Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection

Anonymous ACL submission

Abstract

Detecting the user's intent and finding the corresponding slots among the utterance's words are important tasks in natural language understanding. Their interconnected nature makes their joint modeling a standard part of training such models. Moreover, data scarceness and specialized vocabularies pose additional challenges. Recently, the advances in pre-trained language models, namely contextualized models such as ELMo and BERT have revolutionized the field by tapping the potential of training very large models with just a few steps of fine-tuning on a task-specific dataset. Here, we leverage such models, and we design a novel architecture on top of them. Moreover, we propose an intent pooling attention mechanism, and we reinforce the slot filling task by fusing intent distributions, word features, and token representations. The experimental results on standard datasets show that our model outperforms both the current non-BERT state of the art as well as stronger BERT-based baselines.

1 Introduction

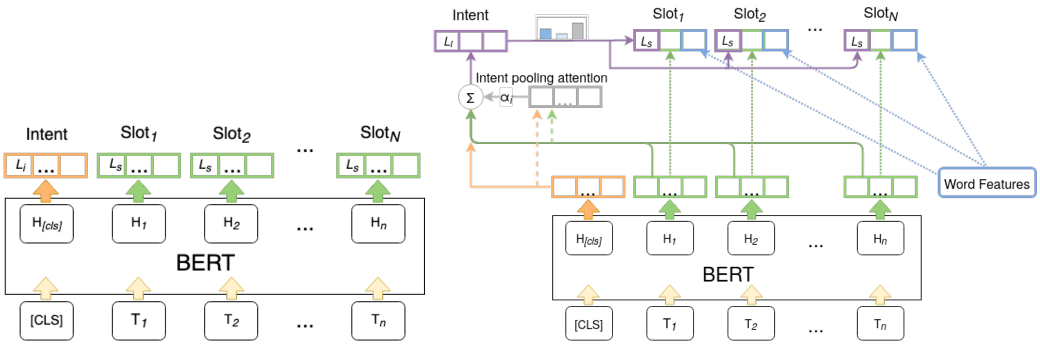
With the proliferation of portable devices, smart speakers, and the evolution of personal assistants, such as Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana, a need for better natural language understanding (NLU) has emerged. Moreover, many Web platforms and applications that interact with the users depend on the abilities of an internal NLU component, e.g., customer service with social media (Huang et al., 2021), in dialogue systems in general (Zeng et al., 2021), for web queries understanding (Tsur et al., 2016; Ye et al., 2016), and general understanding of natural language interaction (Vedula et al., 2020). The major challenges such systems face are *(i)* finding the intention behind the user's request, and *(ii)* gathering the necessary information to complete it via slot filling, while *(iii)* engaging in a dialogue with the user.

Intent	PlayMusic						
Words	play	music	from	2005	by	justin	broadrick
	↓	↓	↓	↓	↓	↓	↓
Slots	O	O	O	B-year	O	B-artist	I-artist

Table 1: Example from the SNIPS dataset with slots encoded in the BIO format. The utterance's intent is *PlayMusic*, and the given slots are *year* and *artist*.

Table 1 shows a user request collected from a personal voice assistant. Here, the intent is to *play music* by the artist *Justin Broadrick* from year *2005*. The slot filling task naturally arises as a sequence tagging task. Conventional neural network architectures, such as RNNs or CNNs are appealing approaches to tackle this problem. Various extensions thereof can be found in previous work (Xu and Sarikaya, 2013a; Goo et al., 2018; Hakkani-Tür et al., 2016; Liu and Lane, 2016; E et al., 2019; Gangadharaiah and Narayanaswamy, 2019). Moreover, sequence tagging approaches such as Maximum Entropy Markov model (MEMM) (Toutanvoa and Manning, 2000; McCallum et al., 2000) and Conditional Random Fields (CRF) (Lafferty et al., 2001; Jeong and Lee, 2008; Huang et al., 2015) have been added on top to enforce better modeling of the dependencies between the posteriors for the slot filling task. Recent work has introduced other methods such as hierarchical structured capsule networks (Xia et al., 2018; Zhang et al., 2019), and graph interactive networks (Qin et al., 2020).

In this work, we investigate the usefulness of pre-trained models for the Natural Language Understanding (NLU). Our approach is based on BERT (Devlin et al., 2019) and its successor RoBERTa (Liu et al., 2019). That model offer two main advantages over previous ones (Hakkani-Tür et al., 2016; Xu and Sarikaya, 2013a; Gangadharaiah and Narayanaswamy, 2019; Liu and Lane, 2016; E et al., 2019; Goo et al., 2018): (*i*) they are based on the Transformer architec-



(a) BERT-Joint.

(b) Transformer-NLU (ours).

Figure 1: Model architectures for joint learning of intent and slot filling: (a) classical joint learning with BERT/RoBERTa, and (b) proposed enhanced version of the model.

ture (Vaswani et al., 2017), which allows them to use bi-directional context when encoding the tokens instead of left-to-right (as in RNNs) or limited windows (as in CNNs), and (*ii*) the model is trained on huge unlabeled text collections, which allows it to leverage relations learned during pre-training, e.g., that *Justin Broadrick* is connected to music or that *San Francisco* is a city.

We further adapt the pre-trained models for the NLU tasks. For the intent, we introduce a pooling attention layer, which uses a weighted sum of the token representations from the last language modelling layer. Moreover, we reinforce the slot representation with the predicted intent distribution, and word features such as predicted word casing, and named entities. To demonstrate its effectiveness, we evaluate it on two publicly available datasets: ATIS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018)

Our contributions can be summarized as follows:

- We enrich a pre-trained language model, such as BERT or RoBERTa, to jointly solve the tasks of intent classification and slot filling.
- We introduce an additional pooling network from the intent classification task, allowing the model to obtain the hidden representation from the entire sequence.
- We use the predicted user intent as an explicit guide for the slot fitting layer rather than just depending on the language model
- We reinforce the slot learning with features such as named entity and true case annotations.
- We present exhaustive analysis of the task-related knowledge in the pre-trained model, for both datasets.

2 Proposed Approach

We propose a joint approach for intent classification and slot filling built on top of a pre-trained language model. We further improve the base model in three ways: (*i*) for intent detection, we obtain a pooled representation from the last hidden states for all tokens (Section 2.1), (*ii*) we obtain predictions for the word case and named entities for each token (word features), and (*iii*) we feed the predicted intent distribution vector, BERT's last hidden representations, and word features into a slot filling layer (see Section 2.2). The complete architecture of the model is shown in Figure 1b.

2.1 Intent Pooling Attention

Here, the task is to jointly learn the two strongly correlated tasks, i.e., intent detection and slot filling. Hereby, using the pooled representation from the [CLS] token can miss important information about the slots' tags when used as an input for predicting the users' intent. We hypothesise that using the token-level representation obtained from the last layer before the slot projection one can help the model in learning the intent detection task, as these representations contain important task-specific information.

Therefore, we introduce a pooling attention layer to better model the relationship between the task-specific representations for each token and for the intent. We further adopt a global concat attention (Luong et al., 2015) as a throughput mechanism. Namely, we learn an alignment function to predict the attention weights α_{int} for each token. We obtain the latter by multiplying the outputs from the language model $H \in \mathbb{R}^{N \times d_h}$ by a latent weight matrix $W_{int\_e} \in \mathbb{R}^{d_h \times d_h}$, where N is the number of tokens in an example and d_h is the hidden size of the Transformer. This is followed by a non-linear $tanh$ activation. In order to obtain importance logit for each token, we multiply the latter by a projection vector $v \in \mathbb{R}^{d_h}$ (shown in Eq. 1). We further normalize and scale (Vaswani et al., 2017) to obtain the attention weights.

$$\alpha_{int} = softmax(\frac{v \cdot \tanh(W_{int\_e} \cdot H^T)}{\sqrt{d_h}}) \quad (1)$$

$$h_{int} = tanh(\sum_{i=1}^{N} \alpha_{int}^i h_{enc}^i) \quad (2)$$

$$y_{int} = W_{int} h_{int}^T + b_{int} \quad (3)$$

Finally, we gather a hidden representation h_{int} as a weighted sum of all attention inputs, and we pass it through a $tanh$ activation (see Eq. 2). For the final prediction, we use a linear projection on top of h_{int}. We apply dropouts on h_{int}, and on the attention weights (Vaswani et al., 2017).

2.2 Slots Modeling

The task of slot filling is closely related to tasks such as part-of-speech (POS) tagging and named entity recognition (NER). Also, it can benefit from knowing the interesting entities in the text. Therefore, we reinforce the slot filling with tags found by a named entity recognizer (word features). Next, we combine the intent prediction, the language model's hidden representations, and some extracted word features into a single vector used for token slot attribution. Details about all components are discussed below.

Word Features A major shortcoming of having free-form text as an input is that it tends not to follow basic grammatical principles or style rules. The casing of words can also guide the models while filling the slots, i.e., upper-case words can refer to names or to abbreviations. Also, knowing the proper casing enabled the use of external NERs or other tools that depend on the text quality.

As a first step, we improve the text casing using a *TrueCase* model from CoreNLP. The model maps the words into the following classes: *UPPER, LOWER, INIT_UPPER, and O*, where *O* is for mixed-case words such as *McVey*. With the text re-cased, we further extract the named entities with a NER annotator. Named entities are recognized using a combination of three CRF sequence taggers trained on various corpora. Numerical entities are recognized using a rule-based system. Both the truecaser and the NER model are part of the Stanford CoreNLP toolkit (Manning et al., 2014).

Finally, we merge some entities ((job) title, ideology, criminal charge) into a special category *other* as they do not correlate directly to the domains of either dataset. Moreover, we add a custom regex-matching entry for *airport_code*, which are three-letter abbreviations of the airports. The latter is specially designed for the ATIS (Tur et al., 2010) dataset. While, marking the proper terms, some of the codes introduce noise, e.g., the proposition *for* could be marked as an *airport_code* because of *FOR (Aeroporto Internacional Pinto Martins, Fortaleza, CE, Brazil)*. In order to mitigate this effect, we do a lookup in a dictionary of English words, and if a match is found, we trigger the *O* class for the token.

In order to allow the network to learn better feature representations for the named entities and the casing, we pass them through a two-layer feed-forward network. The first layer is shown in Eq. 5 followed by a non-linear PReLU activation, where $W_w \in \mathbb{R}^{23 \times 32}$. The second one is a linear projection f_{words} (Eq. 6), where $W_{proj} \in \mathbb{R}^{32 \times 32}$.

$$s_w^i = W_w[ners; cases] + b_w \quad (4)$$

$$h_w^i = max(0, s_w^i) + \alpha * min(0, s_w^i) \quad (5)$$

$$f_{words}(ners, cases) = W_{proj} {h_w^i}^T + b_{proj} \quad (6)$$

Sub-word Alignment Modern NLP approaches suggest the use of sub-word units (Sennrich et al., 2016; Kudo and Richardson, 2018), which mitigate the effects of rare words, while preserving the efficiency of a full-word model. Although they are a flexible framework for tokenization, sub-word units require additional bookkeeping for the models in order to maintain the original alignment between words and their labels.

We first split the sentences into the original word-tag pairs, we then disassemble each one into word pieces (or BPE, in the case of RoBERTa). Next, the original slot tag is assigned to the first word piece, while each subsequent one is marked with a special tag (*X*). Still, the word features from the original token are copied to each unit. To align the predicted labels with the input tags, we keep a binary vector for the active positions.

Slot Filling as Token Classification As in Devlin et al. (2019), we treat the slot filling as token classification, and we apply a shared layer on top of each token's representations to predict the tags.

Furthermore, we assemble the feature vector for the i^{th} slot by concatenating together the predicted intent probabilities, the word features, and the contextual representation from the language model. Afterwards, we add a dropout followed by a linear projection to the proper number of slots:

$$y_s^i = W_s[softmax(y_{int}); f_{words}^i; h_{LM}^i] + b_s \quad (7)$$

2.3 Interaction and Learning

To train the model, we use a joint loss function $\mathcal{L}_{joint}$ for the intent and for the slots. For both tasks, we apply cross-entropy over a softmax activation layer, except in the case of CRF tagging. In those experiments, the slot loss $\mathcal{L}_{slot}$ will be the negative log-likelihood (NLL) loss. Moreover, we introduce a new hyper-parameter γ to balance the objectives of the two tasks. Finally, we propagate the loss from all the non-masked positions in the sequence, including word pieces, and special tokens ([CLS], <s>, etc.). Note that we do *not* freeze any weights during fine-tuning.

3 Experimental Setup

Dataset In our experiments, we use two publicly available datasets, the Airline Travel Information System (ATIS) (Hemphill et al., 1990), and SNIPS (Coucke et al., 2018). The ATIS dataset contains transcripts from audio recordings of flight information requests, while the SNIPS dataset is gathered by a custom intent engine for personal voice assistants. Albeit both are widely used in NLU benchmarks, ATIS is substantially smaller – almost three times in terms of examples, and it contains s times less words. However, it has a richer set of labels, 21 intents and 120 slot categories, as opposed to the 7 intents and 72 slots in SNIPS. Another key difference is the diversity of domains – ATIS has only utterances from the flight domain, while SNIPS covers various subjects, including entertainment, restaurant reservations, weather forecasts, etc. (see Table 2) Furthermore, ATIS allows multiple intent labels. As they only form about 2% of the data, we do not extend our model to multi-label classification. Yet, we add a new intent category for combinations seen in the training dataset, e.g., utterance with intents *flight* and also *airfare*, would be marked as *airfare#flight*. A comparison between the two datasets is shown in Table 2.

	ATIS	SNIPS
Vocab Size	722	11,241
Average Sentence Length	11.28	9.05
#Intents	21	7
#Slots	120	72
#Training Samples	4,478	13,084
#Dev Samples	500	700
#Test Samples	893	700

Table 2: Statistics about the ATIS and SNIPS datasets.

Measures We evaluate our models with three well-established evaluation metrics. The intent detection performance is measured in terms of accuracy. For the slot filling task, we use F1-score. Finally, the joint model is evaluated using sentence-level accuracy, i.e., proportion of examples in the corpus, whose intent and slots are both correctly predicted. Here, we must note that during evaluation we consider only the predictions for aligned words (we omit special tokens, e.g., [CLS], [SEP], <s>, </s>) and word pieces).

Baselines For our baseline models, we use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which we fine-tune. Details about the state-of-the-art model are shown in Appendix A.2. The model's architecture is shown in Figure 1a.

- **BERT** For training the model, we follow the fine-tuning procedure proposed by Devlin et al. (2019). We train a linear layer over the pooled representation of the special [CLS] token to predict the utterance's intent. The latter is optimized during pre-training using the next sentence prediction (NSP) loss to encode the whole sentence. Moreover, we add a shared layer on top of the last hidden representations of the tokens in order to obtain a slot prediction. Both objectives are optimized using a cross-entropy loss.

- **RoBERTa** This model follows the same training procedure as BERT, but drops the NSP task during pre-training. Still, the intent loss is attached to the special start token <s>.

	ATIS			SNIPS		
Model	Intent (Acc)	Sent. (Acc)	Slot (F1)	Intent (Acc)	Sent. (Acc)	Slot (F1)
Joint Seq. (Hakkani-Tür et al., 2016)	92.60	80.70	94.30	96.90	73.20	87.30
Atten.-Based (Liu and Lane, 2016)	91.10	78.90	94.20	96.70	74.10	87.80
Sloted-Gated (Goo et al., 2018)	95.41	83.73	95.42	96.86	76.43	89.27
Capsule-NLU (Zhang et al., 2019)	95.00	83.40	95.20	97.30	80.90	91.80
Interrelated SF-First (E et al., 2019)	97.76	86.79	95.75	97.43	80.57	91.43
Interrelated ID-First (E et al., 2019)	97.09	86.90	95.80	97.29	80.43	92.23
Stack-Propagation (Qin et al., 2019)	96.9	86.5	95.9	98.0	86.9	94.2
AGIF (Qin et al., 2020)	97.1	87.2	96.0	98.1	87.3	94.8
BERT-Joint	97.42	87.57	95.74	98.71	91.57	96.27
RoBERTa-Joint	97.42	87.23	95.32	98.71	90.71	95.85
Transformer-NLU:BERT	**97.87**	**88.69**	**96.25**	**98.86**	91.86	**96.57**
Transformer-NLU:RoBERTa	97.76	87.91	95.65	**98.86**	**92.14**	96.35
Transformer-NLU:BERT w/o Slot Features	97.87	88.35	95.97	98.86	91.57	96.25
Transformer-NLU:BERT w/ CRF	97.42	88.26	96.14	98.57	92.00	96.54

Table 3: Intent detection and slot filling results on the SNIPS and the ATIS datasets. The best results in each category are in **bold**. Our models are in *italic*; the non-italic models on top come from the literature. Qin et al. (2019, 2020) report single-precision results.

4 Experiments and Analysis

Evaluation Results Table 3 presents quantitative evaluation results in terms of (*i*) intent accuracy, (*ii*) sentence accuracy, and (*iii*) slot F1.The first part of the tables refers to previous work, whereas the second part presents our experiments and is separated with a double horizontal line.

While, models become more accurate, the absolute difference between two experiments becomes smaller and smaller, thus a better measurement is needed. Hereby, we introduce a fine-grained measure, i.e., *Relative Error Reduction* (RER) percentage, which is defined as the proportion of absolute error reduced by a $model_a$ compared to $model_b$.

$$RER = 1 - \frac{Error_{model_a}}{Error_{model_b}} \quad (8)$$

Table 4 shows the error reduction by our model compared to the current SOTA (see Appx. A.2), and to a BERT-based baselines (see Section 3). Since there is no single best model from the SOTA, we take the per-column maximum among all, albeit they are not recorded in a single run. For the ATIS dataset, we see a reduction of 11.64% (1.49 points absolute) for sentence accuracy, and 6.25% (0.25 points absolute) for slot F1, but just 4.91% for intent accuracy (see Table 3). Such a small improvement can be due to the quality of the dataset and to its size. For the SNIPS dataset, we see major increase in all measures and over 35% error reduction. In absolute terms, we have 0.76 for intent, 4.84 for sentence, and 1.77 for slots (see Table 3). This effects cannot be only attributed to the better model (discussed in the analysis below), but also to the implicit information that BERT learned during its extensive pre-training. This is especially useful in the case of SNIPS, where fair amount of the slots in categories like *SearchCreativeWork, SearchScreeningEvent, AddToPlaylist, PlayMusic* are names of movies, songs, artists, etc.

Transformer-NLU Analysis We dissect the proposed model by adding or removing prominent components to outline their contributions. The results are shown in the second part of Table 3. First, we compare the results of *BERT-Joint* and the enriched model *Transformer-NLU:BERT*. We can see a notable reduction of the intent classification error by 17.44% and 11.63% for the ATIS and the SNIPS dataset, respectively. Furthermore, we see a 19.87% (ATIS) and 17.35% (SNIPS) error reduction in slot's F1, and 11.43% (ATIS) and 11.63% (SNIPS) for sentence accuracy. We also try RoBERTa as a backbone to our model: while we still see the positive effect of the proposed architecture, the overall results are slightly worse. We attribute this to the different set of pre-training data (CommonCrawl vs. Wikipedia). We further focus our analysis on BERT-based models, since they performed better than RoBERTa-based ones. We further report models' variability in Appendix B.1.

Next, we remove the additional slot features – predicted intent, word casing, and named entities. The results are shown as Transformer-NLU:BERT w/o Slot Features. As expected, the intent accuracy remains unchanged for both datasets, since we retain the pooling attention layer, while the F1-score for the slots decreases. For SNIPS, the model achieved the same score as for *BERT-Joint*, while for ATIS it was within 0.2 points absolute.

Finally, we added a CRF layer on top of the slot network, since it had shown positive effects in earlier studies (Xu and Sarikaya, 2013a; Huang et al., 2015; Liu and Lane, 2016; E et al., 2019). We denote the experiment as *Transformer-NLU:BERT w/ CRF*. However, in our case it did not yield the expected improvement. The results for slot filling are close to the highest recorded, while a drastic drop in intent detection accuracy is observed, i.e., -17.44% for ATIS, and -20.28% for SNIPS. We attribute this degradation to the large gradients from the NLL loss. The effect is even stronger in the case of smaller datasets, making the optimization unstable for parameter-rich models such as BERT. We tried to mitigate this issue by increasing the γ hyper-parameter, effectively reducing the contribution of the slot's loss $\mathcal{L}_{slot}$ to the total, which in turn harmed the slot's F1. Moreover, the model does swap interchangeable slots, rather than the *B-* and *I-* prefixes, or slots unrelated to the intent (see the **Error Analysis** below).

BERT Knowledge Analysis As we start to understand better BERT-based large pre-trained transformer models (Petroni et al., 2019; Rogers et al., 2020), we also start to observe some interesting phenomena. BERT is trained on Wiki articles, which allows it to learn implicit information about the world in addition to learning knowledge about language itself. Here, we evaluate how that former type of knowledge reflects on the two NLU evaluation datasets. As a first step, we extract all the slot phrases from the training sets, i.e., all the words in the slot sequence. Next, we send the latter as a query to Wikipedia and we collect the article titles. Then, we try to match the phrase with an extracted title. In order to reduce the false negatives, we normalize both texts (strip punctuation, replace digits with zeros, lower-case), allow difference of one character between the two, and finally if the title starts with the phrase, we count it as a match

Metric	Relative Error Reduction	
	ATIS	
Intent (Acc)	4.91%	17.44%
Sent. (Acc)	11.64%	11.43%
Slot (F1)	6.25%	19.87%
	SNIPS	
Intent (Acc)	40.00%	11.63 %
Sent. (Acc)	35.91%	6.76%
Slot (F1)	37.64%	17.35%
Transformer-NLU	vs. SOTA	vs. BERT

Table 4: Relative error reduction (Eq. 8) comparing *Transformer-NLU:BERT* to the two baselines: *i)* current SOTA for each measure, and *ii)* conventionally fine-tuned BERT-Joint without the improvements.

(e.g., *Tampa* vs. *Tampa, Florida*). Overall, 66% of the slots in ATIS and 69% in SNIPS matched a Wikipage title.

Next, we evaluate how much of that information is stored in the model by leveraging the standard masking mechanism used during pre-training. In particular, we split each slot in subwords, and then we replace them one by one sequentially with the special [MASK] token. We then sort the predictions for that position by probability and we take the rank of the true word. Finally, we calculate the mean reciprocal rank (MRR) over all the aforementioned ranks: 0.46 for ATIS, and 0.36 for SNIPS. We must note that the BERT's dictionary contains 32K pieces, and the expected uniform MRR is $\sim$1/16,000. Below, we present two examples to illustrate both high- and low-ranked predictions.
High ranked: *play the album jack takes the floor by tom le [MASK] on netflix*, here the model's top predictions are: [**##hrer**, *##rner, ##mmon, ##hr, ##rman*], and the correct token is ranked with the highest probability.
Low ranked: *play some hong jun [MASK]*, here the model's top guesses are mostly punctuation, and general words such as [*to, ;, ##s, and*]. The correct token *##yang* is at position 3,036, which indicates that this term is challenging.

In SNIPS types such as *track, movie_name, entry_name, artist, album* have very high MRR (0.33–0.40), and ones that require numerical value, or are not part of well-known named entities suchf as object_part_of_series_type (OPST) are the lowest (under 0.1). The same in ATIS for country_name

(8e-3), restriction_code (4e-3), meal (4e-3), in contrast to airline_code (0.45), transport_type (0.42), etc. However, ATIS in general does not require such task-specific knowledge, and its MRR is way higher in general, which is reflected by the overall improvement compared to the baseline models.

Error Analysis Here, we discuss what errors the proposed architecture solves compared to the BERT model, and what types of errors are left unsolved. First, we compare the performance of our method (*Transformer-NLU*) to *BERT-Joint (BERT)*. In the intent detection task, the largest improvement (over BERT) comes from examples with slots, indicative for a given intent. This suggests that the model successfully uses the slot information gathered by the pooling attention layer. For the following groups, this is most prominent: (*i*) examples with multi-label intents, e.g., *atis_airline#atis_flight_no – "i need **flight numbers** and **airlines** ... "*, where *BERT* predicted *atis_flight_no*; (*ii*) examples containing distinctive words for another intent class – *"Give me **meal** flights ..."*, *atis_flight → meal (BERT)*, *"I need a **table** ... when it is chiller"*, *GetWeather → BookRestaurant (BERT)*. For all the aforementioned examples, both models filled the slots correctly, but only *Transformer-NLU* captured the correct intent. Moreover, we see a positive effect in detecting *SearchCreativeWork* and *SearchScreeningEvent*, while BERT tends to wrongly fill the slots, and thus swaps the two intents, e.g., *"find **heat wave**"*, or *"find **now and forever**"*. Finally, we see an additional improvement for *AddToPlaylist* and *atis_ground_fare*.

Next, we compare the performance of the two models on slot filling. As expected, the newly proposed model performs better, when the curated features capture some interesting phenomena. We observe that, when filling code slots (**airport, meal, airfare**) – *"what does ... code **bh** mean"*, artists, albums, movies, object names – ***dwele, ny-oil, turk*** (*artist → entity_name (BERT)*), locations – *"... between milwaukee and **indiana**"*, *state → city (BERT)*; BERT confuses ***mango** (city)* with the fruit (cuisine); *"new york city **area**" O → city (BERT)* and time-related ones – ***afternoon, late night, a.m.***.

Finally, we discuss the errors of *Transformer-NLU* in general. For the ATIS dataset, 50% of the wrong intents come from multi-label cases (35% with two labels, and 15% with three), 31% *atis_flight – "how many **flights** does ... **/have to/leave** ... "* (→ atis_quantity), 11% *atis_city – list la* (→ atis_abbreviation), and the others are mistakes in *atis_aircraft*. For the slots, 50% of the errors come from tags that can have a *fromloc/toloc* prefix, e.g., *city, airport_code, airport_name, etc.*, another 20% are time-related (*arrive_date, return_date*), and filled slots without tag 7%. The errors by the model for the SNIPS datasets are as follows: mislabeled intents *PlayMusic* 11%, *SearchCreativeWork* 22%, *SearchScreeningEvent* 67%, slots – *movie_name* 19%, *object_name* 16%, *playlist* 9%, track 9%, entity_name 5%, *album* 4%, *timeRange* 4%, *served_dish* 2%, filled slots without tag 19%. The model misses 9% (ATIS) and 17% (SNIPS) of all the slots that should be filled. This is expected since SNIPS' slots have a larger dictionary (11K words), with a large proportion of the slots being names, and often including prepositions, e.g., *"... trailer of **the multiversity**"*.

5 Related Work

5.1 Intent Classification

Several approaches have focused only on the utterance intent, and have omitted slot information. For example, Hu et al. (2009) mapped each intent domain and user's queries into a Wikipedia representation space, Kim et al. (2017) and Xu and Sarikaya (2013b) used log-linear models with multiple-stages and word features. Ravuri and Stolcke (2015) investigate word and character n-gram language models based on Recurrent Neural Networks and LSTMs (Hochreiter and Schmidhuber, 1997), Xia et al. (2018) proposed a zero-shot transfer thought Capsule Networks (Sabour et al., 2017) and semantic features for detecting the user intent, without labeled data. Moreover, some work addressed the task in a multi-class multi-label setup (Xu and Sarikaya, 2013b; Kim et al., 2017; Gangadharaiah and Narayanaswamy, 2019).

5.2 Slot Filling

Before the rise of deep learning, sequential models such as Maximum Entropy Markov model (MEMM) (Toutanvoa and Manning, 2000; McCallum et al., 2000), and Conditional Random Fields (CRF) (Lafferty et al., 2001; Jeong and Lee, 2008) were the state-of-the-art choice. Recently, several combinations thereof and different neural network architecture were proposed (Xu and Sarikaya, 2013a; Huang et al., 2015; E et al., 2019). Zhu et al.

(2020) explored label embeddings from slots filling and different kinds of prior knowledge such as: atomic concepts, slot descriptions, and slot exemplars. Zhang et al. (2020) used time-delayed neural networks achieving state-of-the-art performance. Siddique et al. (2021) investigated zero-shot transfer of the slot filling knowledge between different tasks. However, a steer away from sequential models is observed in favor of self-attentive ones such as the Transformer (Vaswani et al., 2017; Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2020; Lewis et al., 2020). They compose a contextualized representation of both a sentence, and each of its words, through a sequence of intermediate non-linear hidden layers, usually followed by a projection layer, in order to obtain per-token tags. Such models were successfully applied to closely related tasks, e.g., named entity recognition (NER) (Devlin et al., 2019), part-of-speech (POS) tagging (Tsai et al., 2019), etc.

Approaches modeling the intent or the slot as independent of each other suffer from uncertainty propagation due the absence of shared knowledge between the tasks. To overcome this limitation, we learn both tasks using a joint model.

5.3 Joint Models

Given the correlation between the intent and word-level slot tags, it is natural to train them jointly. Recent surveys covered different aspects of joint and individual modeling of the slot and the intent (Louvan and Magnini, 2020; Weld et al., 2021).

Xu and Sarikaya (2013a) introduced a shared intent and slot hidden state Convolutional Neural Network (CNN), followed by a globally normalized CRF (TriCRF) for sequence tagging. Since then, Recurrent Neural Networks have been dominating, e.g., Hakkani-Tür et al. (2016) used bidirectional LSTMs for slot filling and the last hidden state for intent classification, Liu and Lane (2016) introduced shared attention weights between the slot and the intent layer. Goo et al. (2018) integrated the intent via a gating mechanism into the context vector of LSTM cells used for slot filling.

Qin et al. (2019) used an self-attentive bidirectional LSTM encoder for the input utterances and a dual decoder for the intents and the slots, and they applied both at the token-level. E et al. (2019) introduced a bidirectional interrelated model, using an iterative mechanism to correct the predicted intent and the slot by multiple step refinement. Zhang et al. (2019) tried to exploit the semantic hierarchical relationship between words, slots, and intent via a dynamic routing-by-agreement schema in Capsule Networks (Sabour et al., 2017). Qin et al. (2020) proposed an adaptive graph-interactive framework using BiLSTMs and graph attention networks (GAT) (Velickovic et al., 2018) to model the interaction between intents and slots at the token level. Recently, Qin et al. (2021) introduced a co-interactive Transformer that mixes the slot and the intent information by building a bidirectional connection between them.

Here, we use a pre-trained Transformer, and instead of depending only on the language model's hidden state to learn the interaction between the slot and the intent, we fuse the two tasks together. Namely, we guide the slot filling by the predicted intent, and we use a pooled representation from the task-specific outputs of BERT for intent detection. Moreover, we leverage information from external sources: *(i)* from explicit NER and true case annotations, and *(ii)* from implicit information learned by the language model during its extensive pre-training.

6 Conclusion

We studied the two main challenges in natural language understanding, i.e., intent detection and slot filling. Addressing these tasks is important in a number of scenarios arising on Web platforms and Web-based applications such as customer service in social media, dialogue systems, web queries understanding, and general understanding of natural language interaction with the user.

In particular, we proposed an enriched pre-trained language model to jointly model the two tasks (i.e., intent detection and slot filling), i.e., *Transformer-NLU*. We designed a pooling attention layer in order to obtain intent representation beyond just the pooled one from the special start token. Further, we reinforced the slot filling with word-specific features, and the predicted intent distribution. Our experiments on two standard datasets showed that Transformer-NLU outperforms other alternatives for all standard measures used to evaluate NLU tasks. We found that the use of RoBERTa and adding a CRF layer on top of the slot filling network did not help.

Ethics and Broader Impact

Applicability

Our intent pooling mechanism, as well as the features we introduced, are potentially applicable to other semantic parsing and sequence labeling tasks. They increase the model's size by just few tens of thousands of parameters, which is very efficient in comparison to modern NLP models, which have millions or even billions of parameters.

Biases

On the down side, we would like to warn about the potential biases in the data used for training Transformers such as BERT and RoBERTa, as well as in the ATIS and the SNIPS datasets. Moreover, the use of large-scale Transformers and GPUs could contribute to global warming.

Environmental Impact

Finally, we would also like to warn that the use of large-scale Transformers requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming. This is a bit less of an issue in our case, as we do not train such models from scratch; rather, we fine-tune them on relatively small datasets. Moreover, running on a CPU for inference, once the model has been fine-tuned, is perfectly feasible, and CPUs contribute much less to global warming.

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, INTERSPEECH '16, pages 715–719, San Francisco, USA.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop*, Hidden Valley, Pennsylvania.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jian Hu, Gang Wang, Frederick H. Lochovsky, Jian-Tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, pages 471–480, Madrid, Spain.

Jiantao Huang, Yi-Ru Liou, and Hsin-Hsi Chen. 2021. Enhancing intent detection in customer service with social media data. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 274–275, New York, NY, USA.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging.

Minwoo Jeong and Gary Geunbae Lee. 2008. Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302.

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76(9):11377–11390.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College*, pages 282–289, Williamstown, MA, USA.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, INTERSPEECH '16, pages 685–689, San Francisco, USA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online).

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 591–598, Stanford, CA, USA.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197, Toronto, ON, Canada.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Online.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In *Sixteenth Annual Conference of the International Speech Communication Association*, INTERSPEECH '15, pages 135–139, Dresden, Germany.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 3856–3866, Long Beach, CA, USA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

A.B. Siddique, Fuad Jamour, and Vagelis Hristidis. 2021. Linguistically-enriched and context-awarezero-shot slot filling. In *Proceedings of the Web Conference 2021*, WWW '21, page 3279–3290, New York, NY, USA.

Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3632–3636, Hong Kong, China.

Gilad Tsur, Yuval Pinter, Idan Szpektor, and David Carmel. 2016. Identifying web queries with question intent. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 783–793, Republic and Canton of Geneva, CHE.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in ATIS? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24, Berkeley, California, USA. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, Long Beach, CA, USA.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *Proceedings of The Web Conference 2020*, WWW '20, page 2009–2020, New York, NY, USA.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, Canada.

HENRY Weld, XIAOQI Huang, SIQU Long, JOSIAH Poon, and SOYEON CAREN Han. 2021. A survey of joint intent detection and slot-filling models in natural language understanding.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099, Brussels, Belgium.

Puyang Xu and Ruhi Sarikaya. 2013a. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83, Olomouc, Czech Republic. IEEE.

Puyang Xu and Ruhi Sarikaya. 2013b. Exploiting shared information for multi-intent natural language sentence classification. In *Fourteenth Annual Conference of the International Speech Communication Association*, pages 3785–3789, Lyon, France.

Qi Ye, Feng Wang, and Bo Li. 2016. Starrysky: A practical system to track millions of high-precision query intents. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 961–966, Republic and Canton of Geneva, CHE.

Zengfeng Zeng, Dan Ma, Haiqin Yang, Zhen Gou, and Jianping Shen. 2021. Automatic intent-slot induction for dialogue systems. In *Proceedings of the Web Conference 2021*, WWW '21, page 2578–2589, New York, NY, USA.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy.

Zhen Zhang, Hao Huang, and Kai Wang. 2020. Using deep time delay neural network for slot filling in spoken language understanding. *Symmetry*, 12(6).

Su Zhu, Zijian Zhao, Rao Ma, and Kai Yu. 2020. Prior knowledge driven label embedding for slot filling in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1440–1451.

Appendix

"Enriched Pre-trained Transformers for Joint Slot Filling and Intent Detection"

A Experimental Setup

A.1 Model Details

We use the PyTorch implementation of BERT from the Transformers library of (Wolf et al., 2020) as a base for our models. We fine-tune all models for 50 epochs with hyper-parameters set as follows: batch size of 64 examples, maximum sequence length of 50 word pieces, dropout set to 0.1 (for both attentions and hidden layers), and weight decay of 0.01. For optimization, we use Adam with a learning rate of 8e-05, β_1 0.9, β_2 0.999, ϵ 1e-06, and warm-up proportion of 0.1. Finally, in order to balance between the intent and the slot losses, we set the parameter γ to 0.6, we test the range 0.4–0.8 with 0.1 increment. All the models use the same pre-processing, post-processing, and the standard for these tasks metrics. In order to tackle the problem with random fluctuations for BERT/RoBERTa, we ran the experiments three times and we used the best-performing model on the development set. We define the latter as the highest sum from all three measures described in Appendix 3. All the above-mentioned hyper-parameter values were tuned on the development set, and then used for the final model on the test set. All models were trained on a single K80 GPU instance for around an hour.

A.2 State-of-the-art Models

We further compare our approach to some other benchmark models. Here, we must note that we include models that do not use embeddings from large pre-trained Transformers such as BERT in order to measure the improvements that come solely from the pre-training procedure (see Section 4):

- Joint Seq. (Hakkani-Tür et al., 2016) uses a Recurrent Neural Network (RNN) to obtain hidden states for each token in the sequence for slot filling, and uses the last state to predict the intent.

- Atten.-Based (Liu and Lane, 2016) treats the slot filling task as a generative one, applying sequence-to-sequence RNN to label the input. Further, an attention weighted sum over the encoder's hidden states is used to detect the intent.

- Slotted-Gated (Goo et al., 2018) introduces a special gated mechanism to an LSTM network, thus reinforcing the slot filling with the hidden representation used for the intent detection.

- Capsule-NLU (Zhang et al., 2019) adopts Capsule Networks to exploit the semantic hierarchy between words, slots, and intents using dynamic routing-by-agreement schema.

- Interrelated (E et al., 2019) uses a Bidirectional LSTM with attentive sub-networks for the slot and the intent modeling, and an interrelated mechanism to establish a direct connection between the two. SF (slot), and ID (intent) prefixes indicate which sub-network to execute first.

- Stack-Propagation (Qin et al., 2019) consists of a self-attentive BiLSTM encoder for the utterance and two decoders, one for the intent-detection task that performs a token-level intent detection, and one for the slot filling task.

- AGIF (Qin et al., 2019) uses Adaptive Graph-Interactive Framework to jointly model intent detection and slot filling with an intent-slot graph interaction layer applied to each token adaptively.

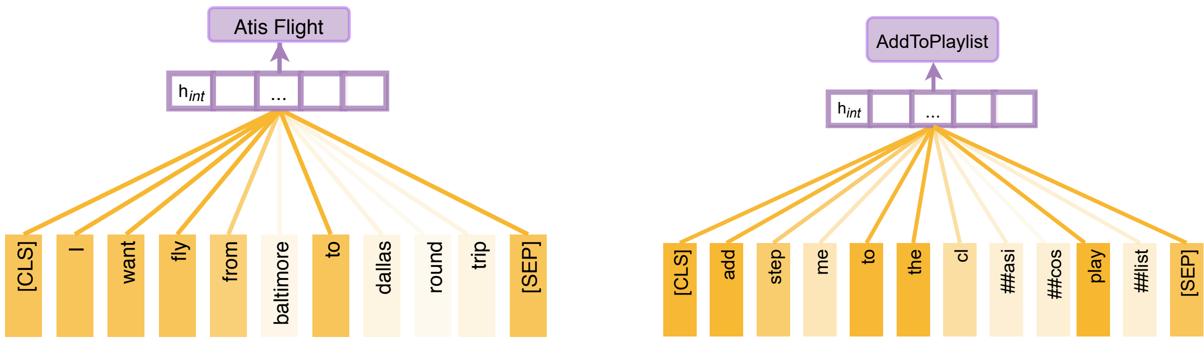
Chen et al. (2019) used BERT with a token classification pipeline to jointly model the slot and the intent, with an additional CRF layer on top.[1] However, they evaluated the slot filling task using per-token F1-score (micro averaging), rather than per-slot entry, as is standard, which in turn artificially inflated their results. As their results are not comparable to the rest, we do not include them in our comparisons.

B Model Analysis

B.1 Variability Analysis

In addition to the results discused in Section 4, we also report the Transformer-NLU:BERT's (and BERT's) μ and σ, 95% confidence internals over all runs: ATIS – Intent 98.0 ± 0.17 (BERT 97.13 ± 0.26), Sentence 88.6 ± 0.23 (BERT 87.8 ± 0), Slot 96.3 ± 0.06 (BERT 96.0 ± 0.14); SNIPS – Intent

[1] In terms of micro-average F1 for slot filling, Chen et al. (2019) reported 96.1 on ATIS and 96.27 on SNIPS (per-token). For comparison, for our joint model, these scores are 98.1 and 97.9 (per-token); however, the correct scores for our model are actually 95.7 and 96.3 (per-slot).



(a) atis_flight (ATIS). (b) AddToPlaylist (SNIPS).

Figure 2: Intent pooling attention weight for one example per dataset. The thicker the line, the higher the attention weight.

98.6 ± 0.14 (BERT 98.42 ± 0), Sentence 92.0 ± 0.17 (BERT 91.8 ± 0.19), Slot 96.2 ± 0.05 (BERT 96.1 ± 0.06). The aforementioned results show that the mean scores of the models in the slot filling task are close, but the variance in Transformer-NLU is lower. Further, we must note that these values are calculated over the best runs from each model retraining, and they are not achieved in a single run.

B.2 Intent Pooling Attention Visualization

Next, we visualize the learned attention weights on Figure 2a. It presents a request from the ATIS dataset: *i want fly from baltimore to dallas round trip*. The utterance's intent is marked as *atis_flight*, and we can see that the attention successfully picked the key tokens, i.e., *I*, *want*, *fly*, *from*, and *to*, whereas supplementary words such as names, locations, dates, etc. have less contribution. Moreover, when trained on the ATIS dataset, the layer tends to set the weights in the two extremes — equally high for important tokens, and towards zero for the rest. We attribute this to the limited domain and vocabulary.

Another example, from the SNIPS dataset, is shown on Figure 2b. Here, the intent is to add a song to a playlist (*AddToPlaylist*). In this example, we see a more diverse spread of attention weights. The model again assigns the highest weight to the most relevant tokens *add*, *to*, *the*, and *play*. Also, the model learned that the first wordpiece has the highest contribution, while the subsequent ones are supplementary.

Finally, we let the pooling attention layer consider the special tokens marking the start and the end ([CLS], and [SEP]) of a sequence, since they are expected to learn semantic sentence-level representations from the penultimate layer. The model assigns high attention weights to both.