Final NLU project template

Mattia Carolo (232126)

University of Trento

mattia.carolo@studenti.unitn.it

1. Introduction

Multitask learning (MTL) is a field of machine learning where the main goal is to tackle multiple problems at the same time. In this work the focus is on Intent classification and Slot Filling which are two sentence level tasks where the framework needs to:

- *Assign an intent (can be more than one) for a given sentence or utterance*
- *map the sentence to a sequence of domain-slot labels*

As we can see this is the perfect environment for MTL where a model can be built in order to predict both tasks by leveraging even on the gained knowledge of the other task.

To make a comparison between different models i picked two that leverages on BiLSTM layers used as encoders, one is BERT and the last one uses a Stack-Propagation method that will be discussed later

2. Task Formalisation (approx. 200 words)

The first task that needs to be solved is the intent detection part. In this part the task is typically a sentence classification problem where key features are passed through a classification algorithm to predict the class of a sentence or utterance from a specified set of classes (An example is ”Have u seen my jacket” that will be classified as ”Info Request).

The second critical task is slot filling where a label is attached to each token in an utterance. Through this method each token will have a representation that will encode his semantic information relative to the word represented. Moreover more tokens can be represented as a span that is represented using the BIO notation. In this case the problem is treated as a sequence labelling task (An example is ” I want to travel to Rome” will become ”O O O O O B-fromloc”)*1

By leveraging on MTL a model can be built so that it can learn and infer on two tasks together in order to be better than two working on the same task individually. This is reached thanks to the better generalization where the system prefers solutions which tackles more than one solution. Through this joint task the goal is to retrieve conditional relationship between the different models in order to better generalize once the learning is done

*1 https://arxiv.org/pdf/2101.08091.pdf

*2https://ruder.io/multi-task/

3. Data Description Analysis (approx. 200-500 words)

For this project both ATIS and SNIPS dataset were used, separately, in order to train the models as per intstruction.

3.1. ATIS

The ATIS (Airline Travel Information Systems) consist on 4978 samples for the training set and 893 for the test. Since there was no validation test for this dataset I used the same method seen during lab lectures to rearrange the sets in order to obtain a validation test from the training one. This results will reduce the size of the training sample to 4381 and will create a validatrion test with thte size of 597. Here 26 different types of intents are found and with 129 slot labels. Each sample of the dataset is composed by three entries

```
{
    'utterance': 'what type of aircraft
does eastern fly from atlanta to denver
before 6 pm',

    'slots': 'O O O O O B-airline_name O O
B-fromloc.city_name O B-toloc.city_name
B-depart_time.time_relative
B-depart_time.time I-depart_time.time',

    'intent': 'aircraft'
}
```

3.2. SNIPS

The SNIPS dataset is composed of 13084 samples in the train set, 700 samples in the dev set and 700 samples in the test set. The total number of intents is 7, while the slot labels are 72. Each sample is composed by the utterance, the sequence of slot labels, and the intent. [3-joe]

```
{
    'utterance': 'what type of aircraft
does eastern fly from atlanta to denver
before 6 pm',

    'slots': 'O O O O O B-airline_name O O
B-fromloc.city_name O B-toloc.city_name
B-depart_time.time_relative
B-depart_time.time I-depart_time.time',

    'intent': 'aircraft'
}
```

1/ C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, “The ATIS spoken language systems pilot corpus,” in Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990, 1990.

4. Model

For the comparison 4 models where trained in order to get various results that will be compared with the results achievable through the model introduced during classes. For the first three models the same loss was used which is a cross enthropy loss which leverages on both losses during training and testing

4.1. Bi-Model RNN

Originally designed by Wang et al. [3], Bi-Model RNN2 consists of two identical bi-directional LSTM [4] models, one for predicting the intent and one for the slots. As shown in Fig.1, the flow of hidden states between the two LSTM encoders connects the information learned by the model about intents and slots. giz For this model there wasn't an implementation given so I resorted to an unofficial representations where some adjustments are made like removing the decoder which reduce the time by a total of 54 minutus/epoch.

4.2. Bi-LSTM + Self-attention

With the rise of the attention mechanism's popularity, many models, such as the one developed by Liu et al. (Attention Bi-RNN, Fig. 2) [5], have been applied to the joint intent classification and slot filling task. Attention Bi-RNN is built upon an bidirectional LSTM encoder which receives the input token embeddings and feeds output and hidden states to an- other LSTM layer. The second layer takes context information as a weighted average of the first layer's output, as per the attention mechanism. Finally, intent detection and slot filling is achieved using the outputs of the second layer. I designed a network based on the work of Liu et al. [5], as illustrated in Fig. 3. This architecture consists of a Bi-LSTM encoder with a hidden size of 100 and a multi-head self- attention layer. The Bi-LSTM encodes the word embeddings (300 embedding dim.) to vectors and the multihead attention layer discovers semantic dependencies between the encodings. This setup improves the training speed due to the superior performance of the attention mechanism compared to LSTM layers.

4.3. BERT

BERT [6] is a transformer-based network [7] pre-trained on vast datasets for masked language modeling and next sentence prediction. It can be fine-tuned for various tasks, including intent prediction and slot filling. Chen et al. [8] have performed this experiment by feeding a prompt's utterance to BERT and taking the first token's output for intent prediction and all the resulting hidden states for slot filling (Fig. 4)

4.4. Stack-Propagation SLU

In this model I haven't done any modifications if not for only a better readability or changing the the internal evaluator in order to have conll-like evaluation. This particular model proposed because it takes a different approach during the learning part. With Stack-Propagation the model opposite to learn the correlations between different tasks by the shared encoder it uses the different tasks together during the learning so that different tasks can enhance the learning of others.

The architecture of this framework can be seen in Figure X where there is present one encoder and two decoders. It's worth noting that the BiLSTM is shared for both slot filling and intent detection part where thanks to a self attention mechanism it can leverage information acquired during the learning where after getting the intent information it can later relate the slots to a specific intent

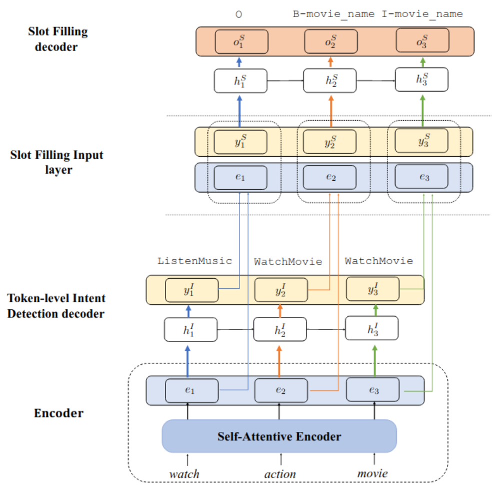

Figure 1: *Example of image crop obtained using Color Mapping. On the left the original image is displayed. On the right the generated fractal is shown*

5. Evaluation (approx. 400-800 words)

5.1. Loss

As a baseline we took

5.2. Loss

I define each model's prediction error as the sum of the cross-entropy between the ground truth intents and slots and the predicted intent and slot probabilities. Formally, if there are I possible intents and S slots, given a sample of T tokens, the total loss of a model is computed as (1). Ground truth intents 3 and slots are represented as y(i) and y(s), where the model's output is ˆy(i) and ˆy(s). In addition, I perform L2 regularization with a factor of = 0.001 to prevent over-fitting. Therefore, the total loss of a model with the parameters W is defined as

5.3. Metrics

The main evaluation has been done on two main metrics, that were specified in the instruction of the project.

- F1 score for slot filling: the F1 score is a metric of accuracy that take into consideration both the precision and the recall of the predictions.
- Accuracy for intent classification: mathematically, the accuracy represents the ratio of the sum of true positive and true negatives out of all the predictions.

5.4. Results

The results can be summarized by Table 1 below for both SNIPS and ATIS datasets and as can be seen Stack-Propagation SLU collects far better results wrt. others that barely reach a 10% negative difference from the given baseline on average. By taking into account the single runs sometimes we have better results that overcome the baseline but still in average especially the SNIPS datasets the result for the remaining models are quite low in average.

Model	ATIS	SNIPS
Bi-Model RNN	28.275	intent 0.833 slot 0.13
Bi-LSTM + Self-attention	0.8241	0.8385714285714285
BERT	0.7826	0.8371 ±1.806
Stack-Propagation SLU	0.9552	0.9880 ±1.806
Average	119.665 ±1.806	119.665 ±1.80

Table 1: *Averages of for both datasets*

6. Conclusion

The major part that are worth talking about is a comparison between the Bi-Model RNN and the Stack-Propagation SLU. The first model in the paper claimed a very high performance on the given dataset but there wasn't any implementation. I tried asking to the curator of the unofficial implementations but it seems that even by adding layers or fine tuning some part the end results won't reach the one given by the authors which cannot be said for the second model that not only has reached the baseline given by the paper but in some runs it managed even to surpass it in average by taking only the epochs after usually 40 generations

7. References