Mao-Zedong At SemEval-2023 Task 4: Label Represention Multi-Head Attention Model With Contrastive Learning-Enhanced Nearest Neighbor Mechanism For Multi-Label Text Classification

Che Zhang[+*] and **Ping'an Liu**[+] and **Zhenyang Xiao**[*] and **Haojun Fei**[+]
[+]Qifu Technology,China
[*]Peking University
mmt,kjn@stu.pku.edu.cn
liupingan-jk,zhangchulan-jk@360shuke.com

Abstract

The study of human values is essential in both practical and theoretical domains. With the development of computational linguistics, the creation of large-scale datasets has made it possible to automatically recognize human values accurately. SemEval 2023 Task 4(Kiesel et al., 2023) provides a set of arguments and 20 types of human values that are implicitly expressed in each argument. In this paper, we present our team's solution. We use the Roberta(Liu et al.) model to obtain the word vector encoding of the document and propose a multi-head attention mechanism to establish connections between specific labels and semantic components. Furthermore, we use a contrastive learning-enhanced K-nearest neighbor mechanism(Su et al.) to leverage existing instance information for prediction. Our approach achieved an F1 score of 0.533 on the test set and ranked fourth on the leaderboard. we make our code publicly available at https://github.com/peterlau0626/semeval2023-task4-HumanValue.

1 Introduction

The identification and analysis of human values in texts has been an important area of research. With the development of computational linguistics, this research has gained widespread attention because of its potential impact on areas such as sentiment analysis, social science.

One of the challenges in this area is to accurately categorize all the human value. Several notable research achievements have been made in the categorization of human values. One of the approach is that classifies human values into 54 categories across four different levels(Kiesel et al., 2022). SemEval2023 task4 uses the classification method in this paper, where an argument is given to identify whether a value is included in the instrument, and the F1 scores of the results at the level2 level are used for the total ranking. There are 20 categories of human values in level 2, and a argument could belong to multiple value categories or not to any one value category. this is a typical multi-label text classification (MLTC) problem which has been applied in many scenarios such as news emotion analysis(Bhowmick et al.) and web page tagging(Jain et al.).

In this paper, we propose a model that combines the label-specific attention network with the contrastive learning-enhanced nearest neighbor mechanism(Su et al.). The multi-headed attention mechanism allows our model to overcome the shortcomings of traditional attention mechanism models and to be able to focus on different parts of a document, resulting in more accurate labeled attention results. And the nearest neighbor mechanism enables our model to not waste the rich knowledge that can be directly obtained from the existing training instances and helps enhance the interpretability and robustness of the model.

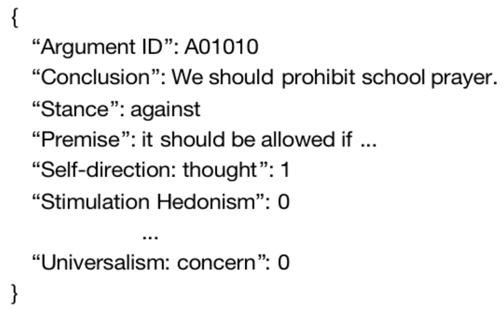
2 Background

2.1 Datasets

The dataset comprises of arguments from six different domains such as news releases, online platforms, etc. originating from four different countries/regions, which are composed of 80% data from IBM argument quality dataset (95% from the original dataset), 15% from the European Future Conference (New), and 5% from group discussion ideas (2% from the original dataset). The training dataset comprises of more than 6500 arguments, whereas the validation and test datasets consist of around 1500 arguments each. In addition, the organizers of the competition provided three additional datasets to evaluate the robustness of methods:validation set Zhihu (labels available), test set Nahj al-Balagha (labels confidential), test set The New York Times (labels confidential). All datasets have been manually annotated.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 426–432
July 13-14, 2023 ©2023 Association for Computational Linguistics

Each sample in the dataset contains an argument ID, conclusion, stance towards the premise, and the premise itself. The labels consist of the argument ID and a column for each of the 20 value categories, indicating whether the sample belongs to each category (0 or 1).



```
{
  "Argument ID": A01010
  "Conclusion": We should prohibit school prayer.
  "Stance": against
  "Premise": it should be allowed if ...
  "Self-direction: thought": 1
  "Stimulation Hedonism": 0
              ...
  "Universalism: concern": 0
}
```

2.2 Related Work

Before the widespread adoption of deep learning, models such as SVM were widely used to minimize an upper bound of the generalization error(Qin and Wang). Simple neural network (NN) models were later used for MLTC and achieved good performance(Nam et al.). Additionally, convolutional neural networks (CNNs) and recurrent networks with gated recurrent units (GRUs) have been successfully used with pre-trained word2vec embeddings(Berger). Feature selection has been shown to be effective in speeding up learning and improving performance by identifying representative words and removing unimportant ones(Spolaor and Tsoumakas).

In recent years, with the development of pre-trained models, the ability to extract semantic information has become increasingly powerful. there have been several representative works that have focused on improving the MLTC models. For example, (Pal et al.) utilized graph neural networks based on label graphs to explicitly extract label-specific semantic components from documents. seq2seq model can capture the correlation between tags(Yang et al.). LSAN(Xiao et al.) can focus on different tokens when predicting each label.

3 System Overview

In this section, we will present our model, which consists of two main parts. The first part is a multi-headed attention mechanism based on a specific label representation, while the second part is a nearest neighbor mechanism enhanced using contrast learning.

The MLTC problem can be described as follows: assuming a set of data $D = \{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i=1}}^{\mathrm{N}}$, N labeled documents, where x_i represents the text and $y_i \in \{0, 1\}^l$ represents the label of x_i, and l represents the total number of labels. Each document x_i consists of a series of words. Our goal is to learn a classifier to establish a mapping from x_i to y_i, so that when a new document x is presented, its label vector y can be correctly predicted.As pre-trained language models (PLMs) show remarkable performance in extracting natural language representations, we use PLMs as base encoder to get document and label feature. A input sample can be expressed as $\mathrm{x}_i = \{w_1, w_2, w_3 \ldots w_{n-1}, w_n\}$, $\mathrm{w_p} \in R^{\mathrm{d}}$ denotes the pth word vector of a document. After calculated by PLMs , the input matrix of the whole sentence is obtained as $H \in R^{\mathrm{n} \times d}$, where d is the hidden dimension of PLMs.

3.1 Label-specific multi-head attention network

In order to explicitly capture the corresponding label-related semantic components from each document, the approach of using label-guided attention mechanisms to learn label-specific text representations has been widely used in previous studies, and such a method is used in LSAN(Xiao et al.). In addition, the success of the Transformer model(Vaswani et al.) illustrates the ability of multi-headed attention mechanisms to extend the model's ability to focus on different locations more effectively than single-headed attention mechanisms. The usefulness of this method for text classification is very intuitive.For example in the following sentence, "Social media is good for us. Although it may make some people rude, social media makes our lives easier." Focusing on the words “although”, “but” and “makes life easier” at the same time is a more accurate way of getting at the value of comfort in life, while ignoring the the disadvantages of social media.As mentioned above, we next show our model.

Firstly, to make use of the semantic information of labels, we initialize the trainable label representation matrix $C \in R^{\mathrm{l} \times d}$ with the mean-pooling of the label features vector which is obtained by the pretrained encoder. Then, the multi-headed attention mechanism is used to compute the label-aware attention score. With the input document representation matrix as $H \in R^{\mathrm{n} \times d}$ and the label representation matrix C, the query Q, key K, and value

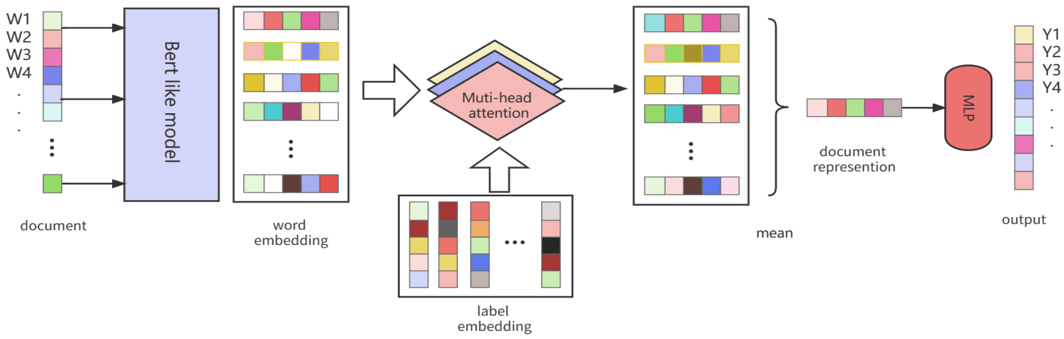


Figure 1: The architecture of the proposed label-specific multi-attention network model

V of the attention mechanism can be expressed as follows:

$$\begin{aligned} Q &= W_\mathrm{q} C \\ K &= W_\mathrm{k} H \\ V &= W_v H \end{aligned} \tag{1}$$

Where $W_Q, W_K, W_V \in R^{d \times d}$ is the weight matrix to be learned.We use the h-head attention mechanism, then the three matrices Q,K,V can be expressed in the following form.

$$\begin{aligned} (Q_1, Q_2, \ldots Q_h) &= Q \\ (K_1, K_2, \ldots K_h) &= K \\ (V_1, V_2, \ldots V_h) &= V \end{aligned} \tag{2}$$

Where $Q_\mathrm{i} \in R^{l \times d_\mathrm{a}}, \quad K_\mathrm{i}, V_\mathrm{i} \in R^{\mathrm{n} \times d_\mathrm{a}}$ correspond to the query, key and value of each attention header, and $d_\mathrm{a} = d/h$ denotes the dimensionality of a single attention mechanism representation space. Attention scores are then computed for each attention head similar to the method used in the Transformer model. Since the length of the document is different in a data batch, we perform a mask operation on the result of the QK matrix multiplication, set the value corresponding to the padding part to $1e^{-12}$, and then use the softmax activation function to activate it.

$$\mathrm{score}_i = \mathrm{softmax}\left(\mathrm{mask}\left(\frac{Q_i K_i}{d_a}\right)\right) \tag{3}$$

$\mathrm{score}_i \in R^{1 \times n}$ denotes the attention score of the label for each word vector in the document. Then we obtain the attention results for each label with respect to the document content.

$$\begin{gathered} Attention = \\ \mathrm{Concat}\left(\mathrm{attention}_1, \ldots, \mathrm{attention}_\mathrm{h}\right) W^O \\ \\ \text{where } \mathrm{attention}_\mathrm{i} = \mathrm{score}_\mathrm{i} \, \mathrm{V}_\mathrm{i} \end{gathered} \tag{4}$$

Where $Attention \in R^{l \times d}$ can be considered as the representation vector of the document under the view of L labels. To obtain the representation vector of the document Z, the row vectors of the Attention matrix for each labeled view are averaged:

$$Z = \mathrm{mean}(\mathrm{row}(\mathrm{Attention})) \tag{5}$$

After obtaining a comprehensive document representation with label-specific correlation, we can construct a multi-label text classifier by means of a perceptron consisting of fully connected layers. Mathematically, the predicted probability of each label of the next document can be determined by: $\hat{\mathrm{y}} = \mathrm{sigmod}\left(W^1 Z^T\right)$. Where $W^1 \in R^{l \times d}$is trainable parameters for the fully connected and output layers, which can transfer the output value into a probability. Since multi-label classification has the problem of unbalanced positive and negative samples, in order to balance the coefficients of positive and negative samples in the loss function and obtain a better trained model, we use the cross-entropy loss function with weights as the loss function of the model:

$$\begin{aligned} L_{BCE} = \sum_{i=1}^{b} \sum_{\mathrm{j}=1}^{l} -(\mathrm{w} \cdot \mathrm{y_{ij}} \log\left(p_{ij}\right) \\ + \left(1 - y_{ij}\right) \log\left(1 - p_{ij}\right)) \end{aligned} \tag{6}$$

Where b is the size of a data batch, w is the weighting factor, $\mathrm{y_{ij}}$ is the true value of the jth label of the ith sample, $\mathrm{p_{ij}}$ is the probability that the model predicts that label to be $\mathrm{y_{ij}}$, and l is the total number of labels. The positive sample size/negative sample size in the training set is taken as the value of w.

3.2 Contrastive Learning-Enhanced Nearest Neighbor Mechanism

We use the k nearest neighbor (KNN) mechanism enhanced by contrast learning(Su et al.). This approach innovatively proposes a KNN mechanism for multi-label text classification that can make good use of the information of existing instances. And a contrastive learning approach is designed to enhance this KNN mechanism mechanism effectively. Specifically, this approach designs a loss function for contrastive learning based on dynamic coefficients of label similarity, which compares the documents representation vectors at training time to let the vector with more same labels be more similar as possible, while the vectors of documents with fewer identical labels are as far away as possible. Assuming a data batch of size b, we define a function to output all other instances of a particular instance in this batch $\mathrm{g(i)} = \{\mathrm{k} \mid \mathrm{k} \in \{1,2,\ldots,\mathrm{b}\}, \mathrm{k} \neq \mathrm{i}\}$.The contrastive loss(CL loss) of each instance pair (i, j) can be calculated as:

$$\begin{aligned} \mathrm{L}^{\mathrm{ij}}_{\mathrm{con}} &= -\beta_{ij} \log \frac{e^{-d(z_i,z_j)/\tau'}}{\sum_{k\in g(i)} e^{-d(z_i,z_k)/\tau'}} \\ C_{\mathrm{ij}} &= y_i^T \cdot y_j, \quad \beta_{ij} = \frac{C_{ij}}{\sum_{k\in g(i)} C_{ik}} \end{aligned} \tag{7}$$

where $d(\cdot,\cdot)$ is the euclidean distance, τ' is the contrastive learning temperature and Z denotes the document representation. C_{ij}denotes the label similarity between i, j, and normalized to obtain β_{ij}. The CL loss of the whole batch can be expressed as:$L_{\mathrm{con}} = \sum_i \sum_{j\in g(i)} L^{ij}_{con}$. The cross-entropy loss function is expressed as L_{BCE}, then the whole Loss function is $\mathrm{L} = L_{BCE} + \gamma L_{\mathrm{con}}$, where a controls the ratio of the coefficients of the contrast learning loss function and the cross-entropy loss function. Then, we construct a data store of training instances so that we can later use the existing instance information as a comparison. Based on the training set $(xi, yi) \in D$, the storage of a training set of document representation vectors$D' = \{(\mathrm{~hi, yi~})\}_{i=1}^{N}$ is obtained by a trained model. where h_{i}denotes the document representation vector of the training set, which is calculated by the model.

In the inference stage, give an input X, after the model calculation we obtain its document representation vector Z, and the prediction of the model $\hat{y}_M \in \{\mathrm{p} \mid \mathrm{p} \in [0,1]\}$[1]. Next, we compare with the data repository to find the nearest k nearest neighbors $\mathrm{N} = \{(\mathrm{hi, yi})\}_{i=1}^{\mathrm{k}}$, then the KNN prediction can be calculated as:

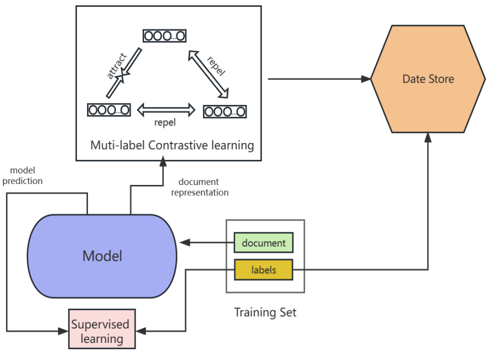


Figure 2: The training process of the wholel model with contrastive learning

$$\begin{aligned} \hat{\mathrm{y}}_{KNN} &= \sum_{i=1}^{k} \alpha_i y_i \\ \alpha_i &= \frac{e^{-d(h_i,Z)}/\tau}{\sum_j e^{-d(hj,Z)}/\tau} \end{aligned} \tag{8}$$

where $d(\cdot,\cdot)$ is the euclidean distance, τ is the temperature of KNN, α_i is the weight coefficient of the ith neighbor, when the closer the test instance vector representation is to this neighbor, the larger the weight will be. The final prediction form is expressed as follows:

$$\hat{\mathrm{y}} = \lambda \hat{\mathrm{y}}_{KNN} + (1-\lambda)\hat{\mathrm{y}}_M \tag{9}$$

where λ is the weight coefficient that regulates the KNN prediction and the model prediction.

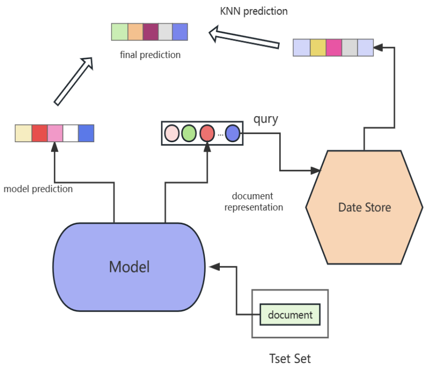


Figure 3: The Overall prediction process with KNN mechanism

4 Experimental Setup

In the dataset, many pronouns in the premise were used such as "this research", "it", "this way" etc, and these pronouns refer to objects contained in the conclusion. Whereas our model tries to establish the attention scores of different semantic components of a document for a specific label, it is clear that the presence of these words with unclear denotations affects the attention results. In addition, stance toward conclusions also influence value judgments. Therefore, we use a simple strategy: combining the three parts of conclusion, preference, and stance into a sentence that conforms to natural language conventions. Specifically, the data were preprocessed uniformly and simply, and the input structure was: "I agree (disagree) that" + conclusion content + ", because" + premise content.

We use the Roberta model(Liu et al.) as the base pre-trained model to obtain word representation. Experimenting with our architecture on the base model. And we use the K-fold cross-validation method. We merge the training and validation sets and then randomly divide them into six copies. We perform the training process six times, each time using 5/6 of the data as the training set and 1/6 of the data as the validation. During the training process, the best-trained model for each fold is saved, and the average output probability of all models is taken as the final prediction score.

5 Results

On the leaderboard of the TIRA, our method achieved an macro-F1 score of 0.53 and ranks fourth, while the baseline which use a bert model(Devlin et al.) achieved 0.42, with the best result for the whole competition being 0.56. In addition, our model also achieved an F1-score of 0.32 on each of the two test sets, Nahj al-Balagha and New York Times. This effect is relatively high among all participating teams, which fully demonstrates the robustness and stability of our approach.[1]

To illustrate the effectiveness of our architecture, we conducted ablation experiments. The ablation experiments evaluate the performance effect of the model directly on the validation set merged with the dataset from Zhihu. In the ablation experiments, we did not use the strategy of k-fold cross-validation. The results of the ablation experiment are shown in Table 1, which shows all strategy results with Precision, Recall, and marco-F1. We show the results for each method using the average results from three runs. At first, we use the word vector corresponding to the output of the Roberta model [CLS] as the representation vector of the document to connect the classifier as the baseline. We then compared the effect of baseline with LSAN(Xiao et al.), just use multi-attention mechanism, and the effect of removing the multi-headed attention mechanism part. As can be seen in the table, after using the multi-headed attention mechanism, the marco-F1 value improves by about 0.3% compared to the baseline model, while the LSAN mechanism get 0.2% improvement on F1-score. And after adding the KNN mechanism augmented with contrastive learning alone, the marco-F1-score is improved by about 0.4%. In the case of the full model, the marco-F1-score improves by about 0.7% compared to the baseline. This result is within our expectation and illustrates the effectiveness of our method. Then in order to increase the stability and robustness, and to avoid overfitting generation, we use the K-fold cross-validation method, so that our experimental results can be shown relatively stable, which leads to an improvement of about 0.4 percentage points in the F1-scores.

model/score	**precise**	**recall**	**F1**
baseline	0.474	0.572	0.518
multi-attention	0.475	0.579	0.522
LSAN	0.472	0.580	0.520
baseline+KNN	0.462	0.603	0.523
multi-attention+KNN	0.482	0.577	0.525

Table 1: Ablation Experiment Results

6 Conclusion

We propose a multi-label text classification model using a label-specific multi-headed attention mechanism. Compared to previous models of attention mechanisms, the use of multi-headed attention enables specific labels to focus on different semantic components of the document more effectively. Besides, we use the KNN mechanism to exploit the instance information in the training set. We then perform ablation experiments on our architecture to analyze the role of each part and demonstrate the superiority of using a multi-headed attention mechanism.

[1] Find it from Appendix

References

Mark J Berger. Large scale multi-label text classification with semantic word vectors.

Plaban Kr. Bhowmick, Anupam Basu, Pabitra Mitra, and Abhishek Prasad. Multi-label text classification approach for sentence level news emotion analysis. In *Pattern Recognition and Machine Intelligence*, Lecture Notes in Computer Science, pages 261–266. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding.

Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme multi-label loss functions for recommendation, tagging, ranking &amp; other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 935–944. Association for Computing Machinery.

Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. Identifying the Human Values behind Arguments. In *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 4459–4471. Association for Computational Linguistics.

Johannes Kiesel, Milad Alshomary, Nailia Mirzakhmedova, Maximilian Heinrich, Nicolas Handke, Henning Wachsmuth, and Benno Stein. 2023. Semeval-2023 task 4: Identification of human values behind arguments. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification — revisiting neural networks. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 437–452. Springer.

Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. Multi-label text classification using attention-based graph neural network. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*, pages 494–505.

Yu-ping Qin and Xiu-kun Wang. Study on multi-label text classification based on SVM. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 1, pages 300–304.

Newton Spolaor and Grigorios Tsoumakas. Evaluating feature selection methods for multi-label text classification.

Xi'ao Su, Ran Wang, and Xinyu Dai. Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 672–679. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 466–475. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. SGM: Sequence generation model for multi-label classification.

A Appendix

Test set / Approach	All	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Face	Security: personal	Security: societal	Tradition	Conformity: rules	Conformity: interpersonal	Humility	Benevolence: caring	Benevolence: dependability	Universalism: concern	Universalism: nature	Universalism: tolerance	Universalism: objectivity
Main																					
Best per category	.59	.61	.71	.39	.39	.66	.50	.57	.39	.80	.68	.65	.61	.69	.39	.60	.43	.78	.87	.46	.58
Best approach	.56	.57	.71	.32	.25	.66	.47	.53	.38	.76	.64	.63	.60	.65	.32	.57	.43	.73	.82	.46	.52
BERT	.42	.44	.55	.05	.20	.56	.29	.44	.13	.74	.59	.43	.47	.23	.07	.46	.14	.67	.71	.32	.33
1-Baseline	.26	.17	.40	.09	.03	.41	.13	.12	.12	.51	.40	.19	.31	.07	.09	.35	.19	.54	.17	.22	.46
our model	.53	.53	.70	.26	.29	.60	.45	.54	.31	.77	.65	.58	.60	.51	.16	.59	.42	.73	.85	.43	.55
Nahj al-Balagha																					
Best per category	.48	.18	.49	.50	.67	.66	.29	.33	.62	.51	.37	.55	.36	.27	.33	.41	.38	.33	.67	.20	.44
Best approach	.40	.13	.49	.40	.50	.65	.25	.00	.58	.50	.30	.51	.28	.24	.29	.33	.38	.26	.67	.00	.36
BERT	.28	.14	.09	.00	.67	.41	.00	.00	.28	.28	.23	.38	.18	.15	.17	.35	.22	.21	.00	.20	.35
1-Baseline	.13	.04	.09	.01	.03	.41	.04	.03	.23	.38	.06	.18	.13	.06	.13	.17	.12	.12	.01	.04	.14
our model	.32	.06	.39	.31	.44	.66	.10	.33	.59	.41	.16	.45	.24	.16	.31	.35	.20	.25	.25	.00	.28
New York Times																					
Best per category	.50	.50	.22	.00	.03	.54	.40	.00	.50	.59	.52	.22	.33	1.00	.57	.33	.40	.62	1.00	.03	.46
Best approach	.34	.22	.22	.00	.00	.48	.40	.00	.00	.53	.44	.00	.18	1.00	.20	.12	.29	.55	.33	.00	.36
BERT	.24	.00	.00	.00	.00	.29	.00	.00	.00	.53	.43	.00	.00	.00	.57	.26	.27	.36	.50	.00	.32
1-Baseline	.15	.05	.03	.00	.03	.28	.03	.00	.05	.51	.20	.00	.07	.03	.12	.12	.26	.24	.03	.03	.33
our model	.32	.22	.12	.00	.00	.47	.29	.00	.22	.53	.41	.00	.32	.50	.15	.21	.40	.56	.33	.00	.38

Table 2: Achieved macro-F1-score of our team per test dataset, for each of the 20 value categories.