INF554 Report

Extractive Summarization in Dialogue with Graph Convolutional Neural Network

Xianjin GONG[1] (Kaggle Username: xavierscor)
Chenwei WAN[1] (Kaggle Username: chenweiwan)
Mengfei GAO[1] (Kaggle Username: mengfeigao)
[1]firstname.lastname@polytechnique.edu

Kaggle Teamname: GWG

École Polytechnique
IP Paris
France
07/12/2023

1 Introduction

The extractive summarization task involves selecting key components from a given document to create a condensed summary. While in dialogue settings, such task becomes challenging due to the complex dependency relations between utterances and the frequent shift of speaker turns.

In this report, we present a graph neural network based approach to leverage intricate discourse structures. Following the best practice, we utilize RoBERTa, a pretrained large language model for textual feature extraction. We also designed a multi-task training strategy to better differentiate utterances with similar semantics but distinct contexts.

Our method achieves competitive F1 score (0.6148) on the test set, demonstrating the effectiveness of our design. We also empirically study the trade-offs of certain design choices, such as speaker embedding and additional GCN modules incorporating speaker-aware relations.

2 Feature Extraction

2.1 Dialogue Visualization

In order to gain a more intuitive understanding of the data, we first built a program with an open-source framework called "Streamlit" [2] to plot the dialogue graph as shown in Figure 1. Important and non-important nodes are represented by star and circle tokens respectively, with numbers indicating the ID in the dialogue. Colors of links represent the relation between two utterances. The system is interactive such that users can specify any node from any dialogue as the root, and by giving the wanted depth, it will plot the dialogue graph by breadth-first search.

With this program, it is both easy to observe linking patterns of the whole dialogue before choosing proper network architecture, and easy to study local neighboring nodes when some utterance is wrongly predicted.

2.2 Utterance Information

The first feature we would like to extract is the meaning of each utterance.

Utterances can be decomposed into smaller units, English words, but they are not a good starting point here. Because the dialogue is produced by role-playing conversations, there are a lot of informal words such as modal particles ("Mmm", "Uh") and spoken reductions ("n" in "we're n using kinetic"). It is better to decompose English words further into sub-words [5], which we can utilise to guess the meaning of a word if it does not occur in our dictionary.

Further more, the meaning of a word (in following report, a word can mean both an English word or a subword) can change based on its context. When calculating the meaning of an utterance, we should calculate the meaning of a word based on its context instead of using a direct linear combination, which means we need attention-based models.

2.3 Context Information of Utterances

In the last section, we have taken care of the context information of each word. The context information in this section specifically refers to the context of each utterance within its corresponding dialogue. This context is based on an utterance's neighborhood, and the relation between them.

For example, utterances from 653 to 658 in Figure 1 repeat the information "curved" four times. But only the first utterance is considered as important. We think one potential reason is that 653 is the first utterance which brings up a new concept ("curved") and it is the head of an explanation relation. If a new concept requires a second utterance to explain further in a dialogue, it might be important. The following utterances 654 and 656 are only repetitions of 653, which reduces their importance.

Therefore, context information of utterances is crucial for determining importance of an utterance and it is the second feature we would like to extract from the data set.

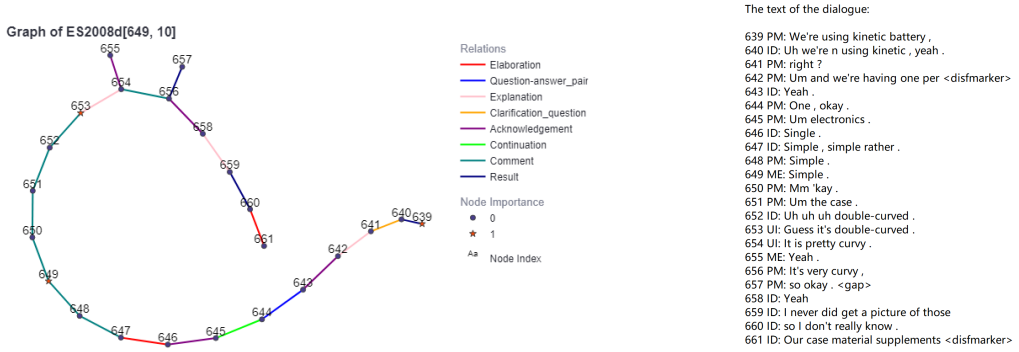
2.4 Speaker Information

Speaker type is a trivial information in the data set but we left it as the last feature to extract because there are two different ways to embed it.

One assumption is that the type of speaker will only have local influence. For example, if an utterance is describing an user interface concept, and it is delivered by the interface designer (UI), it might be considered as more important than when it is delivered by the marketing expert (ME).

Another assumption is that the speaker type of an utterance will influence the importance of its neighbors. If we take a look at utterances 646 to 649, we can observe that the concept "simple" is repeated by 3 speakers. 646 by the industrial designer, as the first person who brings up the concept and elaborates on the concept with two utterances, is not considered as important. On the contrary, 649 is considered as important. We think it is because, within the context, the comment by marketing expert is more important than other speakers.

We tested both assumptions, but we did not find an optimal one. Further analysis will be shown in the following sections.



(a) ES2008d Dialogue Graph from Utterance 649 (b) Text

Figure 1: Dialogue visualization with its original text data. a) Dialogue graph plotted starting from utterance 649. The graph is plotted with breadth-first search without considering the direction of the relation. The largest depth is set as 10 in this graph. b) The original text of the dialogue part plotted.

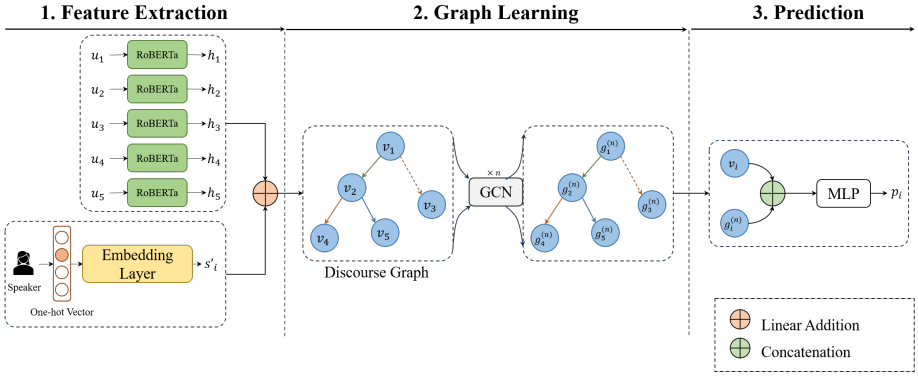


Figure 2: The neural network structure implemented.

3 Model Choice, Tuning and Comparison

3.1 F1 Loss Function

We decided to discuss loss function before the details of model because this task naturally suffers from the fact that the data set is highly unbalanced. This is because the summary of a dialogue should contain much less words than the original text, which means the number of important utterances is much smaller than the number of non-important ones.

If regular loss function for binary classification such as the Binary Cross Entropy (BCE) is used, the model will fall into the local minimum by predicting all utterance as non-important. Traditional F1 loss can capture the prediction ability of model properly but it is a discrete function, which can not be used to guide gradient descent. One solution is that we replace the binary prediction by a continuous prediction from 0 to 1 (which can be interpreted as the probability of being important).

Further improvement can be made by passing the prediction through a scaling function before inputting the "Macro F1":

$$pred' = \frac{(2 \times (pred - 0.5))^s}{2} + 0.5$$

where s is an odd hyper parameter.

This function will penalty on predictions close to 0.5. And the higher s is, the higher the penalty is. The aim is to reduce ambient predictions which easily jump from important to non-important by a slight perturbation on the weights of network because they are to close to 0.5.

By changing from BCE loss to "Macro F1", our 5-fold cross validation discrete F1 score increased from 0.58 to 0.612.

3.2 Utterance Encoding

We define each utterance as a sequence of words:

$$u_i = \{w_1^i, w_2^i, ..., w_{|u_i|}^i\}$$

We fine-tuned two models, "RoBERTa" [3] and "RoBERTa Large" [3] by making context-free prediction on each utterance individually. Specially, for each utterance, we prepend a "[CLS]" token to the sequence of words. Subsequently, we extract the output activation from the last layer corresponding to the "[CLS]" token, which serves as the feature representation $h_i \in \mathbb{R}^d$ of u_i, where d is the dimension of utterance embedding. The representation of all the utterances can be represented as $H^u \in \mathbb{R}^{|D| \times d}$.

They are fed into an MLP clasifier composed of a linear layer, a GELU layer, and a sigmoid layer to produce the context-free prediction for fine-tune. "RoBERTa Large" performs better with an F1 score = 0.608 over the "RoBERTa" with an F1 score = 0.604, which fits our expectation because "RoBERTa large" has more parameters and a higher embedding dimension = 1024.

3.3 Utterance Context Embedding by Graph Learning

Because our graph is used to represent a dialogue and for different pairs of utterances we have different relations, our model is inspired by the DialogueGCN [1].

Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, with nodes $v_i \in \mathcal{V}$, labeled edges $r_{ij} \in \mathcal{E}$ with $r \in \mathcal{R}$ as the relation type between two connected utterance nodes v_i and v_j.

RGCN Conv:

$$g_i^{(1)} = \sigma(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(1)} v_j + W_0^{(1)} v_i)$$

$$\text{for } i = 1, 2, ..., |D|$$

where N_i^r denotes the neighboring indices of utterance i under relation r and $c_{i,r}$ is a problem specific normalization constant. σ is an activation function, $W_r^{(1)}$ and $W_0^{(1)}$are learnable parameters of the transformation.

$$g_i^{(2)} = \sigma(\sum_{j \in N_i} W^{(2)} g_j^{(1)} + W_0^{(2)} g_i^{(1)})$$

$$\text{for } i = 1, 2, ..., |D|$$

Again, an MLP layer is used for the importance prediction:

$$h_i' = v_i || g_i^{(2)}$$

$$\hat{y}_i = MLP(h_i')$$

where || denotes concatenation.

We also implemented contrastive loss [4] to further aggregate embeddings with same importance:

$$\mathcal{L}_{contrastive} = -\frac{1}{|P|} \sum_{(p,i) \in P} \log \frac{\exp(h_p' \cdot h_i'/\tau)}{\sum_{j=0}^{K} \exp(h_p' \cdot h_j'/\tau)}$$

where $(\cdot)$ denotes cosine similarity, P denotes the set of all the possible tuples (p, i) that $y_p = y_i$, and K is the set of all the possible indices that h_j' can form a negative pair with h_p'.

The complete loss function can be written as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{f1} + \lambda \mathcal{L}_{contrastive}$$

where λ is a hyper parameter to balance the weights of loss components.

We used following hyper parameters in our model to produce highest F1 score on 5-fold cross validation:

- Learning rate: 2e-6
- Batch size: 64
- Epoch: 8
- Utterance embedding dimension: 1024
- Dropout: 0.5
- L2 regularization: 1e-3

with F1 score = 0.614.

3.4 Speaker Types

We implemented two structures to embed speaker types. First we used an embedding layer to produce speaker types vectors which has the same dimension as the utterance encoding. Then we linearly added the speaker types vectors to the utterance encoding before inputting into GCNs:

$$s_i' = W_s s_i + b_i^s$$

$$v_i = h_i + s_i'$$

where $W_s \in \mathbb{R}^{|Speakers| \times d}$.

In another structure, for each utterance, we defined a new relation by the speaker types of the two. Because relations are directed in the graph, this will produce 4x4 = 16 new relations types.

Inspired by DualGATs [6], a separate parallel GCN is used to embed the speaker relations. The output of two parallel GCNs are concatenated before final MLP layers. Unfortunately, this structure produces an F1 score = 0.6128 on the test data set, which is lower than the structure shown in Figure 2 with F1 score = 0.6148.

Due to page limitation, we decided to not to elaborate more on that structure in this report.

References

[1] Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation, 2019.

[2] Mohammad Khorasani, Mohamed Abdou, and Javier Hernández Fernández. *Streamlit Basics*, pages 31–62. Apress, Berkeley, CA, 2022.

[3] 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, 2019.

[4] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

[6] Duzhen Zhang, Feilong Chen, and Xiuyi Chen. DualGATs: Dual graph attention networks for emotion recognition in conversations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7395–7408, Toronto, Canada, July 2023. Association for Computational Linguistics.