Assessing Humor in Edited News Headlines: A Knowledge Graph Based Approach

Zhe Niu
HKUST
zniu@cse.ust.hk

Yuen-hoi Lau
HKUST
yhlauai@cse.ust.hk

Hua Kang
HKUST
hkangae@cse.ust.hk

Abstract

Automatic humor recognition is a literally interesting but challenging topic. It is related to the NLP applications like sentiment analysis, public opinions analysis and dialogue systems. In this work, we propose to use the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) architecture together with the automatic knowledge graphs generation model Commonsense Transformers (COMET) (Bosselut et al., 2019) to predict the funniness level of a micro-edit on the news headline. We conduct thorough experiments using the Humicroedit dataset (Hossain et al., 2019). Our results show that the proposed method reduces the RMSE on the development set by around 5% comparing with the official baseline result.

1 Introduction

Recently, automatic humor recognition has attracted widespread attention from natural language processing (NLP) researchers (Hossain et al., 2019; Khodak et al., 2017; Cattle and Ma, 2018). Humor recognition is a literally interesting topic, and is related to the NLP applications like sentiment analysis, public opinions analysis and dialogue systems. To automatically recognize humor, machines should be able to tell how funny a sentence is by assigning it a score, or recognize the funnier sentence over two sentences. Automatically humor recognition is a challenging task as machines not only need to know the statistics of the funniness over words but also have to understand the sentence and the scene behind.

Previous works use non-deep learning based method like logistic regression classifier (Khodak et al., 2017), random forest (Cattle and Ma, 2018), or simple deep neural networks architecture like CNN (Chen and Soo, 2018) or Bidirectional LSTM (Hossain et al., 2019) to classify whether a sentence is funny or not, which may not be powerful enough to obtain a deeper sense of humor. Moreover, many of the works like (Hossain et al., 2019; Khodak et al., 2017; Cattle and Ma, 2018) trained classifier or regressor over sentences. Since these models take only one sentence as input at one time, it is difficult for them to learn the common sense knowledge underlying between sentences.

In this work, we propose to use the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) architecture together with the automatic knowledge graphs generation model Commonsense Transformers (COMET) (Bosselut et al., 2019) to predict the funniness level of a microedit on the news headline. The BERT framework enables us to pretrain our model using external large scale corpus. Even if without pretraining, the BERT loss still provides additional supervision and enables more detailed contextual feature extraction. As humor is situated in a broader context requiring considerable external knowledge, incorporating commonsense knowledge graph enables the model to have a extended understanding of one specific headline. By using the COMET network pretrained on ATOMIC (Sap et al., 2019) or ConceptNet (Liu and Singh, 2004), we are allowed to generate the unseen phrases and address the unseen input event, which solves the problem of incompleteness in existing knowledge graphs.

We evaluate our method using the Humicroedit (Hossain et al., 2019) task-1 dataset, which consists official training set and official development set. The official test set has not been published at the time of this writing. Root Mean Squared Error (RMSE) will be used as the metrics to measure the differences between the human-labeled grade and predicted grade.

The major contributions of our work can be summarized as below:

1. We propose to use the BERT framework on the Humicroedit challenge, we evaluate the effectiveness of pretraining our framework on an additional dataset Examiner (Deceased, 2018).
2. We propose to use commonsense network (COMET) pretrained on ATOMIC and ConceptNet to augment the input sequence of the network.
3. We provides two choices for the loss of the grade prediction, the mean squared error (MSE) loss and the soft cross entropy (SCE) loss.
4. Our methods reduce the RMSE by 5% comparing to the official baseline result.

The rest of this paper is organized as follows: First introduce the related works in Section 2. Then, we introduce our methodologies in Section 3 and the experiment results in Section 4. Finally, we conclude in Section 5.

2 Related Works

Classifying sentences into funny ones or not has attracted the attention of NLP researchers. Recent works including logistic regression classifier (Khodak et al., 2017), random forest (Cattle and Ma, 2018), convolutional neural networks(CNN) (Chen and Soo, 2018) and bidirectional LSTM (Hossain et al., 2019) illustrate the application of simple model structures in this field. However, they may not be able to capture deep meanings of a sentence given limited data sets and simple model structures. Logistic regression classifier simply puts word embeddings into the logistic regression model and outputs a score to classify whether it is funny or not. Random forest is an ensemble learning method for classification and choose the mode of classes as the decision given there are a set of statistical machine learning methods. They used word association strength features to train a random forest classifier using scikit-learn with 100 estimators. While CNN can learn a fixed set of local neighborhood features which are phrases depending on the shape and the number of the CNN filters, it may not understand the sentence as a whole. Bidirectional LSTM can capture the meanings from both directions of a sentence only in a sequential manner. However, it cannot relate a specific word to other parts of a sentence, which limits its ability to understand a sentence. Given that a word in a sentence is edited, the degree of humor is changed. All the existing works cannot understand whether a sentence with a word edited is more humorous or not in a deeper way. On the other hand, some methods such as various types of transformers can be used to extract features from the context and knowledge graphs can add more relevant information to a given sentence to help to better understand the sentence.

Contextual Feature Extraction Models Various recurrent neural network architectures have been used for contextual feature extraction in NLP. They take sentences of arbitrary length as input, operate sequentially on the input and retain hidden vectors as the memory to pass through RNN cells. However, due to vanishing gradients, it is difficult for RNNs to learn long-range dependencies in the sequence. Long short term memory (LSTM) models, which is a variant of RNN models, are developed to deal with vanishing gradient problems encountered by common RNN models. They are composed of LSTM cells that have three gates, an input gate, an output gate and a forget gate to remember information over arbitrary time intervals and control the input and output information of each cell. Transformer (Vaswani et al., 2017) is an architecture well-suited for NLP tasks such as machine translation and text summarization. Transformers consist of an ordered set of encoders and decoders. Each encoder takes in input vectors, uses a self-attention scheme to weigh the relevance of a set of input encodings from the previous encoder and a feed forward neural network to further process. Decoders have the similar fashion as encoders but have an additional attention mechanism across encodings of encoders. Compared to RNN, Transformers allow for more parallelization, enabling training on much more data than before and leading to the development of pretrained models including BERT, which is developed to pretrain bidirectional deep representations which based on left and right context. The pretrained BERT model (Devlin et al., 2018) can be fine-tuned with just one additional output layer to obtain the state-of-the-art results on a wide range of tasks such as question answering and classification.

Knowledge Graph Understanding the semantics requires understanding of objects and their relations. There have been considerable knowledge graphs to formalize them such as ConceptNet (Speer et al., 2017), ATOMIC (Sap et al., 2019) and ASER (Zhang et al., 2019). However, for different corpus, these knowledge graphs may be restricted by their small size and cannot have good performance under unbounded situations. Thus, automatic knowledge graph construction is required. COMmonsEnse Transformers (COMET) (Bosselut et al., 2019) is a generative model of commonsense knowledge trained with existing tuples as a seed set of knowledge. A generative approach to knowledge base construction is developed and large-scale transformer language models are used to create commonsense knowledge tuples. Unseen words or events can be automatically linked to the original knowledge bases such as ConceptNet and ATOMIC to enrich the knowledge. We can query words or sentences in the knowledge base to find out relevant tuples with certain kinds of relations. Therefore, the sentence to be classified can be enriched with more relevant information, which helps to identify whether it is funny or not.

3 Methodologies

In this section, we introduce our methodologies including the basic formulation of this task, our network architecture, the knowledge graph data augmentation and the training scheme.

3.1 Basic Formulation

Given the headline vocabulary: $\mathcal{V}$ and the set of possible grades: $\mathcal{G} = \{0, \dots, G\}$, where G is the maximal grade and $G = 3$ in this task, we denote an input sequence with length n as $\mathbf{x} = [x_1, \dots, x_n] \in \mathcal{V}^n$ and the corresponding grades given by the M human graders as $\mathbf{y} = [y_1, \dots, y_m] \in \mathcal{G}^m$.

The target is to train a neural network parameterized by θ: $\mathcal{F}_\theta : \mathcal{G}^n \rightarrow \mathcal{R}$ to predict the mean grade $\tilde{y}_{\text{mean}} = \mathcal{F}_\theta(\mathbf{x})$ such that the RMSE between the predicted mean grade $\tilde{y}_{\text{mean}}$ and the ground truth mean grade $y_{\text{mean}} = \frac{1}{m}\sum_{i=1}^{m} y_i$ can be minimized, i.e. find the set of parameters θ^* such that:

$$\theta^* = \arg\min_\theta \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_{\text{mean}}^{(i)} - y_{\text{mean}}^{(i)})^2} \quad (1)$$

Where N is the total number of training samples and the superscript $^{(i)}$ indicates the i-th training sample.

3.2 Network Architecture

In this work, we adopt the similar architecture as the BERT (Devlin et al., 2018), which has been seen as a powerful network in the NLP field. The BERT architecture uses the Transformer Encoder (Vaswani et al., 2017) as its main component for contextual feature extraction. Besides, it introduced segment embeddings and a new training scheme.

An overview of our network can be found in Figure 1. Both the original sequence and the edited sequence are feeded to our network. The grade loss (introduced in Section 3.4.1) and the BERT loss are jointly trained.

We also use an additional news headline dataset to pretrain our network. During pretraining, we only input one sequence to our network as shown in Figure 2. We set the grade to 0 during pretraining, it will not effect the training process as the loss quickly goes to 0 but increase the reusability of code.

3.3 Assembling of Input Sequences

We introduce two assembling methods to convert a data sample to an input sequence $\mathbf{x}$: *basic assembling* and *knowledge graph assembling*. Basic assembling uses only the headlines from the humicroedit dataset, while knowledge graph assembling utilizes knowledge graphs to incorporate relevant text hence augments the original dataset.

3.3.1 Basic Assembling

One obvious way to assemble one input sequence is to directly use the edited headline as the input to the network and let the network directly predict how funny the input sequence is. However, in this task, human graders does not assign grades directly based on the edited headlines but assigned to edited headlines against the original headline. Hence, a better way is to combine both the original and edited sequence together as the input sequence.

Given the original headline $\mathbf{x}_o$ and the edited headline $\mathbf{x}_e$, we assemble the input sequence $\mathbf{x}$ the concatenation of this two sequences: $\mathbf{x} = [\langle cls \rangle, \mathbf{x}_o, \langle sep \rangle, \mathbf{x}_e]$. Where $\langle cls \rangle$ is for training usage, which will be elaborated in Section 3.4.1, and $\langle sep \rangle$ is a special token for separating

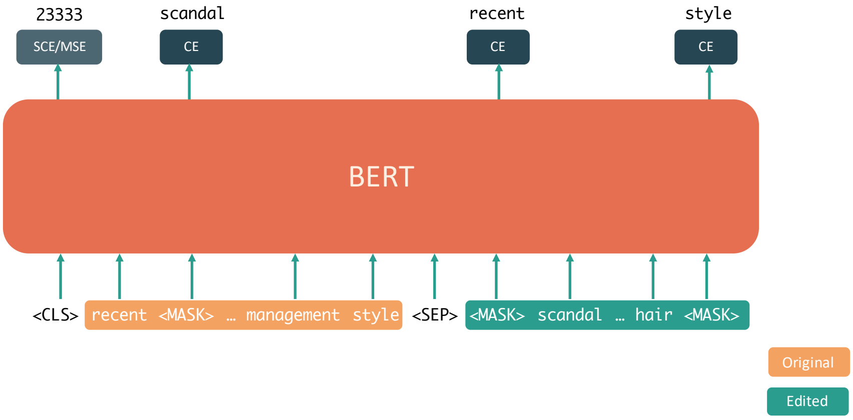


Figure 1: Our framework. We input the original sequence and the edited sequence simultaneously to our network. The grade loss and the BERT loss are jointly trained. The sample sequence shown in the figure is *Recent Scandals Highlight Trump 's Chaotic ~~Management~~* ***Hair*** *Style* , and the five grades are 2,3,3,3 and 3.

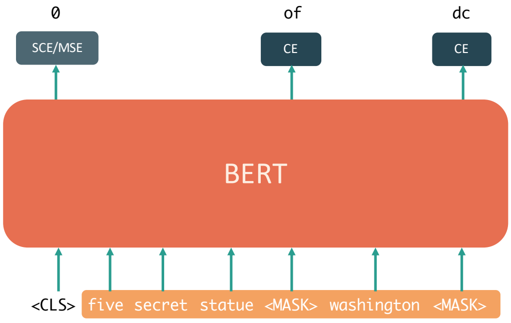


Figure 2: Pretraining on the Examiner dataset. The sample sentence shown in the figure is *Five Secret Statue of Washington DC*.

two headlines segments $\mathbf{x}_o, \mathbf{x}_e$. Note that the two segments use different segment embeddings when feeded into the neural network.

3.3.2 Knowledge Graph Assembling

To further augment the input sequence for training, we propose to use knowledge graph to generate the related knowledge graph events for each headline. Instead of directly querying from the knowledge bases using the preprocessed headlines, we use the COMET (Bosselut et al., 2019) models pretrained on the ATOMIC (Sap et al., 2019) and ConceptNet (Liu and Singh, 2004) respectively to generate the objective event given a subjective event and a relation, which is able to handle the unseen terms that cannot be directly queried from a knowledge base.

Given an original headline $\mathbf{x}_o$, we extract one event from this headline e_o and then use this event together with relation r_k to generate the consequential objective events $e_o^{r_k}$. An augmented sequence for the original headline is assembled as: $\tilde{\mathbf{x}}_o = [\mathbf{x}_o, \langle sep\rangle, e_o^{r_1}, \langle sep\rangle, \ldots, \langle sep\rangle, e_o^{r_K}]$, where K is the number of relations in the database. The same for the edited headline $\mathbf{x}_e$. Finally, we assemble our augmented input sequence as: $\mathbf{x} = [\langle cls\rangle, \tilde{\mathbf{x}}_o, \langle sep\rangle, \tilde{\mathbf{x}}_e]$.

For the ATOMIC COMET model, we use the longest skeleton sentence that contains the edited (or editing) term as the subjective event to query the related objective events, where the skeleton sentences are generated using the Stanford CoreNLP (Manning et al., 2014) toolkit. There are totally 9 relations for the ATOMIC COMET model. For the ConceptNet COMET model, we use the edited (or editing) term as the knowledge graph events directly. The number of relations in the ConceptNet is 34.

3.4 Training Objectives

We introduce two kinds of training losses. The first is the grade loss, which is for the grade estimation. The second is the BERT loss, which adds more supervision for training.

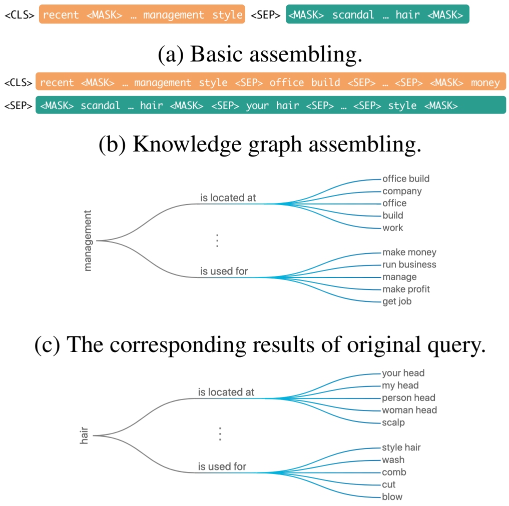
3.4.1 Grade Loss

For the mean grade regression, we introduce two choices: the mean square error (MSE) loss and the soft cross entropy (SCE) loss.

Mean Squared Error (MSE) Loss: The basic idea is to adopt the mean square loss as objective and use the ground truth mean grade as the target. Given a data sample $(\mathbf{x}, \mathbf{y})$, the MSE loss can be written as:

$$\mathcal{L}_{MSE}(\mathbf{x}, \mathbf{y}) = (\mathcal{F}_\theta(\mathbf{x}) - y_{\text{mean}})^2 \qquad (2)$$

Soft Cross Entropy (SCE) Loss: To further



(a) Basic assembling.

(b) Knowledge graph assembling.

(c) The corresponding results of original query.

(d) The corresponding results of edited query.

Figure 3: Comparison between basic assembling and knowledge graph assembling on ConceptNet. The orange sequence is for the original headline and green one is for the edited headline. In the example shown in this figure, the objective events are generated with beam size 5. Only the first no empty queried event will be assembled to the input sequence.

improve the label granularity, we propose to use the soft cross entropy loss, which is defined as the cross entropy between the empirical grade distribution $\hat{p}(g|\mathbf{x})$ and the model grade distribution $p_\theta(g|\mathbf{x})$:

$$\mathcal{L}_{SCE}(\mathbf{x}, \mathbf{y}) = -\sum_{g=0}^{G} \hat{p}(g|\mathbf{x}) \log p_\theta(g|\mathbf{x}) \quad (3)$$

Where G is the maximal grade and $\hat{p}(g|\mathbf{x})$ is a normalized count over grades given by $\mathbf{y}$.

When SCE loss is adopted, we change the output of the network from 1-dimensional regression layer to a G-dimensional softmax layer, which models $p_\theta(g|\mathbf{x})$. For prediction, the expectation of g: $E[g|\mathbf{x}] = \sum_{g=0}^{G} g \cdot p(g|\mathbf{x})$ is used as the predicted mean grade, i.e. $\mathcal{F}_\theta(\mathbf{x}) := E[g|\mathbf{x}]$ for the SCE loss based model.

3.4.2 BERT Loss

To enforce the supervision of model training, we adopt the BERT Loss $\mathcal{L}_{BERT}(\mathbf{x})$, which is a cross entropy loss over the masked tokens. This loss forces the network to predict the masked token based on the contextual, which enhances the contextual feature extraction of the transformer layers.

3.4.3 Overall Loss

Finally we give the overall training loss:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathcal{L}_G(\mathbf{x}, \mathbf{y}) + \lambda \mathcal{L}_{BERT}(\mathbf{x}) \quad (4)$$

Where $\mathcal{L}_G$ is the grade loss, which can be either $\mathcal{L}_{MSE}$ or $\mathcal{L}_{SCE}$, λ is a hyperparameter to balance between the two loss, which is set as $\lambda = \frac{1}{\log |\mathcal{V}|}$ in our work.

4 Experiments

4.1 Dataset and Experimental Settings

As required, we adopt the Humicroedit (Hossain et al., 2019) task-1 dataset for our experiments. At the time of writing, only the training set and development set are provided, thus we further split official training set to 95% local training set and 5% local validation set, which contains 9169 and 482 data samples respectively. For the development set, there are 2419 data samples. Each data sample in the training set contains an original sentence, an edit word and grades from $5/10/15$ graders.

For experiments, we implement our network using PyTorch (Paszke et al., 2017) and run our experiments on either a GeForce RTX 2070 or NVIDIA Tesla M60 depends on which GPU is available at the time of running the experiment. The first GPU is preferred if both are available.

For the network architecture, we use $l = 4$ layers transformer encoder layers with the number of heads $h = 4$, dimension $d = 128$, feed-forward dimension $d_{ff} = 512$, and dropout probability $p_{\text{dropout}} = 0.1$. For BERT training, we randomly mask out 15% non-special (i.e. not $\langle sep \rangle$, $\langle cls \rangle$, $\langle unk \rangle$ etc.) input words.

For all the trainings, we use the Adam (Kingma and Ba, 2014) optimizer with learning rate $\eta = 3 \times 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$. For all training on the Humicroedit dataset, the batch size is set to 32. In some experiments, we also use the Examiner (Deceased, 2018) dataset to pretrain our model, where the batch size is set to be 256. For the pretrainings on the Examiner dataset, we only use the first $300,000$ samples instead of the full dataset for time efficiency. We select the most common 10,000 words from the Examiner $300,000$ dataset as our vocabulary no matter pretraining is adopted or not to ensure the consistency of vocabulary.

[1] The top-1 result of this challenge by other competitors at the time of writing.

Method	RMSE (val)	RMSE	RMSE@10	RMSE@20	RMSE@30	RMSE@40
baseline	-	0.57840	1.00394	0.83760	0.72726	0.64419
state-of-the-art[1]	-	**0.51942**	**0.86123**	**0.72373**	**0.63351**	**0.57022**
BERT	0.53963	0.57640	0.98788	0.82771	0.72110	0.64037
BERT (SCE)	0.54360	0.58094	0.99592	0.83434	0.72659	0.64534
BERT+PT	0.54443	0.57764	0.96635	0.81452	0.71033	0.6360
BERT+PT (SCE)	0.53160	0.58664	0.99645	0.83341	0.72628	0.64835
BERT+KGA	0.53292	0.57769	0.98525	0.82781	0.72090	0.64057
BERT+KGA (SCE)	0.54489	0.59467	0.99590	0.83575	0.73023	0.65428
BERT+PT+KGA	0.53000	0.57396	0.95943	0.80752	0.70827	0.63369
BERT+PT+KGA (SCE)	0.54074	0.60089	0.98117	0.82994	0.73000	0.65761
BERT+KGC	0.52545	0.55931	0.94898	0.79645	0.69517	0.62001
BERT+KGC (SCE)	0.51352	0.56083	0.91917	0.77136	0.67649	0.61272
BERT+PT+KGC	0.51827	0.56239	**0.89351**	**0.76046**	0.67146	0.61068
BERT+PT+KGC (SCE)	0.52631	**0.54982**	0.91411	0.76439	**0.67107**	**0.60364**

Table 1: Our experimental results. BERT stands for our BERT framework. PT indicates the model is pretrained on the Examiner dataset. KGA or KGC indicate using knowledge graph ATOMIC or ConceptNet to augment the input data respectively. SCE is for the soft cross entropy grade loss. If not indiciated, MSE is used. The RMSE (val) column stands for the RMSE on the local validation set. The other RMSE-s are reported on the development set as the test set has not been published at the time of this writing.

For objective event retrieval, we use the pretrained model provided by the COMET Github Pepository[2]. We use beam width 10 for the ATOMIC knowledge graph and beam width 3 for the ConceptNet knowledge graph for decoding. We retrieve all 9 relations and 34 relations respectively on the ATOMIC knowledge graph and ConceptNet knowledge graph. As described in Section 3.3.2, we use the skeleton sentence as the subjective event for the retrieval on the ATOMIC knowledge graph and the edited (or editing) word as the subjective event for the retrieval on the ConceptNet knowledge graph.

For evaluation, we adopt the metrics officially provided by this challenge: RMSE, RMSE@10, RMSE@20, RMSE@30 and RMSE@40. Here, RMSE is the root mean squared error on the overall test set, RMSE@N, $N \in \{10, 20, 30, 40\}$, is the RMSE on the subset of the test set where only the $N\%$ most funny headlines and $N\%$ least funny headlines in are taken from the test set.

As there is a curve overfitting observed after training more than 15 epochs, we only train our model for 15 epochs. After training, we select the epoch that provides the minimal RMSE on the local validation set for submission. All the results on the development set provided in the following sections are collected from the official online evaluation system[3].

4.2 Results and Analysis

The overall experimental results are shown in Table 1. We first provide the baseline result and the state-of-the-art result at the time of writing. Then we provide the results without any knowledge graph augmentation. After this, we provide the results obtained with the ATOMIC knowledge graph and ConceptNet knowledge graph respectively.

The baseline result[4] is obtained by using the average mean grades over the training set (i.e. 0.93557) as the predictions. While the state-of-the-art results is generated by other competitors whose method is still unrevealed.

The results without knowledge graph augmentation is very closed to the baseline result. We believe this is caused by the limited size of the training set thus the network turns to give the grades centered around 0.93. However, the results generally outperform the baseline sightly on RMSE@10 - RMSE@40 samples since there are more variance on our predictions as the network may grasp some faint information and show preference when grading some samples.

The performance does not improve given the ATOMIC knowledge graph (KGA). We believe the major cause is located at the knowledge graph retrieval process. We found that for most of the data samples, the objective events retrieved by the

[2] https://github.com/atcbosselut/comet-commonsense

[3] https://competitions.codalab.org/competitions/20970

[4] https://github.com/n-hossain/semeval-2020-task-7-humicroedit

Text	oEffect	oReact	oWant	Predicted / GT
medicaid director issue **warning** on new obamacare repeal bill	-PRON- be grateful to personx	grateful	to listen to personx	
medicaid director issue **fatwa** on new obamacare repeal bill	the people of the new obamacare affect by the new obamacare	grateful	to thank personx	0.839 / 0.600
# womensmarch against **donald trump** around the world	lose respect for personx	annoyed	to listen to personx	
# womensmarch against **man** around the world	man lose respect for personx	sad	to get away from personx	0.690 / 1.000
congo 's mining revenue ' **miss** ' - global witness	get yell at by personx	angry	to ask personx what -PRON- be do	
congo 's mining revenue ' **steal** ' - global witness	lose money	angry	to find out what personx be up to	0.281 / 0.200
this be what happen when -PRON- let trump be **trump**	lose the game	happy	to listen to personx	
this be what happen when -PRON- let trump be **orange**	want to see how -PRON- do	happy	to have fun with personx	0.928 / 2.000
among republicans , trump be more popular than congressional **leader**	people listen to personx	annoyed	to listen to personx	
among republicans , trump be more popular than congressional **interns**	people do n't listen to personx	annoyed	to listen to personx	1.082 / 0.800
state official blast ' unprecedented ' dhs **move** to secure electoral system	people lose trust in personx	secure	to thank personx	
state official blast ' unprecedented ' dhs **idea** to secure electoral system	-PRON- do n't have to worry about -PRON- anymore	happy	to listen to personx	0.423 / 0.000
protester rally for **refugee** detain at jfk airport after trump ban	people have to go to the airport	sad	to thank personx	
protester rally for **stewardesse** detain at jfk airport after trump ban	-PRON- go to the airport	nervous	to go to the airport	0.714 / 0.400
cruise line carnival corp . join the fight against bermuda 's same - sex **marriage** ban	lose money	embarasse	to get rid of -PRON-	
cruise line carnival corp . join the fight against bermuda 's same - sex **raisin** ban	lose money	annoyed	to get rid of -PRON-	0.683 / 0.600
columbia police hunt woman see with **gun** near university of missouri campus	person y get shoot .	scared	to tell personx to keep away from the gun	
columbia police hunt woman see with **cake** near university of missouri campus	-PRON- eat the cake too	happy	to eat the cake	1.105 / 1.400
here be what be in the house - approve health **care** bill	be grateful to personx	grateful	to thank personx	
here be what be in the house - approve health **food** bill	be grateful to personx	grateful	to thank personx	0.935 / 0.400

Table 2: Examples from the local validation set based on the ATOMIC knowledge graph result, where GT is short for ground truth graded by human.

Text	At Location	Capable Of	Causes	Predicted / GT
medicaid director issue **warning** on new obamacare repeal bill	war zone	make person nervous	death	
medicaid director issue **fatwa** on new obamacare repeal bill	north america	eat chicken	death	0.760 / 0.600
# womensmarch against **donald trump** around the world	tv show	act as moderator	death penalty	
# womensmarch against **man** around the world	office	think critically	get hurt	1.137 / 1.000
congo 's mining revenue ' **miss** ' - global witness	school	give -PRON- flower	-PRON- feel sad	
congo 's mining revenue ' **steal** ' - global witness	jail	make -PRON- rich	punishment	0.579 / 0.200
this be what happen when -PRON- let trump be **trump**	theater	play hardball	death	
this be what happen when -PRON- let trump be **orange**	fridge	be orange or red	indigestion	1.066 / 2.000
among republicans , trump be more popular than congressional **leader**	army	lead person	leadership	
among republicans , trump be more popular than congressional **interns**	work	work in office	stress	0.631 / 0.800
state official blast ' unprecedented ' dhs **move** to secure electoral system	theater	slow thing down	movement	
state official blast ' unprecedented ' dhs **idea** to secure electoral system	church	divide person	solution	0.575 / 0.000
protester rally for **refugee** detain at jfk airport after trump ban	refugee camp	live in refugee camp	death	
protester rally for **stewardesse** detain at jfk airport after trump ban	hotel	clean room	death	0.986 / 0.400
cruise line carnival corp . join the fight against bermuda 's same - sex **marriage** ban	marriage	last forever	child	
cruise line carnival corp . join the fight against bermuda 's same - sex **raisin** ban	jar	grow up to be adult	-PRON- get fat	1.106 / 0.600
columbia police hunt woman see with **gun** near university of missouri campus	police station	fire bullet	death	
columbia police hunt woman see with **cake** near university of missouri campus	birthday party	taste good	make person happy	1.128 / 1.400
here be what be in the house - approve health **care** bill	home	make -PRON- feel good	-PRON- feel good	
here be what be in the house - approve health **food** bill	pantry	taste good	-PRON- feel full	0.997 / 0.400

Table 3: Examples from the local validation set based on the ConceptNet knowledge graph result, where GT is short for ground truth graded by human.

original sentence and the edited sentence are exactly the same (see Section 4.3), which provides no additional information for training. The causes of issue could be the following: 1. The edited (or editing) word is not attended by the COMET network. 2. The skeleton contains no edited (or editing) word, thus whole sentence is feeded into the COMET network. As necessary cropping is adopted to make the input within the limited length, the edited (or editing) word may be discarded before feeding into the COMET network.

We obtain a general performance gain after adopting the ConceptNet knowledge graph (KGC) instead of the ATOMIC knowledge graph. Our best result 0.54982 RMSE reduces the baseline RMSE by around 5%. The improvement of the results benefits from the input sequence augmentation. As described in Section 3.3.2, we only use the edited (or editing) word to retrieve knowledge graph entries, thus provides different objective events and more discriminating information. The details could be found in the Section 4.3.

By comparing the results from the pretraining, we found that the BERT pretraining generally helps reduce RMES@10 - RMSE@40 by 0.01 - 0.05. However, in most of the cases, the pretraining increases the overall RMSE loss. We assume the pretraining is providing more variance for the predicted grades, thus reduces the RMSE on the samples of off-centered grades. While it failed to generate results with lower RMSE on the overall test set comparing to the conservative predictions.

For the MSE and SCE losses, we found if the information is not sufficient, the MSE loss always perform better. While if the ConceptNet knowledge graph is provided and the pretraining is adopted, the SCE loss will give better results comparing to the MSE loss. We believe the fine-granularity of the labeling will actually help when the training data is sufficiently large.

4.3 Qualitative Study

In this section, we provided qualitative study on our methods by showing some examples on the local validation set.

We provides the knowledge graph retrieval from the first 3 relations **oEffect**, **oReact**, **oWant** for Examples in Table 2. Many duplicate objective events can be observed from the table. For example, the **oReact** for the original and edited headline for the first sample are grateful. And the **oWant** for the fifth sample are *to listen to personx*. The duplicate events fail to provide additional information and thus not providing performance gain as much as the model trained with the ConceptNet knowledge graph (whose examples shown in Table 3).

5 Conclusion and Future Work

In this work, we propose to incorporate knowledge graph into the BERT framework to predict the funniness grade when a micro-edit is applied on a news headline. We use the original and edited headline to query the COMET network and use the query results to augment the input sequence of the BERT framework. We evaluate our BERT framework with both from-scratch training and pretraining on an external news dataset Examiner. Our results show when the pretraining is adopted and the ConceptNet pretrained COMET is used, we can reduce the RMSE on the overall development set by around 5% comparing to the official baseline result.

Currently, we only use a COMET network pretrained on ConceptNet and ATOMIC, while the amount of relations is limited and the knowledge base is relatively small. For the further work, we may try a COMET model pretrained on a larger knowledge base like ASER. Moreover, as we only use the first 300,000 data samples from Examiner for the BERT pretraining, the amount of data used for pretraining may not be sufficient enough for the task. We will try to adopt larger corpus for the network pretraining in our future work.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

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)*.

Deceased. 2018. The Examiner - Spam Clickbait Events.

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*.

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*.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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 *Advances in neural information processing systems*, pages 5998–6008.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, Cane Wing-Ki, et al. 2019. Aser: A large-scale eventuality knowledge graph. *arXiv preprint arXiv:1905.00270*.