IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2021, accepted October 21, 2021, date of publication November 2, 2021,
date of current version November 8, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3124268*

# Unified Transformer Multi-Task Learning for Intent Classification With Entity Recognition

**ALBERTO BENAYAS[1], REYHANEH HASHEMPOUR[2,3], DAMIAN RUMBLE[2], SHOAIB JAMEEL[3], AND RENATO CORDEIRO DE AMORIM[3]**

[1]EPAM Systems, Inc., 29004 Madrid, Spain
[2]BT Group Plc, London EC1A 7AJ, U.K.
[3]Computer Science and Electrical Engineering Department, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Alberto Benayas (alberto_jose_benayas@epam.com)

This work was supported by the Innovate U.K., under Grant 11422.

**ABSTRACT** Intent classification (IC) and Named Entity Recognition (NER) are arguably the two main components needed to build a Natural Language Understanding (NLU) engine, which is a main component of conversational agents. The IC and NER components are closely intertwined and the entities are often connected to the underlying intent. Current research has primarily focused to model IC and NER as two separate units, which results in error propagation, and thus, sub-optimal performance. In this paper, we propose a simple yet effective novel framework for NLU where the parameters of the IC and the NER models are jointly trained in a consolidated parameter space. Text semantic representations are obtained from popular pre-trained contextual language models, which are fine-tuned for our task, and these parameters are propagated to other deep neural layers in our framework leading to a faithful unified modelling of the IC and NER parameters. The overall framework results in a faithful parameter sharing when the training is underway, leading to a more coherent learning. Experiments on two public datasets, ATIS and SNIPS, show that our model outperforms other methods by a noticeable margin. On the SNIPS dataset, we obtain a 1.42% improvement in NER in terms of the F1 score, and 1% improvement in intent accuracy score. On ATIS, we achieve 1.54% improvement in intent accuracy score. We also present qualitative results to showcase the effectiveness of our model.

**INDEX TERMS** Intent classification, named entity recognition, multi-task learning, transfer learning.

## I. INTRODUCTION

As conversational agents become more popular, it is vital to make them more effective. The performance of such agents predominantly relies on their ability to understand what the user says, through the use of a Natural Language Understanding (NLU) engine, so the agent can act in a meaningful way. An NLU engine aims to form a semantic frame that captures the meaning of user utterances or information needs [1]. To this end, each NLU engine performs two main tasks, namely, Intent Classification (IC) and Named Entity Recognition (NER) [2]. The former deals with the understanding of the user's intended desire and the latter identifies references to the real-world objects in the user's utterance. Table 1 presents an example where the user says they want to upgrade their phone. The named entity *"device"* is annotated using the Beginning Inside and Outside (BIO) of a text segment notation.

Developing a reliable NLU engine is not a trivial task. There are many well-documented challenges such as ambiguity of the intents, short sentences, long range dependencies, out-of-vocabulary words, lack of training data, and dealing with new or adapting domains. Traditionally, IC and NER have been independently researched and improved, and in the conversational agents context, NER has usually been referred to as slot filling and framed as a sequence labeling task. Popular approaches to solving sequence labeling problems include Maximum Entropy Markov Model [3] and Conditional Random Field (CRF) [4]. Support Vector Machines (SVM) [5] was used by Moschitti *et al.* [6] with syntactic features and tree kernels for slot filling. Deep learning methods have also been explored such as Recurrent Neural Networks (RNN) [7], [8] and deep belief network [9] which have shown reliable ability to capture dependencies compared to

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

**TABLE 1. An example of an utterance with intent and BIO Named Entity annotation.**

| query | I | want | to | upgrade | my | iPhone | 11 |
|---|---|---|---|---|---|---|---|
| slots | O | O | O | O | O | B-device | I-device |
| Intent | | | | upgrade | | | |

the traditional models such as CRF. Long Short-Term Memory (LSTM) and regression models were used to obtain label dependency for slot filling in Yao *et al.* [10]. For IC, different classifiers have been applied from the traditional machine learning approaches such as SVM [11] and Adaboost [12] to the deep learning models [13]–[15]. A RNN model was proposed in Ji *et al.* [13] to model sentences and words for intent detection. Convolutional Neural Networks (CNN) was used in Hashemi [14] and an RNN in combination with word hashing, to account for the out-of-vocabulary words, was explored in Ravuri and Stolcke [15] to detect intents of the users' queries.

One of the main problems with the models above is that they treat IC and NER, independently. This creates problems as these tasks are not inherently independent. For example, a query like ''pickup my order at orange'' has ''orange'' as the name of a store, while ''buy orange at Tesco'' has ''orange'' as a fruit [16]. An NER model that is aware of the intent is better capable of differentiating between these two entities. Hence, by addressing the two tasks simultaneously, where each task informs the other, one can boost the performance of an NLU engine.

Another serious shortcoming of the approaches mentioned above is error propagation, and amplification from one stage to another. As a result, there has been some effort to model different components as a unified machinery to mitigate this problem. A unified model not only reduces error propagation but the model parameters are shared in a consolidated parameter space, leading to more coherent parameter estimation. Ease of use is another advantage where one has to only train a single model unlike the pipeline approach. The literature shows that a variety of RNN models have been employed to detect intents and entities together [17]–[19]. CNN models have also been employed in Xu and Sarikaya [20], where the features were extracted by the CNN layers and shared between the two tasks. An ensemble of both Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU) was used in Firdaus *et al.* [21] to separate multi-layer perceptrons whose outputs are fused and then projected and a softmax applied to predict intents and slots. The transformer architecture [22], which is a non-recurrent model efficient for capturing global dependencies through multi-head self-attention, has been used in different studies [23], [24] and produced superior results. These unified machine learning models can be categorized as either intent2slot or slot2intent framework [1]. Intent2slot models [25], [26] use intent information to predict the slots and slot2intent models [27] do the opposite, which is using the slot information to predict the intent. We argue that both tasks can inform one another when they are being trained in a unified consolidated parameter space, hence, we benefit from both intent2slot and slot2intent simultaneously.

In this paper, we develop a novel multi-task model for IC and NER. Our main model is a joint computational method where IC and NER parameters are jointly shared. Specifically, we derive text representations from an underlying pre-trained but fine-tuned contextual language model which could be BERT or others. These embeddings are then passed to the IC and the NER model at the same time. The IC model shares its parameter information with the NER model and vice-versa. Our multi-task model makes use of this shared information while solving two problems simultaneously. The other motivation for applying multi-task model is that essential elements of a NLU engine, i.e. intent and entities, can be predicted at once.

Our main technical contribution lies in developing a simple yet effective unified model for IC and NER where features are derived from a pre-trained language model. We have performed extensive experiments and show that our model outperforms current state-of-the-art on both SNIPS [28] and ATIS [29] benchmark datasets. The qualitative results also indicate the effectiveness of our model. In particular, we show that our unified model is able to assign correct intents to the user's utterances where the single-task model lacks this capability, especially in the case of ambiguous intents.

## II. RELATED WORK

In this section, we present an overview of the literature and group the works into the following categories; Traditional, where the two tasks are tackled separately, and Unified models, in which the tasks are handled together.

### A. TRADITIONAL MODELS

Research in NLU emerged from the Airline Travel Information Systems (ATIS) project and some call classification systems in the early 1990s [30]. For IC, features such as n-grams with generic entities, e.g. dates and locations, were typically used. Due to the very large dimension of the input space, classifiers such as Adaboost [31] and SVM [11] were found effective. For NER, most of the works have been based on CRF due to the effectiveness of this model on sequence labeling [4]. The problem with such approaches is that they rely on the domain expert knowledge for feature engineering. To compensate for this, deep learning models were introduced.

Among deep learning models, CNN was used to extract features and LSTM models were applied to take account of sequence (sentence) representation. Costello *et al.* [32] constructed a CNN with character-level embedding and a bidirectional CNN with an attention mechanism. They also explored LSTM and GRU with and without character-level embedding and an attention mechanism for intent detection. Lin and Xu [33] proposed a two-stage method to detect unseen intent labels using a Bi-LSTM to extract features of a sentence. For

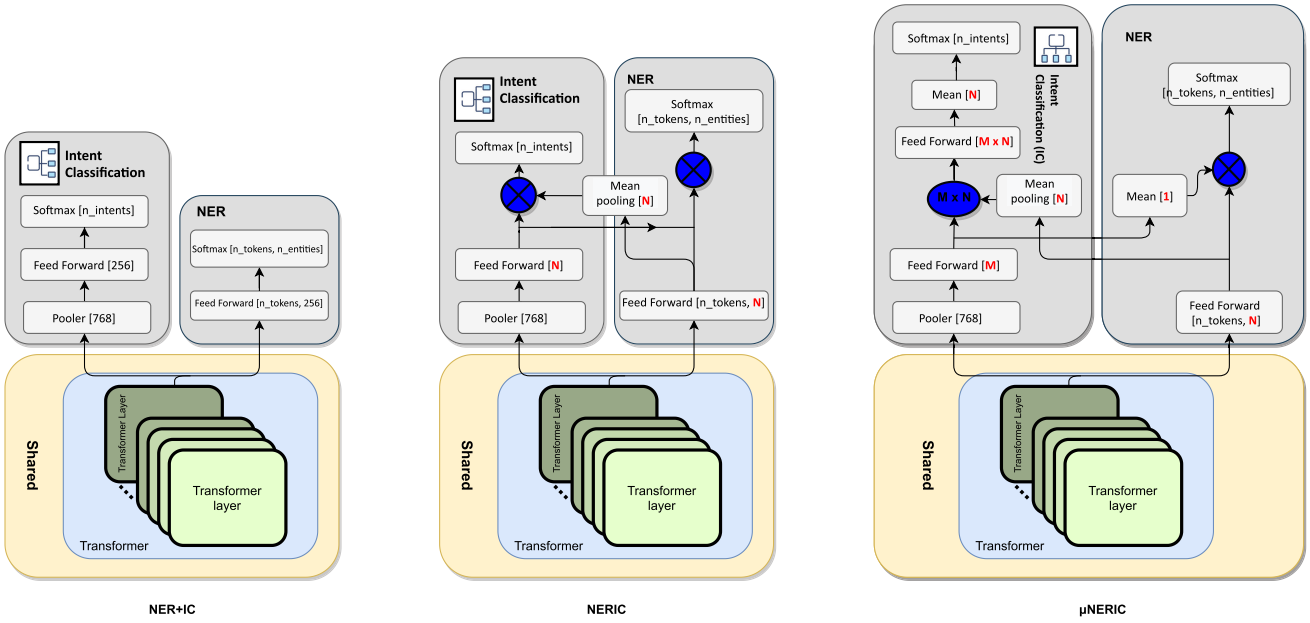


**FIGURE 1.** The Joint models for intent detection and Named Entity Recognition. NER+IC: Two tasks only share the transformer layer. NERIC: Two tasks share parameters through Hadamard product. *µNERIC*: Two tasks share information through Matrix Multiplication.

their loss function, they used Large Margin Cosine Loss arguing that it maximizes the decision margin. With this model, they were trying to solve the problem of unseen intents. The problem with RNN-based models either in the form of LSTM or GRU is that the gradients are vanishing through the process of encoding the input. They become smaller and smaller, so the parameter updates become insignificant which means no real learning is done.

Transformer-based models were introduced to address the problem of vanishing gradients in RNN-based models. Transformers [22] are not sequential so they do not suffer from the problem of vanishing gradients. They also benefit from self-attention and positional embeddings, which help to capture long-range dependencies. A recent work by Ren and Xue [36] proposed using transformer-based models for encoding the input. Specifically, they train triple samples, namely, an anchor sample, a positive sample from the class, and a negative sample from a different class. Then they combined convolutional and BERT encoding of each sample and mapped them to the Euclidean space with Siamese shared weights. They tried to minimize an intermediate loss of anchor-positive distance minus the anchor negative distance.

Capturing sequential information is one of the key components for NER too; hence various RNN, Bi-LSTM, CNN and Bi-GRU have been employed in different studies due to their ability to encode such information. Mesnil *et al.* [8] compared different types of RNN, including Elman-type and Jordan-type networks. Both types were constructed with a 3-word context window. Yao *et al.* [37] used LSTM cells, containing a gate to forget unnecessary information, along with regression to model dependencies. Their regression model took a non-normalized score before applying softmax to avoid label bias. Even though they showed improvement over CRF models, they were not able to beat previous deep learning based models.

All the models above, which tackle IC and NER independently, suffer from a major issue. They do not consider the inter-dependencies between the intent of a sentence and its containing entities. For example, when a customer says, "I would like to downgrade to essential", if a NER model knows that the intent is "*downgrade*", there is a higher chance for it to be able to recognize "*essential*" as a plan name. Unified models were proposed as an alternative to address the problem [18], [35], [38], [39].

### B. UNIFIED MODELS

The earliest work on unified models used a triangular-chain CRF, which outperformed other models tackling the two sub-tasks in a pipeline [38]. Other early studies also used statistical models such as maximum entropy model for IC and CRF for NER [40], and a multi-layer Hidden Markov Model [41]. These models, like any other models based on traditional machine learning, suffer from the laborious task of feature engineering and over-relying on domain experts knowledge. This problem was obviated by applying deep learning models, where the model itself extracts the features.

Deep learning models have been employed to address the two tasks simultaneously. Recursive Neural Networks (RecNNs) (different from recurrent neural networks (RNN)) was used in Guo *et al.* [39] where the RecNNs worked over the constituency parse tree of the utterance. In this case the leaves corresponded to the words, which were in turn represented by word vectors. A RecNN was applied to each node in the

**TABLE 2. Performance comparison on ATIS and SNIPS datasets. Model names written in bold refer to ours.**

| Model | ATIS dataset | | SNIPS dataset | |
|---|---|---|---|---|
| | Slot | Intent | Slot | Intent |
| Joint Seq [17] | 94.3% | 92.6% | 87.3% | 96.9% |
| Attention-based [18] | 94.2% | 91.10% | 87.8% | 96.7% |
| Slot-Gated [26] | 95.42% | 95.41% | 89.27% | 96.86% |
| SF-ID (w/ CRF) [34] | 95.80% | 97.09% | 92.23% | 97.29% |
| Stack-Propagation [35] | 95.90% | 96.90% | 94.20% | 98.00% |
| **our Joint RoBERTa-base NER+IC** | 92.45 % | 98.4% | 95.47 % | 98.85% |
| **ourJoint RoBERTa-base NERIC** | 91.84% | 98.51% | 95.38 % | **99.0**% |
| **our Joint RoBERTa-base $\mu NERIC$** | 94.77% | **98.63**% | **95.62**% | 98.71 % |
| **our RoBERTa-base (Single task: IC)** | NA | 97.6% | NA | 98.5% |

**TABLE 3. Performance comparison on ATIS and SNIPS datasets using different embeddings.**

| Embedding | Model | ATIS dataset | | SNIPS dataset | |
|---|---|---|---|---|---|
| | | Slot | Intent | Slot | Intent |
| roberta-base | NER+IC | 92.45 % | 98.4% | 95.47 % | 98.85% |
| | NERIC | 91.84% | 98.51% | 95.38 % | 99.0% |
| | $\mu NERIC$ | 94.77% | 98.63% | 95.62% | 98.71 % |
| bert-base-uncased | NER+IC | 91.66 % | 98.97% | 93.9 % | 98.85% |
| | NERIC | 93.97 % | 98.63% | 93.59 % | 98.57% |
| | $\mu NERIC$ | 93.81% | 98.74% | 94.13 % | 98.57 % |
| bert-base-cased | NER+IC | 92.4% | 97.6% | 94.6 % | 99.00% |
| | NERIC | 95.7% | 98.51% | 95.32 % | 99.00% |
| | $\mu NERIC$ | 94.8 % | 97.3% | 94.1 % | **99.4** % |
| roberta-large | NER+IC | 94.54% | 98.86% | 95.77 % | 98.71 % |
| | NERIC | 94.08% | 98.97% | 94.5 % | 99.00 % |
| | $\mu NERIC$ | 92.62 % | 98.51% | 95.05 % | 98.71 % |
| bert-large-uncased | NER+IC | 92.37% | 98.74% | 94.74 % | 98.71 % |
| | NERIC | 93.72% | **99.08**% | 94.48 % | 98.71 % |
| | $\mu NERIC$ | 95.21 % | 98.86% | 94.07 % | 98.85 % |
| bert-large-cased | NER+IC | 95.77% | 98.40% | **95.96** % | 99.14 % |
| | NERIC | 95.67% | 98.86% | 93.65 % | 98.85 % |
| | $\mu NERIC$ | **95.92** % | 98.97% | 94.88 % | 98.85 % |

tree recursively from bottom to top, computing the node's state. Individual slot label classifiers were applied to each leaf by combining: (i) each node's word vectors; (ii) the node's neighbours; (iii), and the state vectors from the leaf to the root. Finally, the state at the root was passed to the IC, and a combined loss over the slots and intents was back-propagated. Their model was among the first models to tackle the two task simultaneously. However, they did not improve upon the previous models, as their model was not able to encode the sequential information and the dependencies between the slots.

In order to take the dependencies between the slots into account, later works focused on exploring the power of RNNs in capturing sequential information. For instance, a multi-domain joint semantic frame parsing using bi-directional RNN-LSTM was used by Hakkani-Tür et al. [17]. In their model, features representing the tokens were passed in temporal sequences to RNN units with a hidden state. Intermediate hidden states and the final hidden state were used for slot labeling and intent prediction. Despite the RNNs models' success in encoding the sequences, these models were criticized for sharing the information between the tasks implicitly and only through back propagation of joint loss [1]. They also suffer from information loss in longer sequences due to their sequential nature. This paved the way for models based on attention mechanism which are not sequential and process the whole sequence at once.

Liu and Lane [18] used attention for slot and intent predictions. As they argue, in bidirectional RNN for sequence labeling, the hidden state at each time step carries information of the whole sequence, but information may gradually lose along the forward and backward propagation. Thus, when making slot label prediction, instead of only utilizing the aligned hidden state $h_i$ at each step, they use an additional context vector $c_i$ for additional supporting information, especially in the case of longer-range dependencies that is not usually being fully captured by the hidden state.

Later, Goo *et al.* [26] presented a method benefiting from more explicit attention. It did so by taking the word vectors in sequence and using a different learned weighted sum of the intermediate states of the BiLSTM for each slot prediction and the final state for intent detection. They managed to improve upon the results achieved by Liu and Lane [18].

Attention-based models proved successful in handling the long-range dependencies, however, they were not able to solve the long-lasting problem in the field, which is small-scale human-labeled training data. This results in poor generalization capability, especially for rare words. Transformer-based language models, e.g. BERT, facilitate pre-training deep bidirectional representations on large-scale unlabeled corpora, and has created state-of-the-art models for a wide variety of natural language processing tasks after simple fine-tuning [42]. Since the advent of BERT [43], several researchers have applied it in their unified models. For example, an intent2slot architecture was used in Qin *et al.* [35] with BERT encoding and stack propagation to jointly learn the intents and entities. In their model, the information about the intent detection was used to inform the entity recognition.

The transformer architecture [22], which captures global dependencies through multi-head attention, was also utilized in some papers (see for instance [23], [24], and references therein). In Do and Gaspers [23], the transformer architecture was used to create the contextual embedding of the word tokens and then the attention was applied between them to inform the intent detection sub-task. Their model achieved better results than earlier work in the literature.

Zhang and Wang [24], passed word embeddings to a three-level transformer layer, and then an output was extracted to inform IC and a token level output was passed to a CRF for NER. Unified models can generally be divided into intent2slot and slot2intent models. The inten2slot models use intent information as part of slot prediction and slot2intent models do the opposite. An intent2slot model with BERT encoding was used in Qin *et al.* [35] and Wang *et al.* [44] to learn intent and slots jointly. Haihong *et al.* [34] proposed an intent2slot and slot2inent model within a BiLSTM encoder. In this, a weighted sum of intermediate states for each step (the slot contexts) feeds a slot sub-net and the weighted final hidden state (the intent context) feeds an intent sub-net.

In our work, we propose a novel unified intent2slot and slot2intent approach. We perform IC and NER simultaneously using a task-specific layer on top of a transformer-based language model. We employ three different architectures to

share the information between the two sub-tasks and show that our model outperforms the state-of-the-art results on both SNIPS and ATIS datasets. Qualitative results verify the superiority of our unified model over the single-task model in handling the ambiguous intents.

## III. METHODOLOGY

In this section, we formally define IC and NER. We also present our multi-task framework and each respective model.

### A. IC AND NER

Let $V$ be a vocabulary and $w \in V$ a word. We represent each sentence in our collection as a sequence of words $\boldsymbol{s} = (w_1, w_2, \cdots, w_n)$. In NER, the goal is to formulate the problem as a sequence labeling task that maps an input word sequence $\boldsymbol{s}$ to their corresponding label sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$. Similarly, intent detection can be treated as a classification problem to decide the intent label of the whole sequence $\boldsymbol{s}$ as a single $y'$.

### B. PROPOSED APPROACH

Here, we propose three multi-task transformer-based models capable of performing IC and NER simultaneously (see Figure 1). Each of our models has a different architecture through which the two tasks share the underlying representations. Each sentence's representation is produced using a transformer-based language model, which includes contextual information. This part is common among all of our models, however, there are differences in the way they share information at the higher layers. Both IC and NER are optimized simultaneously via a joint learning scheme. The following sections explain NER+IC, NERIC and *µNERIC* respectively.

### C. MODELS

We propose a simple yet effective encoder-classifier architecture based on transformer-based language models. Specifically, we use a transformer-based language model, (e.g. RoBERTa [45], BERT), as our encoder and then add classification layers on top. Any transformer-based language model could be used, however, we recommend RoBERTa, as it produces better results in most cases (see Table 3). RoBERTa builds on BERT's language masking strategy and modifies key hyper-parameters in BERT, including removing BERT's next-sentence prediction objective, and training with much larger mini-batches and learning rates. RoBERTa was also trained on much more data than BERT, for a longer amount of time. This allows RoBERTa representations to generalize even better to downstream tasks compared to BERT. For each model, we add task-specific layers to share the parameters between the models in three different ways.

As for NER+IC, for the IC channel, first, we pass the sentences through an encoder, which is a transformer-based language model, to get the representations ($h_{cls}$) via the pooler layer. Then we apply a feed-forward layer with 256 nodes; this layer helps the model select important features. More layers could be added at this stage, but we do not advise that, as it could lead to over-fitting, especially, given the small size of the dataset. We add a softmax layer on top to get the probability distribution for the classes (n_intents in Figure 1 refers to the number of classes/intents). We fine-tune the entire model, including the pre-train model, using the training set. For NER in NER+IC, we do not use the pooler layer as we need token-level representations ($h_j$). We just add a task-specific softmax layer on the top of RoBERTa. In this model, the only shared part between the two sub-tasks, i.e. IC and NER, is just the encoder. In other words, we denote the input sequence as $x = (x_{<cls>}, x_1, \ldots, x_l)$, where $l$ is the utterance length. Each word $x_j$ will be embedded into a vector, and the output can be formulated as $\boldsymbol{h} = (h_{<cls>}, h_1, \ldots, h_l)$. The *cls* token is the first token of the sequence when $\boldsymbol{h}$ is built with special tokens. This cls is used in the classification of a whole sequence, instead of per-token classification. Specifically, the IC and NER results are predicted using:

$$y' = \text{softmax}(W_i h_{\text{cls}} + b_i) \tag{1}$$

$$y_j = \text{softmax}(W_s h_j + b_s), \tag{2}$$

where $y'$ and $y_j$ denote intent label of the utterance and NER label for each token $j$, respectively. $W$ and $b$ are corresponding trainable parameters. $W_i$ and $b_i$ are the parameters of the IC part and $W_s$ and $b_s$ are the parameters of the NER part of the model.

For the NERIC, the IC is done by applying a softmax layer on top of the Hadamard product of the feed-forward layer from the IC and the mean-pooling from the NER channel. We use mean-pooling since according to the authors of RoBERTa, it yields the best results. The NER model also uses a softmax layer on the top of the Hadamard product of the feed-forward layer coming from the IC channel and the feed-forward layer of the NER channel itself. In other words, the IC and NER results for NERIC are calculated as follows, respectively:

$$y' = \text{softmax}((W_i h_{\text{cls}} + b_i) \circ h_{s\_mean}) \tag{3}$$

$$y_j = \text{softmax}((W_i h_{\text{cls}} + b_i) \circ (W_s h_j + b_s)), \tag{4}$$

where $h_{s\_mean}$ is the mean-pooling layer coming from the NER channel. This gives us a matrix with the same dimensions as the feed-forward layer in the IC channel allowing us to calculate the Hadamard product. This is the simplest way for the two sub-tasks to communicate and share information.

For the *µNERIC*, first, a matrix multiplication is carried out on the feed-forward layer of the IC and the mean-pooling coming from the NER model. Then a feed-forward layer is applied on the result followed by another mean-pooling layer. Finally, a softmax layer predicts the intent labels.

The NER part of *µNERIC*, first conducts a matrix multiplication on the feed-forward layer of the model itself and the one coming from the IC. Then a softmax layer is applied on top to predict NER labels. In other words, the IC and NER results for *µNERIC* are calculated as follows,

$$y' = \text{softmax}(W_{i2}((W_i h_{\text{cls}} + b_i) \times (W_s h_j + b_s)) + b_{i2}) \tag{5}$$

**TABLE 4. Examples of sentences.**

| Sentence# | Sentence text |
|---|---|
| 1 | what are the times for The Gingerbread Man |
| 2 | how many passengers can fly on a 757 |
| 3 | what is the seating capacity of the type of aircraft m80 |
| 4 | show me the airports serviced by tower air |

$$y_j = \text{softmax}((W_i h_{\text{cls}} + b_i) \times (W_s h_j + b_s)), \tag{6}$$

where $W_i$ and $b_i$ are the parameters of the first feed-forward layer, and $W_{i2}$ and $b_{i2}$ are the parameters of the second feed-forward layer in the IC part.

### D. JOINT LEARNING

To obtain both IC and NER jointly, we formulate the final joint objective as

$$\mathcal{L}_{joint} = W_i \mathcal{L}_i + (1 - W_i)\mathcal{L}_s, \tag{7}$$

where $W_i$ is a trainable parameter. $\mathcal{L}_i$ and $\mathcal{L}_s$ are Loss functions for IC and NER, respectively, which are calculated using,

$$\mathcal{L}_i, \mathcal{L}_s = -y.log(\hat{y}), \tag{8}$$

where y is the ground truth vector and $\hat{y}$ is the estimate.

Then we calculate the joint loss as described in Equation 7, which is the weighted sum of loss of IC and NER. The weight, i.e. $W_i$ is learned while training the model.

## IV. EXPERIMENTS AND RESULTS

In this section, we compare our proposed model against different state-of-the-art methods on benchmark datasets. The setup we use for these datasets is rather popular (see for instance [17], [18], [26], [34], [35], and references therein).

### A. DATASETS

The ATIS dataset [29] is the most widely used dataset in NLU research, containing audio recordings of people making flight reservations. Our setup is composed of 4,455 utterances for training, 878 for validation and 497 for testing from ATIS-2. There are in total 127 distinct slot labels and 18 different intent types.

The SNIPS NLU dataset [46] contains 13,084 utterances in training, 700 in development and 700 in test sets. In training there are 72 slot labels and a vocabulary size of 11,241 words. Unlike ATIS, SNIPS covers different domains - weather, restaurants and entertainment in 7 balanced intent classes.

### B. BASELINE MODELS

We compare our models against the state-of-the-art methods and a variant of our proposed methods in which only the task of intent classification is attempted (single-task model). We summarise all methods compared against below.

- Joint Seq [17]. This bi-directional RNN-LSTM model was proposed for joint modeling of slot filling, intent determination, and domain classification.
- Attention-based [18]. This bi-directional RNN model was introduced to compensate for the inadequacy of regular RNN models in taking long-range dependencies into account. In bi-directional RNN for sequence labeling, the hidden state at each time step carries information of the whole sequence. However, information may gradually erode along the forward and backward propagation. Thus, this Attention-based model uses a context vector along with the hidden state to capture more contextual information, especially in the case of longer range dependencies.
- Slot-Gated [26]. This model introduces an additional gate that leverages intent context vector for modeling slot-intent relationships, in order to improve slot filling performance.
- SF-ID (w/CRF) [34]. This is a slot2intent and intent2slot model using bi-directional interrelated model for joint intent detection and slot filling was proposed.
- Stack-Propagation [35]. This joint model with stack-propagation was developed to directly use the intent information as input for slot filling.
- Single-task IC. In this we used only the IC part of NER+IC to predict the intents.

### C. EXPERIMENTAL SETTINGS

We use roberta-base model to create representations, which gives us a 768-dimensional vector. For our feed-forward layer we choose 256 nodes and 0.4 for dropout rate. We set the batch size to 64 and learning rate to 0.0001. We conduct our experiments on ATIS and SNIPS, reporting our results and those obtained by key texts in the literature (see Hakkani-Tür *et al.* [17], Liu and Lane [18], Goo *et al.* [26], Haihong *et al.* [34], and Qin *et al.* [35]).

We use Adam [47] to optimize the parameters in our model and adopted its suggested hyper-parameters for optimization. For each experiment, we select the model which works the best on the development set, and then evaluate it on the test set.

### D. QUANTITATIVE RESULTS AND DISCUSSION

Following Hakkani-Tür *et al.* [17], Liu and Lane [18], Goo *et al.* [26], Haihong *et al.* [34], and Qin *et al.* [35], we evaluate the performance of our NLU engine for IC using accuracy, and for NER using F1 score. It is worth mentioning that for benchmarking the models performance on NER, we use CoNLL[1] scheme, which is one of the most famous benchmarks for NER; it uses an strict definition for recall and precision to define the F1 score. Table 2 summarizes the models' performance on ATIS and SNIPS datasets.

Table 2 shows our models significantly outperform all the baselines by a noticeable margin, achieving state-of-the-art performance for IC and NER on SNIPS, and for IC on ATIS. On SNIPS, compared to the best prior joint model, *stack-propagation*, we achieve 1.42% improvement

[1]Conference on Computational Natural Language Learning.

**TABLE 5.** Examples of single-task and joint model predictions.

| Sentence | Correct Intent | Single-Task Model | Joint Model | Named Entities detected |
|---|---|---|---|---|
| 1 | SearchScreeningEvent | GetWeather | SearchScreeningEvent | 'B-movie_name', 'I-movie_name' |
| 2 | atis_capacity | atis_quantity | atis_capacity | 'B-aircraft_code' |
| 3 | atis_capacity | atis_aircraft | atis_capacity | 'B-aircraft_code' |
| 4 | atis_airport | atis_city | atis_airport | 'B-airline_name' |

in NER (F1), and 1% improvement in intent (accuracy). On ATIS, we achieve 1.54% improvement in intent accuracy compared to the best performing baseline, *SF-ID (w/CRF)*. This indicates the effectiveness of our simple architecture. We can attribute these results to the fact that our framework is good at capturing the correlation between the intents and named entities. Our qualitative results also verify our assumptions that NER information can be used for guiding intent detection (see Section IV-E).

We also experiment with other versions of transformer-based language models. Table 3 summarizes there results, showing *bert-large (cased/uncased)* performs better than any other models except for the IC on SNIPS where *bert-base-cased* performs slightly better (0.24%). These results ratify the fact that larger models are able to encode more information. In particular, *bert-large-cased* outperforms all models for NER task improving state-of-the-art for NER by 0.02% on ATIS and 1.76% on SNIPS. This is also in line with what was reported in BERT original paper in which *bert-large-cased* outperformed *bert-base-cased* in NER. As for IC, the NERIC built on *bert-large-uncased* and the $\mu$*NERIC* built on *bert-base-uncased* improve state-of-the-art by 2.18% on ATIS and 1.4% on SNIPS respectively.

We have seen all these significant improvements on two publicly available datasets. However, we would like to know the reason for the improvement. To this end, we compare the results achieved by the joint model to those achieved by the single-task model, which only addresses the IC task. The quantitative results in Table 2 show the superiority of our joint models to the single-task model. The joint models achieve higher accuracy scores, with $\mu$*NERIC* being the best performing model, with 1.03% and 0.5% higher scores on ATIS and SNIPS, respectively compared to the single-task model. Moreover, examples from datasets demonstrate that how the IC in our joint model is able to predict the correct intent where the single-task model fails to do so. The following section discusses this in more detail.

### E. QUALITATIVE RESULTS AND DISCUSSION

Here, we present some examples from the datasets where the joint model predicts the intent correctly, while the single-task model is incapable of doing so. Table 4 shows the list of the sentences and Table 5 demonstrates the predictions of the IC model both in the single-task and joint models.

For example, the intent for sentence 1, "*what are the times for The Gingerbread Man*", is predicted as "**GetWeather**" by the single-task model while the correct intent is "**SearchScreeningEvent**". The IC in joint-model is able to predict this intent correctly, as it is correctly informed by the NER part of the model that "**The Gingerbread Man**" is the name of a movie.

As for the sentence 2, *"how many passengers can fly on a 757"*, the single-task model is not able to predict the correct intent, which is "**atis_capacity**"; the models is probably misled by the mention of a number, i.e. 757. On the other hand, the unified model successfully predicts the intent. This could be due to the fact that the NER part of the model has identified *757* as an **aircraft_code**.

In sentence 3, "*what is the seating capacity of the type of aircraft m80"*, the user is asking about the capacity of the aircraft, i.e. "**atis_capacity**". This has correctly been identified by the joint model, as it has access to another piece of information, provided by the NER part of the model. The NER model has recognised a mention of "**aircraft_code** " in the sentence. The single-task model's failure can be linked to the model's lack of access to such information and relying only on keywords like *aircraft*.

Finally in sentence 4, "*show me the airports serviced by tower air*", the joint model is able to predict the correct intent, which is "**atis_airport**"; however, the single-task model falls short by classifying the intent as "**atis_city**". This might be linked to the information that the IC part of the model receives from the NER part of it. In this example, *tower* was identified as "**airline_name**" by the NER model, which has led to the IC to choose "**atis_airport**" intent over "**atis_city**".

## V. CONCLUSION

In this paper, we propose a simple yet effective unified model for IC and NER. Our transformer-based model enhances the interrelated connections between the intent and entities. The intent2slot and slot2intent interrelated model helps the two tasks enhance each other mutually. Our model outperforms the baselines on two public datasets by a noticeable margin. These positive results are further reinforced by our qualitative analysis, which shows the effectiveness of our unified model as it is able to assign correct labels to ambiguous sentences whereas the single-task model fails to do so. In our unified model, the two tasks share information through simple Hadamard and matrix multiplication. In future, we would like to explore other ways of sharing parameters between the IC and NER channels.

## 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," Feb. 2021, *arXiv:2101.08091v3*.

[2] Y. Wang, Y. Shen, and H. Jin, "A Bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, Jun. 2018, pp. 309–314. [Online]. Available: https://www.aclweb.org/anthology/N18-2050

[3] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proc. ICML*, 2000, pp. 591–598.

[4] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proc. INTERSPEECH*, 2007, pp. 1–5.

[5] P. Haffner, G. Tur, and J. Wright, "Optimizing SVMs for complex call classification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, pp. 1–5.

[6] A. Moschitti, G. Riccardi, and C. Raymond, "Spoken language understanding with kernels for syntactic/semantic structures," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, Jan. 2008, pp. 183–188.

[7] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. INTERSPEECH*, 2013, pp. 3771–3775.

[8] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 530–539, Mar. 2015.

[9] A. Deoras and R. Sarikaya, "Deep belief network based semantic taggers for spoken language understanding," in *Proc. INTERSPEECH*, 2013, pp. 2713–2717.

[10] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 189–194.

[11] P. Haffner, G. Tur, and J. Wright, "Optimizing svms for complex call classification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Oct. 2003, pp. 1–4.

[12] G. Riccardi, A. Potamianos, and S. Narayanan, "Language model adaptation for spoken language systems," in *Proc. ICSLP*, vol. 98, Jan. 1998, pp. 2327–2330.

[13] Y. Ji, G. Haffari, and J. Eisenstein, "A latent variable recurrent neural network for discourse-driven language models," in *Proc. NAACL*, 2016, pp. 1–11.

[14] H. Hashemi, "Query intent detection using convolutional neural networks," in *Proc. Int. Conf. Web Search Data Mining, Workshop Query Understanding*, 2016.

[15] S. V. Ravuri and A. Stolcke, "Recurrent neural network and lstm models for lexical utterance classification," in *Proc. INTERSPEECH*, 2015, pp. 1–5.

[16] S. Shah and R. Siskind, "Multi-task learning of query intent and named entities using transfer learning," 2021, *arXiv:2105.03316*.

[17] D. Z. Hakkani-Tür, G. Tür, A. Çelikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Proc. INTERSPEECH*, 2016, pp. 1–5.

[18] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proc. INTERSPEECH*, 2016, pp. 1–5.

[19] X. Zhang and H. Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *Proc. IJCAI*, 2016, pp. 1–5.

[20] P. Xu and R. Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 78–83.

[21] M. Firdaus, S. Bhatnagar, A. Ekbal, and P. Bhattacharyya, "A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding," in *Proc. ICONIP*, 2018, pp. 647–658.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010.

[23] Q. Do and J. Gaspers, "Cross-lingual transfer learning with data selection for large-scale spoken language understanding," in *Proc. 9th Int. Joint Conf. Natural Lang. Process.* Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 1455–1460. [Online]. Available: https://aclanthology.org/D19-1153

[24] L. Zhang and H. Wang, "Using bidirectional transformer-CRF for spoken language understanding," in *Proc. CCF Int. Conf. Natural Lang.*, Sep. 2019, pp. 130–141.

[25] C. Li, Y. Zhao, and D. Yu, "Conditional joint model for spoken dialogue system," in *Proc. Int. Conf. Cogn. Comput.*, Jun. 2019, pp. 26–36.

[26] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 753–757.

[27] S. Yu, L. Shen, P. Zhu, and J. Chen, "ACJIS: A novel attentive cross approach for joint intent detection and slot filling," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–7.

[28] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[29] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Proc. Workshop Held Hidden Valley*, Pennsylvania, PA, USA, Jun. 1990, pp. 1–6. [Online]. Available: https://aclanthology.org/H90-1021

[30] A. Gorin, G. Riccardi, and J. Wright, "How may I help you?" *Speech Commun.*, vol. 23, no. 1, pp. 113–127, 1997.

[31] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 135–168, May 2000.

[32] C. Costello, R. Lin, V. Mruthyunjaya, B. Bolla, and C. Jankowski, "Multilayer ensembling techniques for multilingual intent classification," 2018, *arXiv:1806.07914*.

[33] T.-E. Lin and H. Xu, "Deep unknown intent detection with margin loss," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5491–5496. [Online]. Available: https://aclanthology.org/P19-1548

[34] H. E, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," 2019, *arXiv:1907.00390*.

[35] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proc. EMNLP/IJCNLP*, 2019, pp. 1–10.

[36] F. Ren and S. Xue, "Intention detection based on Siamese neural network with triplet loss," *IEEE Access*, vol. pp. 82242–82254, 2020.

[37] K. Yao, B. Peng, Y. Zhang, Y. Dong, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2014, pp. 189–194.

[38] M. Jeong and G. G. Lee, "Triangular-chain conditional random fields," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 7, pp. 1287–1302, Sep. 2008.

[39] D. Guo, G. Tür, W. Tau Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 554–559.

[40] Y.-Y. Wang, "Strategies for statistical spoken language understanding with small amount of data—An empirical study," in *Proc. INTERSPEECH*, 2010, pp. 1–4.

[41] A. Celikyilmaz and D. Hakkani-Tur, "A joint model for discovery of aspects in utterances," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*. Jeju Island, South Korea: Association for Computational Linguistics, Jul. 2012, pp. 330–338. [Online]. Available: https://aclanthology.org/P12-1035

[42] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," 2019, *arXiv:1902.10909*.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 1–16.

[44] C. Wang, Z. Huang, and M. Hu, "SASGBC: Improving sequence labeling performance for joint learning of slot filling and intent detection," in *Proc. 6th Int. Conf. Comput. Data Eng.*, Jan. 2020, pp. 29–33.

[45] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[46] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, Dec. 2015.

**ALBERTO BENAYAS** received the M.S. degree in computer science from the Complutense University of Madrid, Madrid, Spain, in 2008, and the B.S. degree in economy from UNED, Madrid, in 2015. He is currently pursuing the Ph.D. degree in artificial intelligence with the University of Alcala, Madrid.

He worked as a Software Engineer at Indra Systems, from 2008 to 2013. Then, he worked as a Software Engineer at the United Nations, from 2013 to 2017. From 2017 to 2020, he worked as a Data Scientist (as a Freelancer). Since 2020, he has been working as a Data Scientist at EPAM Systems, Madrid. His research interests include conversational mining, natural language processing, and deep learning.

**REYHANEH HASHEMPOUR** received the bachelor's degree in computer engineering, the master's degree in applied linguistics, and the double master's degree in natural language processing (NLP). She is currently pursuing the Ph.D. degree in NLP with the University of Essex. She is working as a full-time Data Scientist at BT.

**DAMIAN RUMBLE** received the Ph.D. degree in observational astrophysics from the University of Exeter, in 2016. He subsequently retrained as a Data Scientist, receiving a S2DS Fellowship, in 2016. Since then, he has been working at the Post Office, Aviva, Aviva Asia, and finally at BT, where he is the Current Head of consumer data science.

**SHOAIB JAMEEL** received the Ph.D. degree from The Chinese University of Hong Kong. He is currently a Lecturer in computer science and artificial intelligence at the School of Computer Science and Electronic Engineering, University of Essex, U.K. He works with various technology startups in the U.K., spearheading their technical sphere, where his research outputs are directly applied to their production systems. His works have appeared in various prestigious conferences and journals, such as SIGIR, AAAI, ACL, IJCAI, and *TOIS*. His research interests include text mining, natural language processing, and computer vision. He is a fellow of the Higher Education Academy.

**RENATO CORDEIRO DE AMORIM** received the Ph.D. degree in computer science from the Birkbeck University of London, in 2011. He is currently a Senior Lecturer in computer science and AI at the University of Essex. His research has been funded by Microsoft, the Royal Society, and Innovate U.K. He is an associate editor for journals published by Springer and Elsevier. He has authored a number of papers introducing novel methods following the unsupervised and semi-supervised learning frameworks, with applications in fields, such as security, biosignal processing, and data science in general. He received the Chikio Hayashi Award, in 2017.