Final NLU project

Francesco Bozzo (mat. 229312)

University of Trento

francesco.bozzo@studenti.unitn.it

1. Introduction

This project presents some deep learning techniques to predict intents and slots in a multitask learning setting. This work presents and compares three different models by using the ATIS and SNIPS datasets. Obtained by one of the course labs, the baseline model consists of a one-layer LSTM with two heads. The second one is a direct evolution of the baseline by using Bi-LSTM, bigger hidden and embedding sizes, and a more convenient learning rate. The third model uses Bi-GRU, and it shows that it can achieve similar performance with respect to Bi-LSTM while saving 10% on computation time.

2. Task Formalization

The objective consists of optimizing a model to fulfill to tasks that are related between each other: *intent classification* and *slot filling*.

While at the beginning these tasks were considered independently, a strong relationship between them has been found by obtaining state-of-the-art performances. Therefore, a single model can be used to estimate the joint distribution of intent and slot filling labels [1].

Here follows a more formal description with examples of the two considered tasks.

2.1. Intent Classification

Intent classification is defined as the task of assigning a specific intent to a sentence or utterance. More formally:

- Given a sequence of tokens $w = w_1, w_2, ..., w_n$
- and a set of labels L where $l \in L$
- estimate the label $\hat{l}$ such as $\hat{l} = \underset{l}{\text{argmax}} P(l|w)$.

An example from the SNIPS dataset: the given utterance "find heat wave" is associated with the intent `SearchScreeningEvent`.

2.2. Slot filling

Slot filling is defined as a sequence labelling task where the aim is to map a given sentence to a sequence of domain-slot labels, in this case IOB tags. More formally:

- Given a sequence of tokens $w = w_1, w_2, ..., w_n$
- and a sequence of labels as $l = l_1, l_2, ..., l_n$,
- compute the sequence $\hat{l}$ such as $\hat{l} = \underset{l}{\text{argmax}} P(l|w)$.

Table 1 describes a slot filling example from the SNIPS dataset.

Utterance	find	heat	wave
Slots	`O`	`B-movie_name`	`I-movie_name`

Table 1: *Slot filling example from SNIPS.*

3. Data Description Analysis

To assess the quality of the proposed models, two different datasets have been considered: ATIS and SNIPS. Both of them have been post-processed[1] to make each item follow the structure described in Listing 1.

```
{
    "utterance": "what 's the airport at
        orlando",
    "slots": "O O O O O B-city_name",
    "intent": "airport"
}
```

Listing 1: *Example of dataset record from ATIS.*

Here follows a more detailed description regarding both the considered datasets.

3.1. ATIS

The ATIS dataset (Airline Travel Information Systems) contains manual transcripts about humans asking for flight information on automated airline travel inquiry systems.

Originally the dataset is divided only in training and test dataset, respectively 4978 and 893 samples each. A validation dataset has been added by getting 597 items from the training dataset, which corresponds to the 10% of the entire dataset.

Moreover, the dataset contains respectively distinct 863 words, 130 slots, and 26 intents.

The ATIS dataset is also not balanced at all: some slot labels are very common, such as `O` and `B-toloc.city_name` that appear respectively 41579

[1] `https://github.com/BrownFortress/IntentSlotDatasets`.

times (63%) and 5059 (8%), while most of the others do not even appear more than 100 times. Futhermore, 13 slot labels appear only a single time. Moreover, also in terms of intent ATIS is very unbalanced: the intent `flight` appears 73% of the times.

Another limitation of the ATIS dataset is the fact that there are some labels that appear only in test set and not in the training set, for both slot filling (`B-booking_class`, `B-compartment`, `B-flight`, `B-stoploc.airport_code`, `I-depart_time.time_relative`, `I-flight_number`, `I-state_name`) and intent classification (`airfare+flight`, `day_name`, `flight+airline`, `flight_no+airline`). Unless the objective is to test zero shot capabilities (for example using a pre-trained model), this means that there is an upper bound on the final accuracy.

3.2. SNIPS

SNIPS is a dataset composed by 14484 crowd-sourced queries that can be classified on 7 user intents which are not theme-specific: `AddToPlaylist`, `BookRestaurant`, `GetWeather`, `PlayMusic`, `RateBook`, `SearchCreativeWork`, `SearchScreeningEvent`. The SNIPS dataset is also composed by 11420 unique words and 73 slot labels. It is already split in three different datasets: train of 13084 records, validation of 700, and test of 700.

Even though it is more balanced with respect to the ATIS dataset (especially for intent classification), still some slot labels are rare (such as `I-object_select` and `I-object_part_of_series_type`) or too common (such as `O`, which corresponds up to 49% of the dataset).

Differently from ATIS, the SNIPS dataset does not have labels which are only available in the test dataset.

4. Model

In this section three different models will be described:

- *Baseline Model*: it is the same LSTM-based model proposed during the LAB10 lecture;
- *Model A*: it is an evolution of the baseline model, by using bidirectional LSTM;
- *Model B*: it is a slight modification of the Model A, obtained by using bidirectional GRU.

4.1. Baseline Model

The baseline model has a very straightforward architecture:

- embedding layer (size 300);
- 1-layer LSTM (size 200);
- two heads, composed by a linear layer, one for each one of the two tasks.

The module has been trained by using the same techniques and parameters presented in LAB10, specifically 1e-4 as learning rate, Adam as optimizer and plain cross entropy as loss. Moreover, a simple early stopping technique is used to stop the training on based on the slot filling accuracy. Figure 1 presents the model architecture.

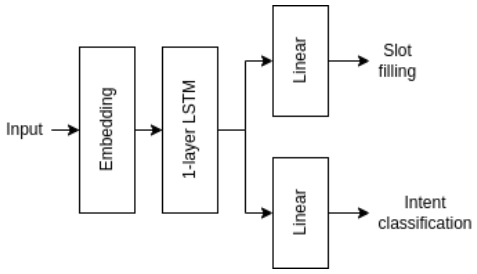

Figure 1: *Baseline model.*

4.2. Model A - Bi-LSTM

The Model A is an evolution of the baseline model, by using bilinear LSTM. In this case the architecture is slightly more complex:

- embedding layer (size 600);
- 2-layer bilinear LSTM (size 400). Instead of using only the past, with Bi-LSTM the model can exploit both past and future words at the same time. Such subsequent layers also improve the results;
- two heads, one for each one of the two tasks:
  - LayerNorm [2], which is nowadays the standard normalization layer for recurrent-based models [3];
  - Linear.

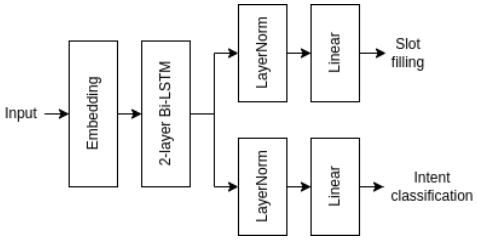

Figure 2: *Model A.*

With this second model, some improvements have been introduced in the training procedure:

- As already described, especially the ATIS dataset is very unbalanced. To improve the results, the cross entropy loss is weighted according to the label frequency, so that also less frequent classes can get more importance. For each class c:

$$w_c = 1 - \frac{freq_c}{\text{dataset size}}$$

- An improved early stopper which takes into account also the loss on the intent classification task.

4.3. Model B - Bi-GRU

The Model B is a slight modification of the Model A: the main difference is that it uses a bidirectional GRU layer instead of the LSTM one. Even though in terms of performance GRU and LSTM should be similar, the former is more efficient and lightweight. Figure 3 shows Model B's architecture.

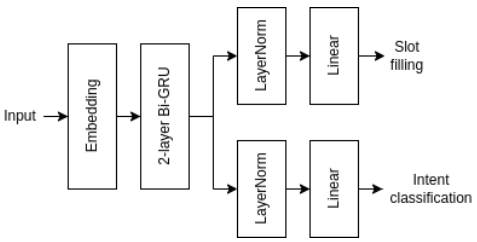

Figure 3: *Model B.*

5. Evaluation

5.1. Metrics

Following the project instructions, the main two metrics that have been considered are *accuracy* and *F1* score:

- Accuracy is an evaluation metric defined as the ratio of true positives and true negatives to all the observations. This value express how likely we can expect a correct prediction from the model. It can be formalized as follows:

$$\text{acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

  where TP, TN, FP, and FN are respectively true positives, true negative, false positives, and false negatives.

- F1 score is an evaluation metric that is built on top of precision and recall. This single-value metric is useful when trying to optimize for both precision (very likely that positive predictions are actually positives) and recall (find all the positive cases). It can be formalized as follows:

$$\text{f1} = 2 * \frac{precision * recall}{precision + recall}$$

5.2. Model Scores

Table 2 collect the scores of the three proposed models on both the ATIS and SNIPS dataset. Even though these results are far away from the state-of-the-art (achieve with pretrained models, such as BERT [4], GPT2 [5], T5 [6]), Model A and B are able to obtain pretty competitive scores.

5.3. Results and Interpretation

Even though the scores seem to be quite decent also for the baseline model, it struggles a lot when dealing with classes that are not frequent: this conclusion is more evident on the ATIS dataset, since it is more unbalanced that SNIPS. In this situation the model tends to overpredict the most common labels, such as `flight` for intent classification on the ATIS dataset. On the contrary, as provided in Table 2, the baseline model is already able to obtain good results when dealing with the balanced intent classification task on the SNIPS dataset.

Moreover, the baseline model struggles when doing slot filling on the ATIS dataset when dealing with numbers departure and arrival information. Even though the model is able to understand if it is dealing with a city, day name, or month number, it is not very good at identifying if it is an information linked to departure or arrival. This problem seems to be partially solved with model A.

Another similar issue that all the models present is the fact they struggle to distinguish between song and album names in the SNIPS dataset.

With respect to the baseline, the model A improves performances of $\sim$2.5% for both the datasets. Once noticeable improvement is in terms of faster convergence thanks to the higher learning rate. This helps to avoid local minimum and to reduce of the required number of epochs to 1/4 with respect to the baseline.

Even though the performance improvement is not that noticeable, the normalization layer on the model A gives more stability (less variance in model performance) and better gradient flow during the backpropagation of the computational graph.

Further tests have been attempted on model A to increase the number of LSTM layers or the depth of the heads. Even though incrementing the number of parameters is generally a rule of thumb in machine learning, in this case the datasets, especially ATIS, are not big enough: during the training process the model suffered from overfitting. Still, the big improvement on SNIPS (which is bigger than ATIS) of model A with respect to the baseline is probably due to the increment of trainable parameters. Dropout has been tested on model A's heads, but it does not improve the performance.

Finally, Model B obtains very similar accuracies and F1 scores with respect to Model A. As already explained, GRU should guarantee the same performance of LSTM, but with less computational power required, thanks to their simplified cell structure. Specifically, Model B's training is $\sim$10% faster than Model A's one.

Figures 4 to 7 show the confusion matrices for Model B, which can be viewed at high resolution on the project repository.

Model	ATIS		SNIPS	
	Slot F1	Intent Acc.	Slot F1	Intent Acc.
Baseline	0.918 ± 0.003	0.937 ± 0.004	0.803 ± 0.012	0.961 ± 0.005
Model A	$\mathbf{0.947 \pm 0.001}$	0.965 ± 0.002	0.899 ± 0.006	0.971 ± 0.002
Model B	0.946 ± 0.001	$\mathbf{0.968 \pm 0.003}$	$\mathbf{0.906 \pm 0.006}$	$\mathbf{0.972 \pm 0.004}$

Table 2: *Model Scores.*

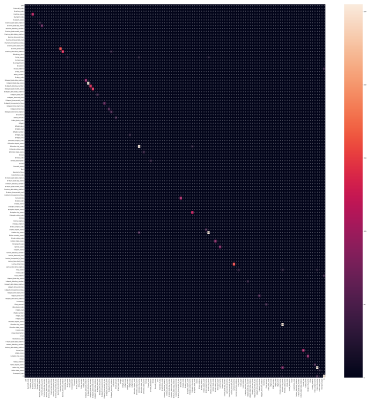

Figure 4: *Confusion matrix for slot filling of Model B on ATIS.*

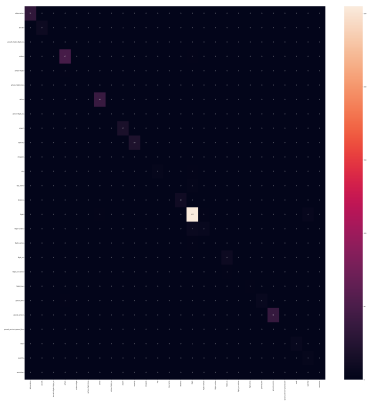

Figure 5: *Confusion matrix for slot filling of Model B on ATIS.*

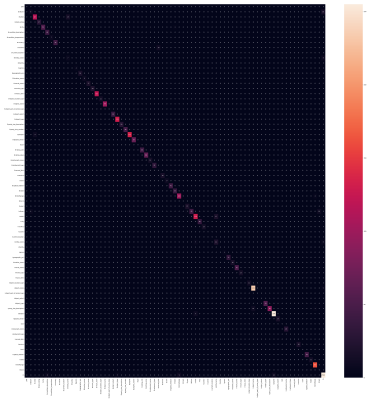

Figure 6: *Confusion matrix for intent classification of Model B on SNIPS.*

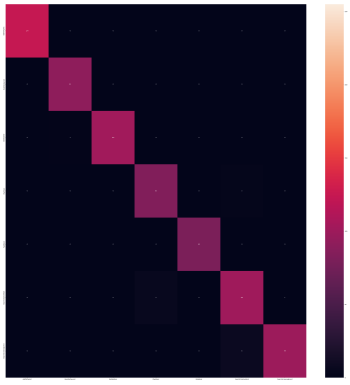

Figure 7: *Confusion matrix for intent classification of Model B on SNIPS.*

6. Conclusion

This work presents and compares three different models by using the ATIS and SNIPS datasets to solve jointly intent classification and slot filling. Moreover, an extensive analysis on results and performance has been carried out, specifically on the dataset composition and model limitations.

Further improvements could be focusing on pretrained models to obtain state-of-the-art results, such as BERT [4], GPT2 [5], and T5 [6]: in this case tokenization and loss computation should be handled carefully. Moreover, to improve the performance on slot filling, also Conditional Random Field (CRF) could be

applied [7].

7. References

[1] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot-filling models in natural language understanding," *CoRR*, vol. abs/2101.08091, 2021. [Online]. Available: https://arxiv.org/abs/2101.08091

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. [Online]. Available: https://arxiv.org/abs/1607.06450

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[7] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *CoRR*, vol. abs/1508.01991, 2015. [Online]. Available: http://arxiv.org/abs/1508.01991