INLP Project

Final Submission Report

Abhijeeth Singam [2019101065]
Tushar Jain [2019101091]
Rishabh Khanna [2019113025]

Work Done

Link to models: NLP Proj Models

Problem Statement -

We scoped down our problem statement to focus on one of two tasks in the shared tasks. The two proposed tasks were:

- Main + Others: In this task the model was only required to classify the referenced character if they were one of the 6 main characters of the show and all other characters could be classified as "other". This makes the task simpler as there is no need to have a large number of classes but wouldn't be able to handle classes with smaller frequencies which can at times be very important in a real world scenario.
- All: In this task the model is required to classify between every character in the show, and not just the main 6 characters. In order to simplify the task a little bit the entities with very few references are also clubbed into an 'other' class. While this also may not be perfect, it is a better fit for the real world. This task ends up having a much large set of classes, resulting in a unique approach to solving it.

In general models that perform well on the first task tend to have a decent accuracy on the second as well, but their f1 falls considerably which is indicative of the drastic class imbalance in the dataset.

Thus, we chose to focus on the second task ('all') as it seemed more interesting in it's solutions and scope for exploration, as well as more true to the intent of the task.

Baseline:

The baseline we are building is the AMORE-UPF system which was the winning solution for the 'all' task. It makes use of an entity library and cosine similarity alongside a relatively simple Bi-LSTM model to handle the vast number of entities effectively. The results show that even with relatively little model complexity and sophistication, the addition of an entity library was able to contribute greatly to the performance and beat out other, more complex, architectures.

We hope to build upon this baseline by modifying the neural model and trying to incorporate attention-based mechanisms.

The scheduled time to complete the baseline was 20th March and we are on track to do so.

Below we describe the preprocessing done on the data and a brief description of the model architecture of the AMORE-UPF system.

Preprocessing the data -

- First, we combine every 24 consecutive scenes together, as each individual scene is very short, and this will allow our model to understand context properly.

- For cases where multiple consecutive words are labeled the same (for example “ ugly naked man”) we modify it such that only the last word(“man”) is labeled and the rest are unlabeled.
- We set all characters that have been labeled less than 20 times as “<OTH>”.
- We set the label of the speakers to be their corresponding label from the reference. This is done just for uniformity.
- We converted all the words to vectors using the Google's News 300 word2vec.

Model Architecture -

- Embedding Layer: responsible for creating and learning embeddings for each of the different entities and speakers. This also acts as the entity library.
  The embedding layer is used to get the embedding for the speaker of the utterance which is used later.
- The word embedding and the speaker embedding are concatenated and passed through a tanh activation which together constitute a single unit for later processing.
- BiLSTM Layer: a simple Bi-LSTM layer which takes in the word and speaker embeddings. The hidden state of this BiLSTM is the used to calculate the predicted speaker vector.
- Dense Layer: a dense layer is then used to get the predicted speaker vector from the hidden state of the BiLSTM for each reference.
- We then check cosine similarity of the reference speaker vector with each of the entities in the entity library and add a softmax layer over the cosine similarity values.
- We also have two dropout layers after the tanh activation and after the bi-lstm la
- This gives us the class scores which is then used to calculate loss.
- We used the Adam optimizer and Negative Log Likelihood loss for training.

Transformer Architecture -

The proposed C2 framework for joint coreference resolution and character linking in multiparty conversation encodes the conversation and mentions with a shared mention representation encoder module, which includes a pre-trained transformer text encoder and a mention-level self-attention module. The model uses pre-trained language models to obtain contextualized representations for mentions and appends speaker embeddings to each mention. The Mention-Level Self-Attention layer is introduced to refine mention representations given the presence of other mentions in the document. Coreference resolution is modeled as an antecedent finding problem, and character linking is formulated as a multi-class classification problem. The joint loss function optimizes both tasks simultaneously by minimizing the negative log-likelihood of the possibility that each mention is linked to a gold antecedent or the correct referent character.

Progress against Timeline

On time, planned to complete lit review and baseline model by 20th. On track to complete baseline by 20th.

We have finished implementing the current SOTA model. We couldn't make any considerable improvements or adjustments to AMORE-UPF and the current SOTA.

Literature Review

We have read through the following papers, expanding our literature review -

- AMORE-UPF at SemEval-2018 Task 4: BiLSTM with Entity Library
- KNU CI System at SemEval-2018 Task4: Character Identification by Solving Sequence-Labeling Problem
- Joint Coreference Resolution and Character Linking for Multiparty Conversation

The former is a paper that describes the submission made by the authors to the same task. The paper describes the AMORE-UPF system, which was developed for SemEval-2018 Task 4, "Character Identification on Multiparty Dialogues". The system utilizes a Bidirectional Long Short-Term Memory (BiLSTM) model with an Entity Library that stores all the named entities mentioned in the training set. The Entity Library is used to help the model recognize and identify named entities in the test set. Use of the Entity Library proved to be particularly effective, as it improved the accuracy of named entity recognition and reduced the amount of training data required.

The second paper describes the KNU CI system developed for the same task. This system uses a sequence-labeling approach to identify characters in multiparty dialogues. The approach involves labeling each word in the dialogue as either a character or not a character, using a BiLSTM-CRF model. Use of contextual word embeddings, including ELMo and BERT, improved the performance of the system.

The third paper describes the C^2 Model for the same task.This is a joint learning model of Coreference resolution and Character linking which adopts a transformer-based text encoder and includes a mention-level self-attention (MLSA) module that enables the model to do mention-level contextualization. Along with that, a joint loss function is designed and utilized so that both tasks can be jointly optimized.