RAPPORT D'INF554 - GIANT STEPS

Extractive Summarization with Discourse Graphs

8 décembre 2023

Ziad Oumzil
Xiaoxian Yang
João Pedro Sedeu Sedeu GODOI

SUMMARY

1 DATA PROCESSING

1.1 DATASET DESCRIPTION

The DATASET provided involves 97 and 40 dialogues for training and test respectively.

The dialogues are discussions (meeting) between a project manager (PM), a marketing expert (ME), a user interface designer (UI), and a industrial designer (ID).

For each dialogue, we have access to :

1. **Transcription** : a sequence of utterances.
2. **Discourse graph** : a graph in which the nodes represent utterances and the edges represent discourse relation.

1.2 TEXT EMBEDDING

The first step is to transform the text data into a numerical format, specifically a vector of numbers. The baseline code provided in Moodle uses the **BERT** transformer (Bidirectional Encoder Representations from Transformers).

We asked ourselves the following question : should we use embedding by phrase or embedding by word ? The baseline used embedding by sentences ; we primarily chose this approach because it is simpler. We also tried word embedding ; however, this method consumed a lot of time and memory space without yielding any significant improvement. Therefore, we decided to move on and stick to sentence embedding.

[19]:
```
bert = SentenceTransformer('all-MiniLM-L6-v2')
sentences = ["I love dogs", "I love cats", "Spread love everywhere you go"]
text_embedding = bert.encode(texts, show_progress_bar = False)

print("The size of the sentence encoding ", text_embedding[0].shape)
print("Distance between the two first sentences : ", np.linalg.norm(text_embedding[0] - text_embedding[1]))
print("Distance between the two last sentences : ", np.linalg.norm(text_embedding[1] - text_embedding[2]))
```

```
The size of the sentence encoding  (384,)
Distance between the two first sentences :  0.6751319
Distance between the two last sentences :  1.2267762
```

FIGURE 1 – BERT correctly identifies the high similarity between the first two sentences.

1.3 BERT MODEL :

BERT (Bidirectional Encoder Representations from Transformersis) a pretrained model that encodes a text into a 384-dimensional vector of floating-point numbers. This embedding captures semantic information about the input sentence.

Initially, we utilized the BERT model for embedding. However, as we explored ways to enhance our predictions, we considered experimenting with a different embedding model, possibly opting for a larger one.

1.4 DEBERTA MODEL :

DeBERTa (Decoding-enhanced BERT with disentangled attention) is indeed described as an enhanced version of BERT and RoBERTa models. Hugging Face characterizes DeBERTa as a model that incorporates decoding enhancements and disentangled attention mechanisms.

One issue we encountered when working with DeBERTa is that it provides individual embeddings for each word in a sentence. To obtain a single vector representation for the entire sentence, we opted to compute the mean of the individual word vectors. This averaging approach allows us to condense the information from the individual word embeddings into a more manageable and compact representation, making it suitable for various downstream tasks such as classification or similarity analysis. (We also tried other models of pooling.)

1.5 RELATIONS :

The dataset annotates conversation pairs with 16 different relations. Our analysis indicated varying importance levels among these relations, guiding our focus on effectively leveraging them. (which we will elaborate on in the next subsection).

1.6 GRAPH :

Inspired by the data structure of the received .txt file (directed graphical structure), we naturally think of the GCN models that can be used to deal with this type of problem, and with the addition of edges representing relations, we lead to RGCN (Relational Graph Convolutional Network) and RGAT (Relational Graph Attention Network), which are good machine learning models for solving graphs with relations.

The key here is how to deal with relations, first we have 16 relations written directly in the dataset, and in addition we intuitively consider self-loop relations as well as reverse relations. We think that for the same relation, their inverse relation is also some kind of the same relation. This gives us a total

of 33 relations, and RGCN and RGAT happen to provide the parameter for us to distinguish between the different relations.

2 MODELS & TRAINING

Our final model, RGAT, exhibited superior performance. It consists of a 5-layer architecture : two RGAT layers interspersed with ReLU activations, followed by a linear layer for binary classification. The cross-entropy loss function was used, incorporating the hyperparameter alpha to balance the skewness in label distribution.

Overfitting Challenges : Overfitting is a very serious problem and occurs frequently during training. This is manifested by performing well on the training set, but on the local validation the f1 score decreases as the training progresses. To solve it,

1. we recorded the parameters of the model with the best f1 score and set the number of trials to continue to 50
2. we set the dimensions of the output layer of the two RGAT layers to the hyperparameters as well to minimize overfitting by manually tuning the parameter
3. ,we utilized the max voting technique at the end by creating multiple submissions to vote on a final submission.

Comparison with other models :

1. Decision Tree or RandomForest (f1-score : 0.34 - 0.45) :
   (a) Nature of Models : These models are inherently non-sequential and non-relational, meaning they treat each data point (in this case, each sentence) independently.
   (b) Limitations : They fail to capture the sequential nature of dialogues, where the context and flow of conversation are crucial. The inability to effectively utilize relational information between sentences (e.g., discourse relations) leads to a significant loss of contextual information.
2. RGCN (f1-score :0.599 really close to RGAT) :
   Relational Context : Unlike Decision Trees and RandomForest, RGCN is capable of incorporating relational information between the utterances, thanks to its graph-based nature.