Template and Slot filling using Generative Transformers and Deep Learning networks

Anupam Dwivedi
Enrollment no. IIT2020198
B-TECH Information Technology
Indian Institute of Information Technology
Prayagraj, India

***Abstract*—Template filling is generally tackled by a pipeline of two separate supervised systems – one for role-filler extraction and another for template/event recognition. Since pipelines consider events in isolation, they can suffer from error propagation. A framework is introduced, based on end-to-end generative transformers for this task (i.e., GTT). It naturally models the dependence between entities both within a single event and across the multiple events described in a document. It is further shown that the framework specifically improves performance on documents containing multiple events. Also different combinations of encoders and decoders are used for the task specifically RNNs and CharCNNs. Different pre-trained transformers such as RoBERTa and DistilBERT are used and evaluated and compared against models in previous literature. Github Project link: https://github.com/a-n-u-p-a-m/NLPCOURSEPROJECT**

***Index Terms*—Transformers, Template, Tokenization, Encoder-decoder, Role-filler Entity extraction(REE)**

I. Introduction

To naturally model between-event dependencies across a document for template filling, a framework is proposed called "GTT" based on generative transformers. The framework is built upon GRIT, which tackles role-filler entity extraction (REE), but not template/event recognition. GRIT performs REE by "generating" a sequence of role-filler entities, one role at a time in a prescribed manner. For the template-filling setting, the GRIT approach is extended to include tokens representing event types as part of the input sequence. The decoder is further modified to attend to the event type tokens, allowing it to distinguish among events and associate event types to each role-filler entity that it generates. The models also use different types of neural networks such as RNNs and CharCNNs as well as different transformers as the base model for the encoder-decoder architecture. The model is evaluated on the MUC-4 (1992) template filling task.

II. Dataset Description

- MUC-4 consists of 1,700 documents with associated templates. We split the dataset as: 1,300 documents for training, 200 documents (TST1+TST2) as the development set and 200 documents (TST3+TST4) as the test set.
- There are 6 event types (i.e., kidnapping, attack, bombing, robbery, arson, forced work stoppage) in MUC-4..

III. Literature Review

There is previous work available that used both transformers as well as different Deep Learning models for the task of template filling. The task of this project is to evaluate models defined in this project and compare them with the existing models defined in the following papers.

- A generative model for template filling, a task that involves filling structured templates with information extracted from a given document.
- The model transforms template filling into a sequence generation problem and uses a generative transformer-based approach to solve it.
- The source sequence is prepared by adding event/template type tokens to the words of the document, along with separator tokens.
- The target sequence consists of the concatenation of template extractions, separated by the separator token, with each template's event type and role-filler entities represented as sub-sequences.
- The model uses a BERT model as an encoder and decoder, and pointer decoding is used to select the next token in the generated sequence.
- The paper introduces several decoding constraints to ensure that the generated output is structured and valid, including downweighting factor, decoding cutoff stop, and a constraint to ensure that the pointers for the start and end token for one entity are in order.
- The authors evaluate their model on the MUC-4 dataset and show that it outperforms several strong baselines, achieving state-of-the-art results.
- They also perform ablation studies to analyze the effect of different components of their model and demonstrate the importance of each component.
- A new approach for extracting event role fillers from a document.
- The approach uses a hierarchical model that combines different levels of representation to generate the final output.
- The different levels of representation include word-level, sentence-level, and document-level representations.
- The word-level representations are obtained using pre-trained contextualized embedding models.

- The sentence-level representations are obtained using a bi-directional LSTM.
- The document-level representations are obtained by encoding the document as a sequence of sentences using another bi-directional LSTM.
- The final output is generated using a joint inference model that takes into account all the representations at different levels.
- The approach was evaluated on the MUC-4 dataset.
- The approach achieved state-of-the-art results on both datasets.
- The authors conclude that their approach is effective for document-level event role filler extraction and can be easily adapted to other related tasks.
- We examine the capabilities of a unified, multitask framework for three information extraction tasks: named entity recognition, relation extraction, and event extraction.
- Token encoding: DYGIE++ uses BERT for token representations using a "sliding window" approach, feeding each sentence to BERT together with a size-L neighborhood of surrounding sentences.
- Span enumeration: Spans of text are enumerated and constructed by concatenating the tokens representing their left and right endpoints, together with a learned span width embedding
- Span graph propagation: A graph structure is generated dynamically based on the model's current best guess at the relations present among the spans in the document. Each span representation

$$g_j^t$$

  is updated by integrating span representations from its neighbors in the graph according to three variants of graph propagation.
- Multi-task classification: The re-contextualized representations are input to scoring functions which make predictions for each of the end tasks.A two-layer feedforward neural net is used (FFNN) as the scoring function.
- Four different datasets are used for experimentation: ACE05, SciERC, GENIA and WLPC.
- A novel approach is proposed that models the dependencies among variables of events, entities, and their relations, and performs joint inference of these variables across a document. The goal is to enable access to document-level contextual information and facilitate contextaware predictions.
- The learning problem is firstly decomposed into three tractable subproblems: (1) learning the dependencies between a single event and all of its potential arguments, (2) learning the cooccurrence relations between events across the document, and (3) learning for entity extraction.
- A joint inference framework is introduced that combines probabilistic models of within-event structures, event-event relations, and entity extraction for joint extraction of the set of entities and events over the whole document.
- Extensive experiments are conducted on the Automatic Content Extraction (ACE) corpus, and show that the described approach significantly outperforms the state-of-the-art methods for event extraction and a strong baseline for entity extraction.

IV. Methodology

The model armature of BERT is amulti-layer bidirec- tional Transformer encoder grounded on the original Trans- former model. The input representation is a consecution of WordPiece embeddings, positional embeddings, and the member embedding. Especially, for single judgment bracket and trailing tasks, the member embedding has no demarcation. A special bracket embedding (CLS)) is fitted as the first commemorative and a special commemorative (SEP)) is added as the final commemorative. The BERT model is pre-trained with two strategies on large- scale unlabeled textbook, i.e., masked language model and coming judgment pre- diction. Thepre-trained BERT model provides a important environment-dependent judgment representation and can be used for colorful target tasks, i.e., intent bracket and niche stuffing, through the finetuning procedure, analogous to how it's used for other NLP tasks. • BERT can be fluently extended to a common intent classification and niche filling model. Grounded on the retired state of the first special commemorative((CLS)), denoted h1, the intent is prognosticated as y i = softmax(Wih1 b i),(1) For niche stuffing, we feed the final retired countries of other commemoratives h2, , hT into a softmax subcaste to classify over the niche filling markers. • For common intent discovery and niche stuffing is illustrated, no encoder side, we use a bidirectional RNN. Bidirectional RNN has been successfully applied in speech recognition and spoken language understanding. We use LSTM as the introductory intermittent network unit for its capability to more model long- term dependences comparing to simple RNN. The decoder is a unidirectional RNN. Again, we use an LSTM cell as the introductory RNN unit. • For common modeling of intent discovery and niche stuffing, we add an fresh decoder for intent discovery(or intent bracket) task that shares the same encoder with niche filling decoder. During model training, costs from both decoders are back-propagated to the encoder. The intent decoder generates only one single affair which is the intent class distribution of the judgment , and therefore alignment isn't needed. • In the proposed model, a bidirectional RNN(BiRNN) reads the source sequence in both forward and backward directions. We use LSTM cell for the introductory RNN unit. niche marker dependences are modeled in the forward RNN. analogous to the encoder module in the below described encoder- decoder armature, the retired state hi at each step is a consecution of the forward state fhi and backward state bhi, hi = (fhi, bhi). Each retired state hi contains information of the whole input word sequence, with strong focus on the corridor girding the word at step i. This retired state hi is also combined with the environment vector ci to produce the marker distribution, where the environment vector ci is calculated as a weighted average of the RNN retired countries h = (h1,., hT). • The

model transforms template filling into a sequence generation problem and uses a generative motor- grounded approach to break it. The source sequence is prepared by adding event/ template type commemoratives to the words of the document, along with division commemoratives. The target sequence consists of the consecution of template lines, separated by the division commemorative, with each template's event type and part- padding realities represented assub-sequences. The model uses a BERT model as an encoder and decoder, and pointer decoding is used to elect the coming commemorative in the generated sequence. • Proposition of a template- grounded system for NER, treating NER as a language model ranking problem in a sequence- to- sequence frame, where original rulings and statement templates filled by seeker named reality span are regarded as the source sequence and the navigator- get sequence, independently. For conclusion, the model is needed to classify each seeker span grounded on the corresponding template scores.

V. Results

- On the ATIS dataset the BERT model has 0.9787 intent accuracy, 0.9559 slot F1 and 0.8824 sentence accuracy. On the ATIS dataset the ALBERT model has 0.9764 intent accuracy, 0.9578 slot F1 and 0.8813 sentence accuracy. On the ATIS dataset the DistilBERT model has 0.9776 intent accuracy, 0.9550 slot F1 and 0.8768 sentence accuracy.
- On the MUC-4 dataset it is concluded that for 'Victim' the precision is the best and for 'target' it is the worst. Moreover the generative transformer provides a total precision of 0.6419, a recall of 0.4736 and a F1 score of 0.5450. This accuracy is increased for the testing cases of the dataset.

VI. Future Scope

In this project I have studied and implemented different techniques for different Information Extraction tasks. For future work there is a possibility of checking the models in some Out-Of-Distribution(OOD) cases. Also an ensemble can be used for the task instead of a single encoder-decoder. There is also the possibility of checking different transformer networks for the tasks and fine-tuning them.

VII. Conclusion

In this project I studied, and analyzed some techniques and models for different information extraction tasks. These tasks include Named Entity Recognition, Template Filling, Event extraction, etc. I studied different transformers and deep learning models and used them for the mentioned tasks. In the end I concluded that using generative transformers is better for the task of template filling over traditional deep learning models.

References

[1] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana - Practical Natural Language Processing-A Comprehensive Guide to Building Real-World NLP Systems-O'Reilly Media, Inc. (2020)

[2] Template-Based Information Extraction without the Templates by Nathanael Chambers and Dan Jurafsky

[3] Newton Spolaor, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. 2013. A comparison of multi-label feature selection methods using the problem transformation approach. Electronic Notes in Theoretical Computer Science, 292:135– 151.

[4] GRIT: Generative Role-filler Transformers for Document-level Event Entity Extraction by Xinya Du Alexander M. Rush Claire Cardie

[5] Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling by Bing Liu , Ian Lane