Final NLU project

Jacopo Donà (229369)

University of Trento

jacopo.dona@studenti.unitn.it

This document reports the project of the Natural Language Understanding course of Jacopo Donà. The purpose of this project is to develop a model able to efficiently perform intent classification and slot filling in a multitask learning setting.

1. Introduction

Intent classification and slot filling are two critical tasks for natural language understanding:

- Intent classification aims at assigning a intent to a given utterance.
- Slot filling assigns a label to each token in a utterance, which correspond to different parameters of the user's query.

Initially treated as two independent tasks, recent works have used joint models to solve both tasks, learning shared representation that allow the model to learn the dependency between intents and slots in a sentence.

This report discusses three different proposed solutions:

- The first model is taken from the lab experience and consists in a LSTM encoder and two separate classifying layers, is used as baseline.
- The second model uses a similar architecture to the first one, but the one directional LSTM is substituted with a bi-directional GRU.
- The third model uses a pretrained BERT transformer, exploiting the attention mechanism to better capture context along the sentence.

2. Task Formalisation

Intent detection is the process of analysing an user's utterance and classify its query into one of several predefined classes, that is, intents.

Slot-filling on the other hand consists in a sequence to sequence mapping, which goal is to identify the semantic constituents contained in the user's utterance.

Initially treated as two independent tasks, recent works have explored the possibility of using joint models as a solution for solving both tasks, with the goal of being able to learn the dependencies between intents and slots in sentences. These proposed models have shown to be the better approach, outscoring the separate models in both tasks.

Multitasks model can be divided into two main sections: shared layers and task-independent layers. Earlier layers are trained to learn a shared representation for both tasks, while the final layers are independent from each other and learn task-specific representations. In addition to better learn dependency between the tasks, this practice can also bring good regularization properties, as it counters overfitting which can be a serious issue, especially when working on small datasets.

3. Data Description & Analysis

All models were trained and tested on 2 datasets, ATIS and SNIPS. For ATIS, an additional development set was extracted from the training set in order to test accuracy and f1-scores on data samples not used during training phase.

3.1. ATIS

The ATIS (Airline Travel Information Systems) is a dataset consisting of users' queries asking for flight information on automated airline travel inquiry systems. The dataset is split into 4978 samples in the training set and 893 samples in the test set. Additionally, a development set of 597 samples was extracted from the training set, bringing the final training size to 4381 samples. It contains 26 intent labels and 129 slot labels and 861 different words. Intents distribution is imbalanced, with the *flight* intent appearing in $\sim$73% of the samples in the dataset.

Here is reported a sample example:

```
"utterance": "all flights from boston
                to washington",
"slots": "O O O B-fromloc.city_name
          O B-toloc.city_name",
"intent": "flight"
```

3.2. SNIPS

The SNIPS Natural Language Understanding benchmark is a dataset composed of crowdsourced queries distributed among intents of various complexity. The training, development and test sets contain 13084, 700 and 700 utterances, respectively. It contains 72 slot labels and 7 intent types. Intents are almost uniformly distributed, with each class having similar distributions to the others in both train, dev and test set.

Here is reported a sample example:

```
"utterance": "find animated movies nearest
              at a movie house",
"slots": "O B-movie_type I-movie_type
        B-spatial_relation O O
        B-object_location_type
        I-object_location_type",
"intent": "SearchScreeningEvent"
```

3.3. Sentence Length

Sentence length distribution are shown for both datasets in Figure 1. The longest sentence in ATIS contains 46 words, while in SNIPS the longest one has 31 words. This information will be relevant when talking about the BERT model.

4. Model

All models described in this paper have in common the joint learning architecture.

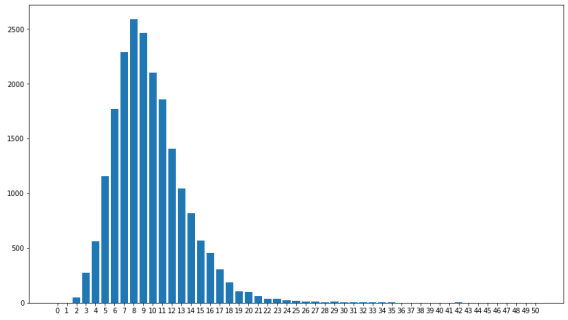


Figure 1: *Sentence length distribution for ATIS and SNIPS*

Given the utterance each model learns a shared representation and then 2 output layers are appended, one for the intent detection task and the other for the slot filling.

4.1. Baseline model

The first model was taken from the lab experience and is used as a baseline to improve upon.

The model is composed of an embedding layer, a LSTM encoder and the task-specific output layers.

During training, a loss is computed for each task (L_i for intent detection and L_s for slot filling) using Cross Entropy Loss and the two values are then summed:

$$Loss = Loss_i + Loss_s$$

The model uses a early stopping criterion based on f1-score during training, it has a patience value of 3. Once the model outputs a worse f1-score for 3 times, the training is stopped and the best performing model is returned. The number of epochs is 200 but the early stopping usually stops training earlier.

The learning rate is 1e-5, Adam is used as optimizer and the network has a 0.1 dropout probability.

4.2. Bi-directional GRU

The second model is an upgraded version of the baseline model, as it shares similarities in the whole architecture.

The embedding layer remains unaltered, while the LSTM encoder is replaced by a 2-layer, bi-directional Gated Recurrent Unit.

The dropout probability is increased from 0.1 to 0.4 and the loss is computed through a weighted summation:

$$Loss = max(\alpha, \beta) * Loss_s + min(\alpha, \beta) * Loss_i$$

Where the two weights are sampled randomly at each batch iteration s.t.

$$\alpha + \beta = 1$$

This was done because, as suggested in [1], training a multi-task model with a weighted summation using randomly sampled weights can improve the model efficiency. In addition to that, since the slot-filling task is the one that more frequently produces an higher loss value, it receives more relevance by being multiplied by the larger of the two weights.

Learning rate has been increased to 5e-5, while the optimizer, epochs and early stopping criterion have been preserved from the previous model.

4.3. BERT

The third proposed model switches the LSTM/GRU encoder with a BERT transformer [2], exploiting the attention mechanism to better model dependencies and context along the sentences.

The architecture of BERT is a multi-layer bidirectional Transformer encoder based on the original Transformer model [3], pretrained on the NLP tasks of Masked Language Modeling and Next Sentence Prediction.

As proposed by [4], Bert can be fine-tuned to jointly learn Intent classification and slot-filling in a multitask setting.

The embedding layer of previous models is replaced by BERT tokenizer, a word-piece tokenizer which "breaks" unknown words into sub-words. The BERT tokenizer and pre-trained model used in this project is *bert-base-uncased*, and as vocabulary, the original BERT vocab created during pre-training is used.

A small pre-processing step is used for the SNIPS dataset, as it was noticed that some sentences in SNIPS contain multiple white-spaces between words (example: *"play a tune or two from kansas city missouri"* contains two white-spaces between city and missouri). This small error caused issues when evaluating slots proposal using *conll*, as 'O' slots were assigned to empty tokens that would interrupt the evaluation methods. To address this, a small pre-processing phase is performed on SNIPS to remove these extra white-spaces and solve the tokenization problem. This issue was not recorded on the ATIS dataset.

In addition to that, BERT tokenizer requires as parameter a max-length value. Every sentence with fewer words than this value is padded, while longer sentences are truncated. Since both datasets do not contain overly long sentences, as described previously, the max-length for the tokenizer was set to 50, causing no utterances to be truncated.

For the training phase, the same loss of Bi-Directional GRU is used, learning rate is set to 5e-5 and epochs are set to 10. The dropout probability is 0.1.

5. Evaluation

The models' performance were evaluated using two metrics: accuracy for intent detection and F1-score for slot filling.

Accuracy is the ratio between the sum of true positive T_P and true negatives T_N divided by the number of total samples

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

The F1-score combines the precision P and recall R of a classifier into a single metric by taking their harmonic mean.

$$F1 = 2 * \frac{P * R}{P + R}$$

where:

$$P = \frac{T_P}{T_P + F_P}$$

$$R = \frac{T_P}{T_P + F_N}$$

Due to the small dataset size, the first two models were trained 5 times on each dataset and mean and standard deviation were computed. Bert model, on the other hand, was trained only once due to longer training times. Results are displayed in Table 1.

Table 1: *Results*

	ATIS		SNIPS	
	Intent Accuracy	Slot F-1	Intent Accuracy	Slot F1
Baseline Model	93.1 +- 0.4 %	92.0 +- 0.4%	96.0 +- 0.9%	79.1 +- 0.9%
Bi-directional GRU	95.7 +- 0.1%	94.9 +- 0.1%	97.0 +- 0.6%	90.0 +- 0.6%
Bert	98.7%	92.0%	98.7%	92.9%

For each dataset, the best and worse performing labels for both slot filling and intent detection were computed, in order to understand which classes the models had more trouble identifying and hypothesize the reasons.

In addition to that, for the intent detection task another method of evaluation was developed, analyzing intent accuracy by sentence length, to see if the models would struggle to identify the context by increasing utterance lengths.

5.1. Baseline model

The model performs reasonably well in intent detection, obtaining good results on both datasets. On ATIS, the model has a 97% accuracy on *flight*, the most frequent intent but struggles with composed intents, such as *airfare-flight, flight-airline* and *flight_no-airline*, scoring 0%. For slot-filling, the worst performing slot labels are the one containing codes or numbers, such as *meal_code,airport_code,fare_amount* or the ones about names, such as *state_name*, scoring 0 f1-score.

On SNIPS, the model has a very good accuracy on intent detection, with the lowest accuracy being 92% on *SearchCreativeWork*, but has issues recognizing slot names, such as *geographic_poi, track, city* having f1-scores below 30%.

On intents classification, the model did not show an accuracy drop off by increasing utterance length, however, the bi-directional analysis carried out by the GRU model and by BERT were overall more capable of capturing the context in both short and long sentences. A comparison between the three models is visible in Figure 2

5.2. Bi-directional GRU

The main motivation for switching from an LSTM encoder to a GRU was to decrease the number of learnable parameters of the network. Early experiments were conducted using mono-directional GRU encoder to have a direct comparison with the baseline LSTM encoder.

Results showed that just switching the encoder to a Gated Recurrent Unit provided a performance increase in both tasks, while also bringing a reduction in training times. My assumption on why the smaller encoder could provide better results is because of the datasets size. The ATIS dataset, in particular, does not contain many training samples. My hypothesis is that the GRU was less susceptible to overfitting due to less parameters and was therefore more capable of learning a more generalized representation.

In addition to that, training times were much faster. Even after switching the 1-layer, mono-directional GRU with a 2-layer, bi-directional one, the second developed architecture is still the faster one to train.

GRU encoders with more than 2 layers were tried but did not bring improvements in neither task.

Compared to the baseline, it increases intent classification on both datasets by $\sim$2%, but still has issues classifying composed intents such as *airfare-flight, flight-airline and flight_no-airline*, having all 0% accuracy. On SNIPS, entity names such as *album* and *track* slots are still difficult to identify, while geographical locations slots such as *geographic_poi,country* and *city* show large improvements.

5.3. BERT

As visible from the Results table, BERT performs brilliantly in the Intent detection, increasing accuracy scores by 4% on ATIS and 2% on SNIPS compared to the baseline and by 2% and 1% compared to the second model.

On ATIS, 7 intents have 100% accuracy, and the most common class *flight* has a 98.8% score. Composed intents still do not perform very well, with *airfare-flight,flight-airline and flight_no-airline* still having 0% accuracy. These labels are not recognized with any of the presented models, making me believe the problem could be caused by the dataset. In Figure 3 the percentage distribution of these labels on the training set is shown, highlighting a poor presence of these labels on the training set, with *flight-airline and airline-flight_no* not being present at all, explaining why every model was not able to identify them.

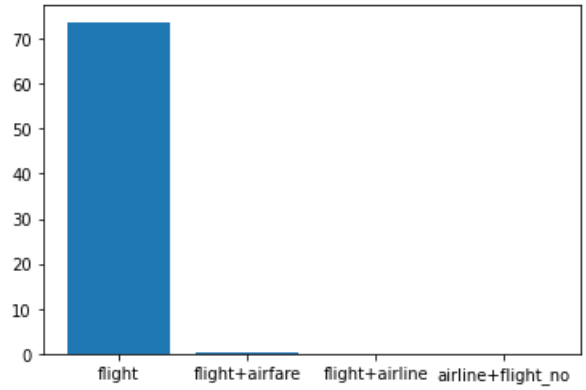
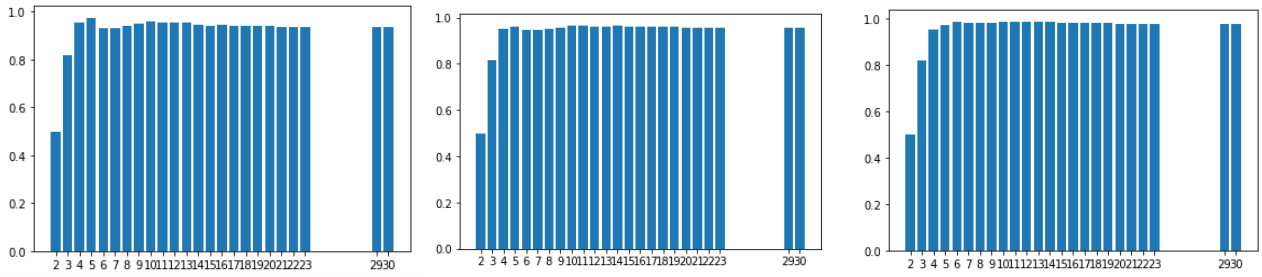


Figure 3: *Intent distribution of poorly classified labels compared to flight*

On SNIPS, the worst performing intent remains *SearchCreativeWork*, but with a 96% accuracy (4% relative improvement over the baseline).

While performing slot filling, f1-scores are improved compared to the previous models on SNIPS, but BERT displays average performances on ATIS, with the same score compared to the baseline and actually worse by 2% compared to the Bi-directional GRU.

This behaviour, in my opinion, is caused by the vocabulary. As mentioned before, the trained BERT model uses a word-piece tokenizer. If the word is in the vocabulary, it is mapped to a numerical id. However, if the word is unknown, it is "broken up" into pieces that are actually present in the vocabulary



(a) (b) (c)

Figure 2: *Intent accuracy by length on ATIS for (a) Baseline (b) Bi-directional GRU (c) BERT*

(subtoken are saved in the vocabulary with a ## prefix). My belief is that the ATIS dataset, containing very specific airplanes, airport and flight words, probably has many tokens broken up by the tokenizer, which decrease the model precision when performing slot-filling. A possible improvement to the model could be to add more tokens to the vocabulary, specific to the ATIS domain, in order to limit the sub-tokenization process.

On SNIPS slot-filling, worst performing slots are still *album* and *track*, but with a 20% and 30% improvement, respectively, compared with the Bi-directional GRU.

6. Conclusion

In conclusion, this report describes the development and evaluation process of the 3 proposed models and proposes hypothesis on performances and on errors.

The Bi-directional GRU and the joint BERT architectures satisfy the goal of improving the baseline of at least 2-3%, with the exception of BERT on ATIS slot-filling for the explained reasons.

I have to say I enjoyed working on this project, as it provided an opportunity to jointly apply the teachings of Natural Language Understanding and of the Deep Learning courses. Furthermore, this project was my first time implementing a transformer into an architecture. It's really impressive seeing the capabilities of this architecture, especially when performing a text classification task, with BERT obtaining very high accuracy scores even with the loss function giving less relevance to the intent detection loss.

7. References

[1] B.Lin, F. YE, Y. Zhang "A closer look at loss weighting in multi-task learning"

[2] J. Devlin, M. Chang, K. Lee, K. Toutanova "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, . Polosukhin "Attention Is All You Need"

[4] Q. Chen, Z. Zhuo,W. Wang "BERT for Joint Intent Classification and Slot Filling"