Stay on Topic, Please: Aligning User Comments to the Content of a News Article

Jumanah Alshehri[1][0000−0002−0077−7173], Marija Stanojevic[1][0000−0001−8227−6577], and Eduard Dragut[1][0000−0002−3103−054X] Zoran Obradovic[1][0000−0002−2051−0142]

Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA, USA
{jumanah.alshehri,marija.stanojevic,edragut,zoran.obradovic}@temple.edu

Abstract. Social scientists have shown that up to 50% of the comments posted to a news article have no relation to its journalistic content. In this study we propose a classification algorithm to categorize user comments posted to a news article based on their alignment to its content. The alignment seeks to match user comments to an article based on similarity of content, entities in discussion, and topics. We propose a BERTAC, BERT-based approach that learns jointly article-comment embeddings and infers the relevance class of comments. We introduce an ordinal classification loss that penalizes the difference between the predicted and true labels. We conduct a thorough study to show influence of the proposed loss on the learning process. The results on five representative news outlets show that our approach can learn the comment class with up to 36% average accuracy improvement comparing to the baselines, and up to 25% comparing to the BA-BC. BA-BC is our approach that consists of two models aimed to capture dis-jointly the formal language of news articles and the informal language of comments. We also conduct a user study to evaluate human labeling performance to understand the difficulty of the classification task. The user agreement on comment-article alignment is "moderate" per Krippendorff's alpha score, which suggests that the classification task is difficult.

Keywords: text mining · text classification · online news · news comments · relevancy · understanding user-generated text

1 Introduction

The study of user comments is essential for social scientists, policymakers, and journalists since virtual discussions offer an insight into the public opinion. In 2020, people shifted more toward online discussions due to COVID-19. Many survey-based studies tried to understand the users' behavior by characterizing and categorizing comments in online news [6, 10, 22, 28]. A salient outcome of these studies is that 20% to 50% of users' comments are irrelevant to the content or topic of those articles since users drift from the original topic to irrelevant

sub-discussions [15]. Our goal in this work is to understand commenting behavior, more precisely, to automatically identify the subset of comments, from the set of comments an article receives, that are pertinent to the content of the article. The challenge is multi-fold: e.g., comments tend to be terse, colloquial, often nonliterary, containing grammatical errors, misspellings, and punctuation misuse. Our premise is that users are inclined to write comments that diverge from the article topic to different extents, especially in lengthier discussions. This noise in the data affects downstream applications such as opinion mining.

Previous studies tried to remove the noise among comments by studying toxic comments [13, 31], topic drifting [1, 34], and understanding the quality of online news comments [3, 27, 32]. From NLP perspective, this problem is a supervised classification task to separate relevant from irrelevant comments.

In this paper, we introduce the Article-Comment Alignment Problem (ACAP). We aim to define a set of article-comment relevance classes and propose a methodology to classify article-comments pairs automatically. ACAP is a challenging task, for example, consider the article *"This is going to happen in the United States: Donald Trump calls for surveillance of Muslims and advocates waterboarding terror suspects after Brussels attack"* [1] from *Daily Mail* and the comment *"It's not Europe anymore. It's Eurabia. This should not be a news story anymore."* Two human annotators rate the comment as *Irrelevant*, while the third annotator rates it as *Same Category*. The third annotator's label is the most appropriate, but choosing that category requires background knowledge on the political circumstances in Europe in 2016. In solving ACAP, we hypothesize the following: 1) It is possible to capture the extent of a connection and semantics between an article and its comments using globally pre-trained models, fine-tuned with local data. 2) Considering the natural order of labels during training will boost the algorithm learning process.

We test our hypotheses in the following practical scenarios: (1) limiting amount of labeled article-comment pairs (1K per dataset), (2) bounding the number of tokens from each document (article or comment), and (3) concomitantly working with formal text, in the form of news articles, and informal text, in the form of comments. The pairs are extracted from five online news outlets: *Wall Street Journal* (WSJ), *Fox News* (FN), *Daily Mail* (DM), *The Guardian* (TG), and *Market Watch* (MW). This work makes the following contributions: 1) We introduce the Article-Comment Alignment Problem (ACAP) and analyze the hardness of ACAP using an agreement study on the classification of human annotators. 2) We propose BERTAC, which jointly learns embedding representations for articles and their comments, to solve ACAP. We also propose BA-BC, which consists of two models on trained on articles and the other on comments, which attempts to capture the difference in language style between them, formal versus informal. We compare it to several approaches, including BA-BC, and show its superior performance. 3) We develop a novel ordinal classification loss for BERTAC that penalizes the difference between the predicted and true labels. The proposed loss exhibits similar performance to the original loss in

[1] Full article: https://dailym.ai/2Qz7RG9

terms of accuracy, however, it boosts the model performance when trained on high agreement examples. 4) We conduct extensive empirical studies on articles and comments from 5 representative news outlets.

2 RELATED WORK

User comments are a powerful means to understand public opinion and reaction to emerging events. Many organizations invest in mining user comments to improve their decision making. News outlets and social platforms are recommending most relevant user posts to keep the attention of busy readers [28]. Many studies focus on mining the user opinion from social media [9, 23, 24] and online news comments [21, 30]. Other works look into bias in the news, and its influence on user-generated content [25]. The main challenge in those studies is the unpredictable quality of user-generated content.

To solve this problem, a line of research focuses on comment drifting [1, 34] by utilizing the temporal nature of comments. The older an article is, the more commentators it has, and the probability of exposure to topic drift is higher [3]. This phenomenon influences the quality of comments and their relevance.

Another line of work [8, 20, 26] investigates which part of an article a comment aligns with using statistical models, while other [32] use hand-crafted structural, lexical, syntactic, discourse, and relevance features as an input to the logistic regression. Even though their F1 score is in the range of $70-80\%$ and their analysis measures correlation of the attributes with the label, hand-crafting features for each problem is difficult and time-consuming.

The work most related to ours attempts to automatically classify paragraph-comment agreement [3]. They labeled the data based on Likert scale categories [29], which is criticized for introducing bias. For instance, a number of works show that user responses are significantly affected by the order and direction of the rating scale [12, 33].

Instead, we propose to use transformer pre-trained language approach [7, 14]. We also create a new ordinal classification loss. As shown in the experimental study, our approach performs significantly better than the baselines on ACAP. We work with three annotators. Their labels give us support data to study the difficulty of the problem. We show that the annotators exhibit only fair agreement, indicating that ACAP is a difficult problem even for human beings.

3 DATASETS

News articles and their comments were collected between 2015 and 2017 from Google News [11]. The dataset has over 19K articles with 9M comments. For this study, we chose five news outlets that are representative of the problem at hand. The dataset contains articles and comments with a broad range of lengths and different number of comments. This data allows us to test the behavior of the proposed models under varied settings. Table 1 (A) shows the statistics of datasets.

Table 1. (A) Statistics by outlet. We randomly selected 1K article-comments pairs from each outlet and labeled them. ALA is the articles' average length and ALC is average comments' length, measured by number of words. (B) Classes proportions for each dataset. Outlets are sorted based on total number of available articles per outlet.

Outlet	(A) Dataset Statistics			
	#Art.	#Comm.	ALA	ALC
FN	0.3K	72K	250	22
TG	1.6K	428K	797	54
MW	1.7K	65K	512	42
WSJ	3.6K	309K	164	57
DM	10K	1,012K	487	28

(B) Classes proportion			
Relevant	Same Ent.	Same Cat.	Irrelevant
3%	21%	29%	47%
5%	39%	32%	24%
7%	51%	20%	22%
8%	25%	34%	33%
15%	17%	20%	48%

3.1 Labeling

We discard all articles without comments. We randomly select 1K article-comment pairs from each outlet. Then, annotators manually and independently label the pairs in four classes: Relevant, Same Entities, Same Category, and Irrelevant. Relevant class - the content of the comment discusses the same matter as the article. Same Entities class - the comment is not directly relevant, however, it mentions the same main entities within the same scope (category) of the article. For example, the article talks about a Real Madrid - F.C. Barcelona game, mentioning Ronaldo's performance in the game, and the comment talks about Ronaldo's best goal in the Portuguese team. Same Category class - comment in this class is not discussing the article, but it falls into the same category as the article. For example, both comment and article are discussing politics. Irrelevant class - a comment is in this class if it does not belong to any other class.

Figure 1 shows labeled examples of each class from WSJ. First column is part of the article; second column has four comment examples, each example represent a different class. The article in the table discusses Hillary Clinton's email story that came out before the 2016 U.S. election. The first comment is *Relevant*, the second comment does not discuss the main issue, however, it mentions some of the entities discussed in the article within the category of the article (politics). Hence, its class is *Same Entities*. The third comment does not refer to any named entity from the article, but it discussed another political issue. Thus, its class is *Same Category*. In the last comment, the user believes that he looks like *Joe Friday*. This has no connection with the article, therefore, the comment is deemed *Irrelevant*.

To obtain labeled instances, we asked three native English speakers, who were not involved in problem modeling nor solving it, to annotate the article-comments pairs. We provide them with the following information: 1) an article-comment pair without the surrounding context (i.e., the parent and child comments), and 2) the four label categories with an explanation and an example for each of them. Each pair receives 3 labels, one from each annotator. We assign the final label using an averaging aggregation scheme. We map Irrelevant, Same Category, Same Entity, and Relevant to $0, 1, 2$, and 3, respectively. We average

First Part of the article	Comment	Class
The *Clinton Campaign* at *Obama Justice* Emails on *WikiLeaks* show a top federal lawyer giving *Hillary* a quiet heads up. *President Obama* and *Attorney General Loretta Lynchat* the *White House* in July. The most obnoxious pin of the 2016 campaign cames this week, as *Democrats*, their media allies and even *President Obama* accused the *FBI* of stacking the election. It's an extraordinary claim, coming as it does from the same that has -we now know – been stacking the crew election all along in the corridors of the *Justice Department*. This is the true November surprise. For four months, *FBI Director James Comey* has been the public face of the investigation into *Hillary Clinton*'s email server. He played that role so well, putting the *FBI* so front and center, that the country forgot about *Mr.Comey*'s bosses.	As a practicing lawyer this is just embarrassing. Lawyers are governed by a *Code of Professional Conduct*. The *Justice Department* has enormous power in our country. Politics is not supposed to be part of the equation. It is now abundantly clear that politics is now game on for the *Obama Justice Department*.	Relevant
	The news article that never was written is how *Obama* has corrupted the government agencies and how *Clinton* will continue the process? *Kim* has done the best job of placing the blame for the political corruption of these agencies? The *IRS State Dept. Justice Dept.* and compass because he has perpetuated this for political gain	Same Entity
	We allowed *binladen* family to fly out during 911 blackout as soon I read that in the news I swore never to vote *Bush* again.	Same Category
	I always look liked *Joe Friday*	Not Relevant

Fig. 1. Labeling example, entities are colored. The article category is politics.

the (users) scores per pair and round to the nearest integer, which becomes the label of the pair. For example, a pair x-y receives the score $1, 1$, and 2 will have a label 1, which corresponds to "Same Category". Table 1 (B) shows the proportion of each class per outlet. We also binarize labels, by assigning 0 to Irrelevant comments, and 1 to rest of the labels .

3.2 User Agreement Study

ACAP is not an easy task, uing Fleiss Kappa statistic and Krippendorff's alpha coefficient we compare the agreement between annotators. *Fleiss Kappa* statistic [18] calculates agreement between multiple scorers as in Equation 1. The interpretation of Kappa value is, < 0 is poor agreement, [0.01,0.20]= slight agreement, [0.21,0.40]= fair agreement, [0.41,0.60]= Moderate agreement, [0.61,0.80]= substantial agreement, and [0.81,1.00]= perfect agreement.

$$FK = \frac{\sum_{i=1}^{N} \sum_{j=1}^{k} v_{ij}^2 - Nm}{Nm(m-1)} \tag{1}$$

To account for the error magnitude that a scorer makes, we use *Krippendorff's alpha coefficient* [19], this statistic consider the distance between labels given by multiple scorers as in equation 2. $\alpha \in [0, 1]$, where 0= random scoring, and 1= perfect scoring.

$$\alpha = 1 - \frac{(n-1) \sum_i \sum_j o_{ij} * \delta_{ij}^2}{\sum_i \sum_j v_i * v_j * \delta_{ij}^2} \tag{2}$$

Table 2 gives the agreement scores between annotators per dataset. We noticed that labeling WSJ is the hardest. The raters' agreement is "Fair" for WSJ, TG, DM, and MW and "Moderate" for FN, based on Fleiss Kappa. The Krip-

pendorff's score[2] is between 42% and 66% across outlets. Results indicate the difficulty of assigning a category for comments in general.

Table 2. Agreement analysis for annotators labels.

Dataset	WSJ	TG	DM	MW	FN
Fleiss Kappa	0.22	0.36	0.37	0.40	0.45
Krippendorff's α	0.42	0.60	0.61	0.64	0.66

4 Methods

4.1 BERTAC Model - Joint Modeling of Article and Comments

BERTAC leverages BERT_{base} architecture, which allows us to learn more expressive embeddings for articles and comments. To solve ACAP we combine an article and its comment into a pair of segments and separate them with the special token [SEP]. Our goal is to make use of BERT's self-attention mechanism and bidirectional cross attention in an end-to-end fashion to encode the relevance between an article and its comments. One challenge in this setting is that of determining the length (in words) of the input segments that allow the network architecture to encode useful article-comment relations. We explore multiple lengths for each dataset based on the average length of the articles.

4.2 BA-BC Model - Disjoint Modeling of Article and Comments

Our problem consists of two main parts. The first part is the article, where the language is formal and usually formed of long sequences. The second part is the comment, where comments are often written in informal language and consist of short sequences. We explore if a mixture of different pre-trained models can better solve ACAP, we call it BA-BC shown in Figure 2. The model consists of two stages. The first stage *Fine-tune on News* has two sides: (BA) is a BERT_{base} architecture trained and fine-tuned on articles; the second side (BC) is a BERTweet architecture trained and fine-tuned on comments. BERTweet is a pre-trained model proposed by [7] and trained on English Tweets, the underlie architecture is RoBERTa [35]. Their results show that BERTweet outperform RoBERTa_{base} and XLM-R_{base} [2] in many tasks.

In the second stage, *Classification* stage, the output from the first stage is fed into a fully-connected layer with a ReLU non-linearity. To get full advantage of pre-trained models, we designed two versions of BA-BC. The first is called *BC-BA-Emb*, where the output from the *Fine-tune on News* stage contains the

[2] Calculted by http://dfreelon.org/utils/recalfront/recal-oir/ software using ordinal setting

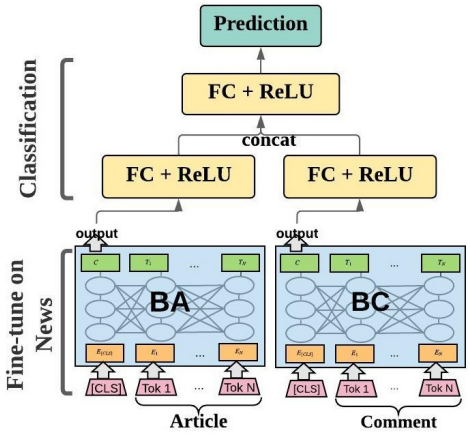

Fig. 2. BA-BC Model for ACAP

Embeddings created by both sides without seeing any training examples from our datasets. The second model, called *BC-BA-Fine-tune*, is additionally fine-tuned in the *Fine-tune on News* stage. Left-side (BA) is fine-tuned on articles and labels and right-side (BC) is fine-tuned on comments and labels. Then, last hidden state of fine-tuned parts is sent to the second, *Classification* stage.

4.3 Ordinal Classification Loss

We introduce the ordinal classification loss, which accounts for the distance between the predicted class and the actual class. Here, we multiply the loss for each example with a weight that is calculated according to equation 3, where $k = 4$ (number of classes), y_i is the actual label and $\bar{y_i}$ is the predicted label of the example.

$$weight = 1 + \frac{|\bar{y_i} - y_i|}{k - 1} \tag{3}$$

If the algorithm chooses the right class, the weight is 1, so the loss is equal to original loss for correct prediction. If the model predicts a wrong category, the classification loss is multiplied by 2, 3, or 4 based on the distance between the real class and predicted class. The proposed loss depends on both the difference between the predicted class and the correct class, and the softmax error during predicting the actual category. We incorporated the proposed loss into BERTAC and compare it to original loss in Section 6.4.

5 EXPERIMENTAL SETUP

5.1 Environment

We run BERTAC, BA-BC, and Siamese LSTM models on four large nodes with 512 GB of DDR4 2400MHz RAM. Each of those machines has two sockets with

Intel Xeon E5-2667 v4 3.2GHz processors, and every node contains two NVIDIA Tesla P100 PCIe 12GB GPUs and SSDs as local hard drives. Since doc2vec experiments are less computationally expensive, we run them on a 64-bit processor, Intel Core i7-6700 CPU @ 2.60 GHz with four cores and 16.0 GB RAM.

5.2 Model evaluation

To evaluate the performance of the models, we use both simple and weighted accuracy since our labels are ordinal. With simple accuracy metric, which is calculated as the percent of correct predictions, predicting 0 or 2 for label 3 are counted as equal mistakes. Instead, we use weighted accuracy to calculate error by summing the absolute difference between predicted class $\widetilde{s_i}$ and ground truth $\bar{s_i}$. Model's error on a dataset is calculated by dividing that error by the number of examples and max difference D between predicted classes. This constant is 3 in the given multi-class settings. The following formula computes the weighted accuracy, where m is the number of examples:

$$WACC = 1 - \frac{\sum_{i=1}^{m} |\widetilde{s_i} - \bar{s_i}|}{mD} \tag{4}$$

For all supervised models, experiments are repeated five times on different randomized split of labeled data. The dataset is split into 70:20:10 ratio for training, testing, and cross-validation, respectively. We report the mean and standard deviation.

5.3 Comparison models

A key challenge in solving ACAP is to establish a similarity of article-comment pair that is indicative of the relevance of the comment to the message of the article. We seek models that can learn long text representations using context and capitalize on the sequential nature of words in a comment. Besides, a model has to be able to embed two types of sequences: articles, which follow formal language, and comments, which may follow colloquial language.

Doc2vec is the first baseline. Since it is unsupervised we use all data, comments, and articles available in each outlet to learn the documents embedding. We learn separate embeddings per outlet to account for the linguistic style accommodations and other biases across outlets [4]. Then, we calculate the cosine similarity score for all labeled pairs and assign a class for each pair based on the rules written below. A represents articles, and C represents comments. We experiment with thresholds in increments of 0.1. The ones used below give the best performance:

$$f(A_i, C_i) = \begin{cases} 0, & \text{if } cos(A_i, C_i) \leq 0.4 \\ 1, & \text{otherwise, if } cos(A_i, C_i) \leq 0.6 \\ 2, & \text{otherwise, if } cos(A_i, C_i) \leq 0.8 \\ 3, & \text{otherwise.} \end{cases}$$

The second baseline is *Siamese LSTM*, which consists of two LSTM that learn representations of articles and comments in separate modules. On top of these

modules, there is a joint loss computation module, which computes the similarity between vectors and uses a dense layer to predict the label. Following [17], we use the Manhattan distance to calculate the similarity for an article-comment pair. We utilize a sparse categorical cross-entropy with a softmax activation function.

BA-BC: To produce the vector representation of articles and comments two main steps are required, embeddings and classifications. In *BA-BC-Emb* model each side learns the embeddings by fine-tuning on articles and comments, respectively, in an unsupervised manner (without using labels). However, for *BA-BC-Fine-tune*, each side is fine-tuned on articles and comments in a supervised way (using their associated labels). Later on, the vector representation of the last hidden layer is injected into the classification stage. The *Fine-tune on News* stage is repeated for 3 epochs and the final representation is fed into the *Classification* stage, in which cross-entropy with a softmax activation function is applied as a loss. The classification stage is repeated for 300 epochs.

BERTAC: We leverage BERT_{base} where it consists of 12 layers, hidden layer size is 768, number of self-attention heads is 12, and the total number of parameters is 110M. BERTAC is trained in two modes, cased and uncased, where letter casing is considered in the first while all letters are converted to small letters in the later. We trained the model for 6 epochs.

6 RESULTS AND DISCUSSION

We study the complexity of this problem. In addition, we thoroughly study the multi-class datasets by employing and analyzing multiple models and evaluation measures to understand their behavior and identify the ones that better capture the semantic between an article and its comments. We also analyze the effect of the proposed Ordinal Classification Loss.

6.1 Binary versus Multiclass ACAP

To characterize the complexity of our problem we compare a binary dataset and multiclass dataset using BERTAC. In Table 3, we can see that the model maximal performance is around 92% when the problem is binary. The accuracy drops between $13\% - 23\%$ when we have 4 classes. Even though having multiple classes helps people understand the relationship between an article and its comments better, it becomes harder for a model to capture the semantics and knowledge that is needed to distinguish some labels.

6.2 Models Comparisons on Multiclass ACAP

As shown in Figure 3, performance of Doc2Vec is the worst among all models, despite training on much larger corpus of unsupervised data. Siamese LSTM accuracy exceeds Doc2Vec with a boost between 1%- 27%. BA-BC-Emb and BA-BC-Finetune outperform Siamese LSTM, the current SOTA for this problem. Both have an increase of 4%- 17% in accuracy and 2%- 4% in weighted accuracy

Table 3. Test accuracy (in %) for BERTAC. The *B* row represents binary dataset and *M* row represents multiclass dataset. The average test accuracy of 5 experiments is reported with standard deviation.

Model	Dataset	FN	TG	MW	WSJ	DM
BERTAC	B	88.30(1.42)	92.45(1.45)	88.64(2.50)	85.07(0.97)	90.46(1.75)
	M	75.60(1.81)	74.58(6.49)	75.26(4.52)	63.17(2.44)	67.36(3.46)

over Siamese LSTM. When we compare BA-BC-Emb and BA-BC-Finetune we observe that they achieve similar performance. BERTAC however outperforms in both metrics all other models across all datasets when trained with the original loss function. In addition, we experiment with increasing the number of training points by merging the datasets. We note that increasing training examples does not improve any of the proposed models over BERTAC, which aligns with our hypothesis that BERTAC can outperform other models using only a small number of training examples.

Our experimental design is such that the number of articles varies between 300 and 10,000 across outlets as shown in Table 1. However, we label a fixed number of random outlet-comments. Therefore, the chance of selecting multiple comments for a single article is much larger at FN, TG and MW compared to WSJ and DM. A higher average accuracy is obtained on FN, TG and MW than on WSJ and DM. This suggests that when the model is trained on the same article with different comments and labels it can learn the pattern and predict the correct labels more accurately.

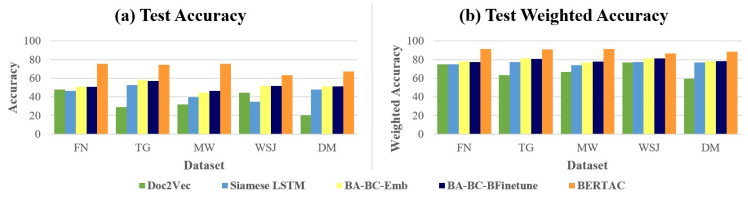

Fig. 3. The average test accuracy of 5 experiments (in %) for all models. (a) shows the accuracy results and (b) shows the weighted accuracy given by Equation 4.

6.3 Weighted versus Un-Weighted Accuracy

Comparing the outcomes of the algorithms on the weighted and unweighted accuracy, we note that their relative performance is unchanged: the model that has the lowest unweighted accuracy has the smallest weighted accuracy performance as well. The same relation stands for the highest results. There are a couple of explanations for this outcome. First, it suggests that, most of the wrongly classified instances are mixed with its neighboring classes. Second, it indicates a

proportional number of stronger misclassifications (where the distance between the actual and predicted category is larger than 1) is present across news outlets.

The analysis of weighted accuracy helps to gain additional insight into the models and the problem hardness. For instance, by comparing the weighted and unweighted accuracy scores, we get a better idea of how well a model learns, since it accounts for the strength of the error, penalizing more the mistakes on harder examples. We observe a large gap between accuracy and weighted accuracy results, where the error is between 2 or 3 times smaller in most cases. This indicates that when a model misclassifies an example, often, the model predicts one of the neighboring classes to the correct class. For example, in *WSJ*, *Doc2Vec*'s accuracy = 44.41% which is higher than of *Siamese LSTM*'s accuracy = 34.81%. However, the weighted accuracy shows the opposite: *Doc2Vec*'s accuracy = 76.88% and *Siamese LSTM*'s accuracy = 77.60%. This indicates that *Siamese LSTM* is able to understand the problem better and address the natural order of the classes during training.

6.4 Ordinal Classification Loss

We designed this experiment to investigate the effect of the *ordinal loss* on BERTAC. We hypothesize that BERTAC trained with the proposed *ordinal loss* will outperform the original loss.

We find that *proposed ordinal loss* has no significant advantage compared to *original loss*, where both losses have similar performance. To better understand this problem we investigate those instances where the annotators highly agree with each other in the labeling task: σ between the annotators's labels is either 0, which means that they all agree, or 0.5, which means that only one annotator disagree, with difference of 1 and this does not affect the final label after aggregating the annotators labels. We call this the *high agreement experiment* indicated by BERTAC$_{high}$ in Figure 4. On the other hand, the *low agreement experiment* indicated by BERTAC$_{low}$, which contains only examples where σ between the annotators' labels, is higher than 0.5. We find that for some datasets BERTAC$_{low}$ accuracy is slightly higher than BERTAC$_{all}$ and BERTAC$_{high}$. However, looking into the high σ we can see that the model is not consistent compared to BERTAC$_{high}$, and the number of examples are much fewer that BERTAC$_{all}$. Analyzing the original and ordinal losses for BERTAC$_{high}$, where annotators highly agree with each other, we find that the proposed ordinal loss is higher but not significantly. The improvement in accuracy is between $1\% - 5\%$ and $1\% - 3\%$ in the weighted accuracy. However, if we study the behavior across the models, we can see that ordinal loss behaves somehow differently across experiments. For examples in Figure 4, we can see that both BERTAC$_{high}$ and BERTAC$_{low}$ agree that ordinal loss is equal or better than original, where BERTAC$_{all}$ disagree. This brings the following question: *if the model were capable to vote for the best possible prediction from different model would this improve results?*

To answer the previous question, we calculate the average vote prediction from different models in order to obtain the best prediction. We consider the predictions from BERTAC uncased trained with ordinal loss and original loss,

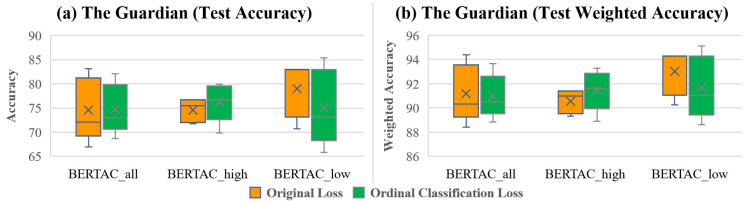

Fig. 4. The Guardian average test accuracy of 5 experiments (a) show the accuracy results and (b) weighted accuracy. The subscript beside BERTAC indicates the experiment type, where *all*= trained on all labeled examples were used, *high*= trained on examples with high agreement score, and *low*= trained on examples with low agreement score.

and BERTAC cased trained ordinal loss. Table 4 shows that the voting system improves the results with respect to accuracy and standard division.

Table 4. Accuracy results in % for BERTAC trained with ordinal loss (BERTAC$_{ord}$) and BERTAC trained with different settings and losses (BERTAC$_{vote}$).

Model	FN	TG	MW	WSJ	DM
BERTAC$_{ord}$	75.08(4.19)	74.78(5.15)	71.08(3.47)	64.45(3.36)	68.42(1.49)
BERTAC$_{vote}$	**76.73 (2.15)**	**76.00 (6.16)**	**74.00 (3.40)**	**64.00 (2.77)**	**69.02 (1.87)**

7 CONCLUSION

In this work, we define the article-comment alignment problem (ACAP) and propose an effective approach to predict the level of relatedness between a comment and an article. We compare Doc2Vec, Siamese LSTM, BA-BC, and BERTAC models and study the performance improvement across them. The results reported in this work show that a joint modeling of article-comments, i.e., BERTAC, is able to capture a deeper level of semantic relatedness between comments and news articles, and help predict better the relevance level of a comment to the content of an article than the current state-of-the-art and other proposed methods.

Even though accuracy values are close, detailed analysis shows that BERTAC trained with proposed ordinal loss perform better than BERTAC on the original BERT loss. With the proposed loss, we can identify common mistakes by annotators and potentially use them to improve the performance of downstream applications, which we will explore in the future.

Acknowledgements

This research was supported in part by the NSF grant IIS-8142183.

References

1. Albert Park, Andrea L. Hartzler, Jina Huh, Gary Hsieh, David W. McDonald, and Wanda Pratt. 2016. "How did we get here?": Topic drift in online health discussions. *Journal of Medical Internet Research*, 18:e284.
2. Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettle-moyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL* 2020, Online, July 5-10, 2020, pages 8440–8451.
3. Ankan Mullick, Sayan Ghosh, Ritam Dutt, Avijit Ghosh, and Abhijnan Chakraborty. 2019. Public sphere 2.0: Targeted commenting in online news media. *In Advances in Information Retrieval*, pages 180–187.
4. Aravindan Mahendiran, Wei Wang, Jaime Arredondo, Bert Huang, Lise Getoor, David Mares, and Naren Ramakrishnan. 2014. Discovering evolving political vocabulary in social media. *International Conference on Behavioral, Economic, and Socio-Cultural Computing, BESC*, pages 1–7.
5. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Conference on Neural Information Processing Systems, NeurIPS*, pages 5998–6008.
6. Carlos Ruiz, David Domingo, Josep Lluís Micó, Javier Díaz-Noci, Koldo Meso, and Pere Masip. 2011. Public sphere 2.0? the democratic qualities of citizen debates in online newspapers. *The International Journal of Press/Politics*, 16(4), pages 463–487.
7. Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen.2020. Bertweet: A pre-trained language model for english tweets. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.*
8. Dyut Kumar Sil, Srinivasan H. Sengamedu, and Chiranjib Bhattacharyya. 2011. Read along: reading articles and comments together. *In Proceedings of the 20th International Conference on World Wide Web,WWW*, pages 125–126.
9. Fabio Celli, Evgeny Stepanov, Massimo Poesio, and Giuseppe Riccardi. 2016. Predicting brexit: Classifying agreement is better than sentiment and pollsters. *In Workshop on PEOPLES*, pages 110–118.
10. Gilad Mishne and Natalie Glance. 2006. Leave a reply: An analysis of weblog comments. *In Third Annual Workshop on the Weblogging Ecosystem.*
11. He, L., Han, C., Mukherjee, A., Obradovic, Z., Dragut, E. (2020). On the dynamics of user engagement in news comment media. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(1), e1342
12. Hershey Friedman and Taiwo Amoo. 1999. Rating the rating scales. *In Journal of Marketing Management*, 9, pages 114–123.
13. Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective API built for detecting toxic comments. *ArXiv Preprint ArXiv:1702.08138.*
14. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *In The North American Chapter of the Association for Computational Linguistics, NAACL*, pages 4171–4186.
15. Jane B. Singer. 2009. Separate spaces: Discourse about the 2007 scottish elections on a national newspaper web site. *In The International Journal of Press/Politics*,14(4), pages 477–496

16. Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 328–339.
17. Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. *In Thirtieth AAAI Conference on Artificial Intelligence*.
18. Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *In Psychological Bulletin*, 76(5), page 378
19. Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. *In Scholarly Commons*.
20. Lei Hou, Juanzi Li, Xiao-Li Li, Jie Tang, and XiaofeiGuo. 2017. Learning to align comments to news topics. *In ACM Transactions on Information Systems*, 36(1), pages 1-31.
21. Mashael Y. Almoqbel, Donghee Yvette Wohn, Rebecca A. Hayes, and Meeyoung Cha. 2019. Understanding facebook news posts comment reading and reacting behavior through political extremism and cultural orientation. *In Computers in Human Behavior*. 100, pages 118-126.
22. Marc Ziegele and Oliver Quiring. 2013. Conceptualizing online discussion value: A multidimensional framework for analyzing user comments on mass-media websites. *In Annals of the International Communication Association*, 37(1), pages 125–153.
23. Marco Bastos and Dan Mercea. 2018. Parametrizing brexit: mapping twitter political space to parliamentary constituencies. *In Information, Communication & Society*, 21(7), pages 921–939.
24. Marija Stanojevic, Jumanah Alshehri, and Zoran Obradovic. 2019. Surveying public opinion using label prediction on social media data. *In IEEE/ACM International Conference on Advances in Social Network Analysis and Mining, ASONAM*, pages 188–195.
25. Marija Stanojevic, Jumanah Alshehri, Eduard Dragut,and Zoran Obradovic. 2019. Biased news data influence on classifying social media posts. *In Workshop on Recent Trends in News Information Retrieval,SIGIR*, volume 2411, pages 3–8.
26. Mrinal Kanti Das, Trapit Bansal, and Chiranjib Bhattacharyya. 2014. Going beyond corr-lda for detecting specific comments on news blogs. *In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14*, pages 483–492
27. Nicholas Diakopoulos and Mor Naaman. 2011. Towards quality discourse in online news comments.*In ACM Conference on Computer Supported Cooperative Work, CSCW*, pages 133–142.
28. Patrick Weber. 2014. Discussions in the comments section: Factors influencing participation and interactivity in online newspapers' reader comments. *New Media & Society*, 16, pages 941–957.
29. Rensis Likert. 1932. A technique for the measurement of attitudes. *In Archives of Psychology*.
30. Sanne Hille and Piet Bakker. 2014. Engaging the social news user.*In Journalism Practice*, 8, pages 563–572.
31. Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional neural networks for toxic comment classification. *In Hellenic Conference on Artificial Intelligence, SETN*, pages 1-6.
32. Swapna Gottipati and Jing Jiang. 2012. Finding thoughtful comments from social media. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 995–1010.

33. Ting Yan and Florian Keusch. 2015. The effects of the direction of rating scales on survey responses in a telephone survey. *In Public Opinion Quarterly*, 79(1), pages 145-165.
34. Toni Gruetze, Ralf Krestel, and Felix Naumann. 2016. Topic shifts in stackoverflow: Ask it like socrates. *In International Conference on Natural Language & Information Systems, NLDB*, pages 213–221
35. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Man-dar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *ArXiv Preprint ArXiv:1907.11692*.
36. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc Le. 2019.Xlnet: Generalized autoregressive pretraining for language understanding. *In Conference on Neural Information Processing Systems, NeurIPS*, pages 5754–5764.
37. Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language modelsbeyond a fixed-length context. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 2978–2988