation	1
	Max sequence length	64
	Train batch size	16
	Eval batch size	8
	Number of epochs	2
ELMO	Number of dimension	Algorithm
	1024	NB
roberta-large-mnli	1024	NB
		0.55741
		0.54931
		0.58980

Table 4: Model information and results of sub-task2.

- Feeding all training datasets after using a clean-text package to the bert-based-uncased type as an input gives the best results.

MLEngineer team has participated in task 7 of SemEval-2020 using the BFHumor model has achieved the 4<sup>th</sup> rank out of 49 participants in sub-task1 based on the RMSE metric, and the 12<sup>th</sup> rank out of 32 participants in sub-task2 based on the accuracy metric as shown in Table 5.

Sub-tasks	Metric	Result
Sub-task-1	RMSE	0.51966
Sub-task-2	Accuracy	0.62291

Table 5: Sub-task-1 result.

## 5 Conclusion

In this paper, we have participated in Task 7 of SemEval 2020 as the MLEngineer team. We presented our novel model, BFHumor, BERT-Flair based Humor Detection Model to predict the funniest values of edited headlines in both sub-tasks. BFHumor is a unique model because it combined the BERT regressor and Flair library and used the same underlying architecture in both sub-tasks.

In the BFHumor model, we selected the bert-base-cased, bert-base-uncased, bert-large-cased, and bert-base-cased-finetuned-mrpc as the underlying BERT regressor models. Also, we chose the roberta-large-mnli as an underlying Flair type. Then, we merged the prediction results of two sub-models.

The performance of the BFHumor model outperformed the baseline system in the competition task, indicating that it is a promising model for detecting humor in the text. based on RMSE value and the accuracy: we were among the top 4 teams in the sub-task1 with an RMSE value of 0.51966, which is 0.02 away from the first ranked model. Also, we got the 12<sup>th</sup> in the sub-task2 with an accuracy of 0.62291, which is 0.05 away from the first ranked model.

## References

- Malak Abdullah and Samira Shaikh. 2018. Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 350–357.
- Malak Abdullah, Mirsad Hadzikadicy, and Samira Shaikhz. 2018. Sedat: sentiment and emotion detection in arabic text using cnn-lstm deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 835–840. IEEE.

- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.
- Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 130–135.
- Vladislav Blinov, Valeria Bolotova-Baranova, and Pavel Braslavski. 2019. Large dataset and language model fun-tuning for humor recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4027–4032.
- Felipe Bravo-Marquez, Eibe Frank, Bernhard Pfahringer, and Saif M Mohammad. 2019. Affectivetweets: a weka package for analyzing affect in tweets. *Journal of Machine Learning Research*, pages 1–6.
- Santiago Castro, Luis Chiruzzo, and Aiala Rosá. 2018. Overview of the haha task: Humor analysis based on human annotation at ibereval 2018. In *IberEval@ SEPLN*, pages 187–194.
- Wim De Mulder, Steven Bethard, and Marie-Francine Moens. 2015. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, 30(1):61–98.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Bobak Farzin, Piotr Czapla, and Jeremy Howard. 2019. Applying a pre-trained language model to spanish twitter humor prediction. *arXiv preprint arXiv:1907.03187*.
- Avishek Garain. 2019. Humor analysis based on human annotation (haha)-2019: Humor analysis at tweet level using deep learning. In *IberLEF@ SEPLN*, pages 191–196.
- Anna Glazkova, Nadezhda Ganzherli, and Elena Mikhalkova. 2019. Utmn at haha@ iberlef2019: Recognizing humor in spanish tweets using hard parameter sharing for neural networks. In *IberLEF@ SEPLN*, pages 222–228.
- Md Kamrul Hasan, Wasifur Rahman, Amir Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed Hoque. 2019. Ur-funny: A multimodal language dataset for understanding humor. *arXiv preprint arXiv:1904.06618*.
- Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. 2015. Applying basic features from sentiment analysis for automatic irony detection. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 337–344. Springer.
- Nabil Hossain, John Krumm, and Michael Gamon. 2019. ” president vows to cut<taxes>hair”: Dataset and analysis of creative text editing for humorous headlines. *arXiv preprint arXiv:1906.00274*.
- 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.
- Jihang Mao and Wanli Liu. 2019. A bert-based approach for automatic humor detection and scoring. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*. CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (9 2019), pages 197–202.
- Rod A Martin, Nicholas A Kuiper, L Joan Olinger, and Kathryn A Dance. 1993. Humor, coping with stress, self-concept, and psychological well-being. *Humor*, 6:89–89.
- Michael Mayo and Eibe Frank. 2020. Improving naive bayes for regression with optimized artificial surrogate data. *Applied Artificial Intelligence*, 34(6):484–514.
- Reynier Ortega-Bueno, Carlos E Muniz-Cuza, José E Medina Pagola, and Paolo Rosso. 2018. Uo upv: Deep linguistic humor detection in spanish social media. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018)*, pages 204–213.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

- Beschi Raja and Sneha S. 2019. A conceptual overview and systematic review on artificial intelligence and its approaches. *International Journal of Emerging Technology and Innovative Engineering*, 5(12):821–828.
- Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.
- Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. *arXiv preprint arXiv:1909.00252*.
- Andrew Wen, Sunyang Fu, Sungrim Moon, Mohamed El Wazir, Andrew Rosenbaum, Vinod C Kaggal, Sijia Liu, Sunghwan Sohn, Hongfang Liu, and Jungwei Fan. 2019. Desiderata for delivering nlp to accelerate healthcare ai advancement and a mayo clinic nlp-as-a-service implementation. *npj Digital Medicine*, 2(1):1–7.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.
- Xinru Yan and Ted Pedersen. 2017. Who’s to say what’s funny? a computer using language models and deep learning, that’s who! *arXiv preprint arXiv:1705.10272*.
- 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 5753–5763.
- Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. Progress in neural nlp: modeling, learning, and reasoning. *Engineering*, 6(3):275–290.