A Survey of Joint Intent Detection and Slot Filling Models in Natural Language Understanding

HENRY WELD, XIAOQI HUANG, SIQU LONG, JOSIAH POON, and SOYEON CAREN HAN, The University of Sydney, Australia

Intent classification, to identify the speaker's intention, and slot filling, to label each token with a semantic type, are critical tasks in natural language understanding. Traditionally the two tasks have been addressed independently. More recently joint models that address the two tasks together have achieved state-of-the-art performance for each task and have shown there exists a strong relationship between the two. In this survey, we bring the coverage of methods up to 2021 including the many applications of deep learning in the field. As well as a technological survey, we look at issues addressed in the joint task and the approaches designed to address these issues. We cover datasets, evaluation metrics, and experiment design and supply a summary of reported performance on the standard datasets.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Information extraction;**

Additional Key Words and Phrases: Intent detection, slot labelling, natural language understanding

ACM Reference format:
Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A Survey of Joint Intent Detection and Slot Filling Models in Natural Language Understanding. *ACM Comput. Surv.* 55, 8, Article 156 (December 2022), 38 pages.
https://doi.org/10.1145/3547138

1 INTRODUCTION

The efficacy of virtual assistants becomes more important as their popularity rises. Central to their performance is the ability for the electronic assistant to understand what the human user is saying to act, or reply, in a way that meaningfully satisfies the requester. The human-device interface may be text based but is now most frequently voice and will probably in the near future include image or video. To put the understanding of human utterances within a framework, within the **natural language processing (NLP)** stack lies **spoken language understanding (SLU)**. SLU starts with **automatic speech recognition (ASR)**, the task of taking the sound waves or images of expressed language, and transcribing to text. **Natural language understanding (NLU)** then takes the text and extracts the semantics for use in further processes—information gathering, question answering, dialogue management, request fulfilment, and so on.

Authors' address: H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, The University of Sydney, Building J12, 1 Cleveland Street, NSW 2006, Sydney, Australia; emails: {hwel4188, xhua7314, slon6753}@uni.sydney.edu.au, {Josiah.Poon, caren.han}@sydney.edu.au.

0360-0300/2022/12-ART156 $15.00
https://doi.org/10.1145/3547138

Table 1. An Example of an Utterance as a Semantic Frame with Domain, Intent, and IOB Slot Annotation [32]

query	find	recent	comedies	by	james	cameron
slots	O	B-date	B-genre	O	B-director	I-director
intent	find_movie					
domain	movies					

The concept of a hierarchical semantic frame has developed to represent the levels of meaning within spoken utterances [40]. At the highest level is a domain, then intent, and then slots. The domain is the area of information with which the utterance is concerned. The intent (a.k.a. goal in early papers) is the speaker's desired outcome from the utterance. The slots are the types of the words or spans of words in the utterance that contain semantic information relevant to the fulfilment of the intent. An example is given in Table 1 for the domain *movies*. Within this domain the example has intent *find_movie* and the individual tokens are labelled with their slot tag using the inside-outside-beginning (IOB) tagging format. The NLU task is thus the extraction of the semantic frame elements from the utterance. NLU is central to devices that desire a linguistic interface with humans: conversational agents, instruction in vehicles (driverless or otherwise), **Internet of Things (IoT)**, virtual assistants, online helpdesks/chatbots, robot instruction, and so on. Improving the quality of the semantic detection will improve the quality of the experience for the user, and from here NLU draws its importance and popularity as a research topic.

In many datasets, and indeed real-world applications, the domain is limited; it is concerned only with hotel bookings or air flight information, for example. In these cases, the domain level is generally not part the analysis. However, in wider ranging applications, for example the SNIPS dataset discussed later or the manifold personal voice assistants that are expected to field requests from various domains, inclusion of the domain detection in the problem can lead to better results. This leaves us with intent and slot identification. What does the human user want from the communication, and what semantic entities carry the details? The two sub-tasks are known as intent detection and slot filling. The latter may be a misnomer as the task is more correctly slot labelling or slot tagging. Slot filling is more precisely giving the slot a value of a type matching the label. For example, a slot labelled "B-city" could be filled with the value "Sydney." Intent detection is usually approached as a supervised classification task, mapping the entire input sentence to an element of a finite set of classes. Slot filling seeks to attach a class or label to each of the tokens in the utterance, making it within the sequence labelling class of problems.

While early research looked at the tasks separately, or put them in a series pipeline, it was quickly noted that the slot labels present and the intent class should and do influence each other in ways that solving the two tasks simultaneously should garner better results for both tasks [40, 104]. A joint model that simultaneously addresses each sub-task must capture the joint distributions of intent and slot labels, with respect also to the words in the utterance, their local context, and the global context in the sentence. A joint model has the advantage over pipeline models that it is less susceptible to error propagation [12], and over separate models that there is a only a single model to train and fine-tune. A drawback is that a large annotated corpus is usually required [85], though this is also true of separate models. The model may also be relatively complicated and take time to train. It has also been observed that joint models may not generalise well to unseen data, due to the variety of natural language expressions of similar intent [23, 81, 123]. In real-world applications the domains and label sets may change over time.

In many ways the development of the field has followed a similar path to other areas of NLP, starting with classical (statistical or probabilistic) models [40]. Neural networks were applied as computing power increased. In particular, due to the sequential nature of the slot labelling

sub-task, **recurrent neural networks (RNNs)** have been a technology frequently used in the field (from Reference [84] onward, see Tables 2–5). In more recent years the Transformer architecture has debuted to address issues like long-range dependency [91, 120]. As far as feature creation goes, convolution, word embeddings, and pre-trained language models have all been applied, amongst many other methods covered in Section 3.3. The use of external knowledge bases has been observed in more recent papers [101]). The most regularly used datasets are two freely available sets—the **Air Travel Information System (ATIS)** and SNIPS. A common experiment is implied by the literature, from which the reported results are compared in this survey. One of the aims of this survey is to address further standardisation, in terms of the parameters of the experiment and the evaluation metrics used. The approaches to the joint task have been manifold and have shown excellent results in standard supervised training/test experiments. As new techniques make what may appear to be incremental increases to the state of the art it is perhaps time to recast the measures of success in the field. Rather than just developing new, more challenging annotated datasets, increasingly important must be the development of unstructured semantic detection in new domains.

The motivation for this survey is to take stock of the state of the field at the end of 2021 following a surge of ideas and approaches over recent years. We collect information on the approaches pursued so far and the issues encountered and addressed. With the survey completed we propose some future directions for the field. While surveying the field we consider three questions:
Q1: How do joint models achieve and balance the two aspects, intent classification and slot filling?
Q2: Have syntactic features been fully exploited or does semantics override this consideration?
Q3: Can models successfully trained on a supervised dataset from one domain be made more generalisable to different domains or languages or to unseen data?

1.1 Scope

The focus of this survey is on extraction of the intent and slots of single utterances. Papers on the following aspects will be reviewed but the information considered as ancillary only: (1) multi-domain datasets with annotated domain. Extraction of the domain is a classification task like intent detection, albeit less granular. Inclusion of the domain identification task may aid the two sub-tasks of interest and this will be mentioned; (2) dialogue action is the identification of the next action to be taken by a dialogue management system once intent and slots have been identified. In some cases dialogue action is a direct substitute for intent and in others a mapping is made from intent and slot labels to the action. We review papers that include a strong focus on intent and slot detection; (3) ASR. Some papers look at error propagation from ASR to intent or slot prediction. We do not consider this aspect.

1.2 Related Surveys

Reference [93] is a complete summary of the SLU field at the advent of the neural era (2011). Reference [99] concentrates on models that jointly address sub-tasks in dialogue systems, including NLU, dialogue management, and Natural Language Generation. Similarly, Reference [77] (pre-print only) places NLU as a step in a dialogue system preceding Dialogue Management and Response Generation and focuses on multilingual aspects. Goal-oriented conversational language understanding is the focus of Reference [92] and within that field they provide an excellent precursor to this survey, covering the state of the art to 2017 in the two sub-tasks and 2016 in the joint task. Reference [37] is a small overview of the separate and joint tasks. A good survey of intent detection methods up to 2018 is given in Reference [54] including multi-intent detection and evaluation methods. Tangentially related surveys include a survey of dialogue datasets available for research [80] and an overview of evaluation methods for dialogue systems [21].

Table 2. Historical Overview of Intent Detection Papers

Year	# papers	Feature engineering	Technologies
2011	3	Dependency parse	SVM, DBN, multi-layer NN, AdaBoost
2012	1	Bag-of-words	C4.5, RF, NB, KNN, Linear SVM
2013	1	n-gram	SVM, SVM-HMMs
2015	4	n-gram, word2vec	LSTM, RNN, ensemble, RF, clustering, SVM, AdaBoost, NN, J48, FFN
2016	1	CNN	RF
2018	7	GloVe, word2vec, character, grammar features, dependency parse, knowledge base, POS, CRF, Regex, PCFG-ML, fastText	(Bi)CNN/LSTM/GRU/RNN, ensemble, capsule network, attention, adversarial network, gradient reversal layer, SVM, Logistic regression, PPN, RF, Gaussian Naïve Bayes, softmax regression
2019	4	n-gram, character, word2vec, CNN, BiLSTM	BiLSTM/GRU, attention, Ridge, KNN, MLP, passive aggressive, RF, linear SVC, SGD, nearest centroid, multinominal & Bernoulli NB, K-means, CNN, density-based, local outlier factor
2020	1	BERT, word2vec, CNN	Siamese, triple loss
2021	13	keyword, USE, ConveRT, pseudo-labels	transformer, capsule

1.3 Structure of the Survey

The survey begins with a broad overview of the literature in Section 2. We then give a detailed description of the technological methods for the joint task, along with the issues addressed and solutions proposed, in Section 3. In Section 4, a survey of the datasets encountered takes place. In Sections 5 and 6, there is a description of the experiments and evaluation methods applied and a discussion of standardisation of these. A summary of the results achieved over the history of the field is given in Section 7. We finish with a review of the research questions in Section 8.

2 OVERVIEW OF THE LITERATURE

Intent classification is a form of text classification where the text is a single sentence that comes from a spoken or written utterance. Much effort has been made to construct features that encapsulate the sentence, both semantically and syntactically, and the words within it. These features have been passed to classifiers from the suite of classical and, from 2011, deep learning methods, as outlined in Table 2. Issues around ambiguity, shortness of sentences, treatment of out-of-vocabulary words, and emerging label sets are among those covered in the literature. Slot tagging (see Table 3) is framed as a sequence labelling problem and in early years drew from methods for statistically modelling the dependencies within sequences, like **conditional random fields (CRFs)** and **Hidden Markov models (HMMs)**. Around 2013, the strength of RNNs in this area had been observed and was applied to the task and developed over the ensuing years. Interestingly the use of CRFs returned, often as a post-RNN step, due to their efficacy at handling label dependency issues. As far as feature creation goes the general goal of the task is to use the semantic information within the words and various context windows from small to long-range within the sentence. Attention is used as one approach for eliciting useful context. Slot tagging has experimented with external knowledge base.

Table 3. Historical Overview of Slot Labelling Papers

Year	# papers	Feature engineering	Technologies
2010	1	Neural network, observation feature vector	Deep learning, CRF
2012	1	n-gram, K-DCN	Kernel learning, deep learning, DCN, log-linear
2013	3	Discriminative embedding, dependency parse, named entity, POS, RNNLM, bag-of-words	DBN, RNN, RNN-LM
2014	2	RNN, lexicon feature	CRF, LSTM, regression model, deep learning
2015	3	Word, named entity	RNN, sampling approach, external memory
2016	3	Word, RNN, CNN, context window	BiRNN, attention, LSTM, encoder-labeler, CNN
2017	1	Word	(Bi)LSTM, encoder-decoder, focus mechanism, entity position-aware attention
2018	5	BiLSTM, word, character, CNN, delexicalisation	CRF, MTL, segment tagging, NER, BiLSTM, attention, DNN, reinforcement learning, GRU, pointer network
2019	6	Word, character, web-data, context information, expert feedback, GloVe, POS, BERT	BiLSTM/GRU, different knowledge sources, context gate, MTL, CNN
2020	2	ResTDNN	Prior knowledge driven label embedding, CRF, TDNN, RNN
2021	16	GCN, syntax embedding	cross-domain transfer, machine comprehension, decoupled local/global context

Methods used by both sub-tasks to extend their features include looking at metadata from the data collection. Multi-task learning has also been used by both tasks to look for synergistic learning from other related tasks. Of course, the joint task itself is an example of this synergistic approach. Both tasks have also considered methods for transfer learning to other languages and to data with new, unseen tag sets. The two earliest papers (2008–2009) addressing the joint task drew methods from classical NLP [40, 60]. Features were constructed from words, n-grams, and suffixes or from a semantic parsing of the utterances. A CRF or a **support vector machine (SVM)** was used for the analysis. In 2013, the first neural network was used though it really just constructed **convolutional neural network (CNN)** features for use in the CRF model from 2008 [108]. In 2014, a **recursive neural network (RecNN)**, which works over trees, was applied to the dependency parse of the utterances [29]. In 2015, the first completely neural network was devised, using a RNN (different to a recursive neural network) embedding of words, CNN sentences, and a **feed forward network (FFN)** for the analysis [84].

By 2016, the RNN encoder-decoder architecture had been found to be useful for **sequence-to-sequence (seq2seq)** tasks and started to make its impact in the joint task [32, 53, 122, 127]. Unidirectional and bidirectional **Long Short Term Memory (LSTM)** and **Gated Recurrent Unit (GRU)** cells were tested within circuits. Attention made its first appearance [52]. On the input feature side knowledge-guided structural attention networks (K-SAN) graphs were used as a knowl-

Table 4. Historical Overview of Joint Task Papers

Year	# papers	Feature engineering	Technologies
2008	1	words/n-grams/suffixes	CRF
2009	1	semantic tree	SVM
2013	1	CNN	CRF
2014	1	dependency parse	RecNN (diff to RNN)
2015	1	RNN words, CNN sentence, bag-of-words	MLP
2016	6	RNN, K-SAN	(Bi)LSTM/GRU, encoder-decoder RNN, attention
2017	4	character, word, CNN	BiLSTM
2018	18	word2vec, GloVe, ELMo, CNN, attention sentence	BiLSTM/GRU, encoder-decoder RNN, capsule NN, Bi-directional
2019	29	BERT, GloVe, character, knowledge base (tuples), delexicalisation	memory NN, transformer, CRF, attention, Bi-directional
2020	10	BERT, Graph embedding	Graph S-LSTM, Bi-directional, GCN, capsule
2021	13	mBERT, syntax GCN, sentiment, transformer	Bi-directional, transformer adaptations

edge base [13]. In 2017, the field appeared to stay progress, with only character embedding being added to the input features [42] and no improvement of performance results on the major datasets. Perhaps though, researchers were working on the many developments that exploded in 2018. Word embeddings were introduced: word2vec, Global Vectors (GloVe), and Embeddings From Language Models (ELMo) [85]. The circuits were still largely RNN based. For new architectures a capsule neural network [117] and bi-directional circuits were introduced [102]. Here bi-directional refers to explicit influence paths through the circuit: intent2slot refers to intent information being used as part of slot prediction and slot2intent the opposite, slot information being used as part of intent prediction.

In 2019, Bidirectional Encoder Representations from Transformers (BERT) debuted as a contextual Transformer-based word embedding technology and ELMo (a contextual LSTM-based model) fell away [11, 122]. More knowledge bases were used as input features [101]. Work on pre-processing the datasets included delexicalisation [75], augmentation [17], and sparse word embeddings using a lasso method [82]. In architecture, RNN and attention continued to be used and CRF made a return to handle label dependency issues [26]. Newly applied architectures included the Transformer [120] and memory neural networks [55]. The indications from 2020 are that graph embeddings are being used more to capture slot-intent and word-slot-intent relationships [89, 119]. In 2021, researchers looked more deeply at Transformer models in the context of the joint task. They addressed the architecture's agnosticism to the importance of global versus local context and the weaker approach to sequence via positional encoding [14, 39, 70, 126]. Multilingual semantic frame parsing also rose in importance [57].

3 JOINT INTENT AND SLOT MODELS

The joint task marries the objectives of the two sub-tasks. As most papers point out there is a relationship between the slot labels we should expect to see conditional on the intent and vice versa. A statistical view of this is that a model needs to learn the joint distributions of intent and slot labels. The model should also pay regard to the distributions of slot labels within utterances, and one would expect to inherit approaches to label dependency from the slot-labelling sub-task.

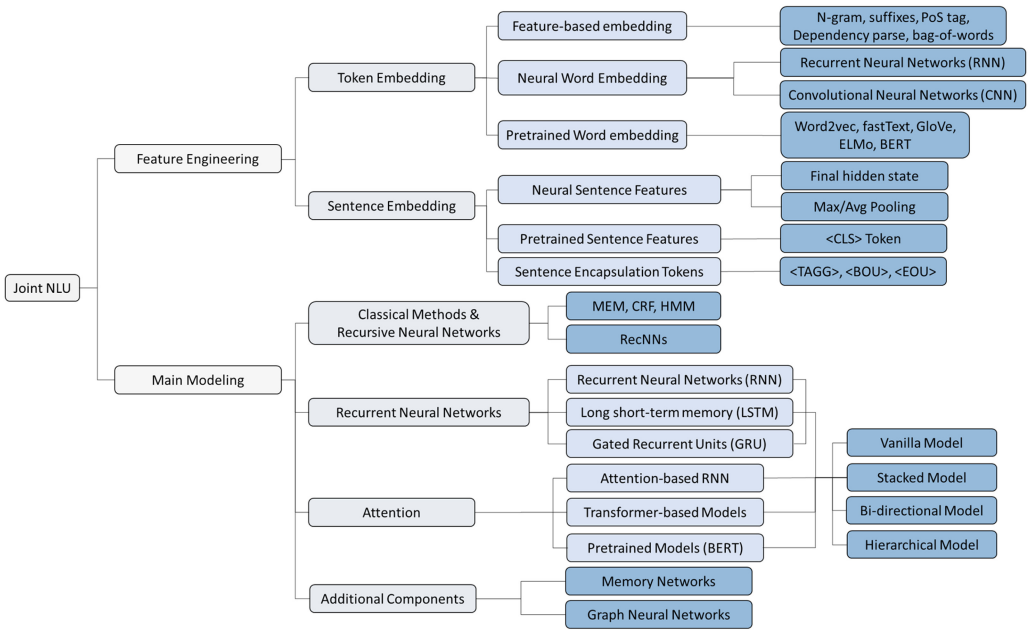

Fig. 1. Summary of technological approaches in joint NLU.

Approaches to the joint task range from implicit learning of the distribution through explicit learning of the conditional distribution of slot labels over the intent label, and vice versa, to fully explicit learning of the full joint distribution.

3.1 Major Areas of Research

Research in the joint task has largely come from the personal assistant or chatbot fields. Chatbots are usually task oriented within a single domain, while the personal assistant may be single or multi-domain. Other areas to contribute papers are IoT instruction, robotic instruction (there is also a different concept of intent in robotics to describe what action the robot is attempting), and in-vehicle dialogue for driverless vehicles. These areas also need to filter out utterances not applied to the device. Researchers have also drawn data from question answering systems, for example Reference [122] annotated a Chinese question dataset from Baidu Knows.

3.2 Overview of Technological Approaches

A summary of the technological approaches in joint NLU can be found in Figure 1. Detailed descriptions for each approach are found in this section.

3.2.1 Classical Methods. The earliest work on the joint task used a tri-level CRF with the three layers being token features, slot labels, and intent labels [40]. This architecture performed better than performing the two sub-tasks in a pipeline. Other early statistical models used a **maximum entropy model (MEM)** for intent and a CRF for slot labelling [104] and a multilayer HMM [9].

3.2.2 Recursive Neural Networks. The first neural model to address the joint task used RecNNs (different to recurrent neural networks) [29]. RecNNs work over trees, in this case the constituency parse tree of the utterance, with leaves corresponding to the words (represented by word vectors). A neural network is applied at each node of the tree, recursively upwards to the root, computing a

state for each node. At each node the states from children nodes are combined with a weight vector representing the node's syntactic type. Individual slot label classifiers are applied to each leaf using a combination of the word vectors of itself and its neighbours and the state vectors along the path from the leaf to the root. The state at the root is passed to an intent classifier. A combined loss over the slots and intent (and domain) is back-propagated. An optional post-processing, Viterbi decoded Markov layer, is applied to the slots. Results were close to, but below, the then state of the art for the tasks treated separately.

3.2.3 Recurrent Neural Networks. In 2016, the power of the RNN circuit for seq2seq tasks was explored in multiple papers. Features representing tokens are passed in temporal sequence to RNN units that have a hidden state. Intermediate hidden states may be used for slot labelling. The final hidden state is an embedding of the entire utterance and may be used for intent prediction. Bidirectional RNNs (BiRNNs, including BiLSTMs and BiGRUs), where the input sequence is passed in in both forward and backward directions, address issues with unidirectional capturing of context. The classic encoder-decoder, which produces a sequential output, is the most commonly used architecture. A typical example of the encoder-decoder RNN model for joint intent detection and slot filling is illustrated in Figure 2 in Reference [52]: The bidirectional RNN encoder first encodes the utterance into a sequence of hidden states, from which the last hidden state is used for intent classification and to initialise the decoder RNN state for generating slot labels. The difference among the three sub-figures in that figure is whether to use (a) context representation c_i only, which is derived as the attended representation over the encoded utterance sequence; (b) encoder hidden state h_i only; or (c) both as the input for intent detection and slot filling, mainly to explore ways of better aligning the encoded utterance with the slot label decoding process. Issues with the original RNN cells (vanishing gradient, long term dependency) are addressed by LSTM and GRU cells.

Other architectures include (a joint loss is back-propagated unless mentioned):

- a two layer LSTM with the top layer hidden states informing slot labelling and the first layer final state informing intent classification [127];
- the slot tagging task is softmax classifiers applied to the output of a simple BiLSTM using the concatenated hidden states. A special token is added to encapsulate the whole utterance for use in intent classification [32];
- a BiLSTM encoder decoder but with separate losses for intent and slot prediction [125];
- rather than seq2seq, perform a global slot prediction (learning the joint distribution) from a matrix of the hidden states to a matrix of slot tag probabilities for each word, intent is predicted from a sum of hidden states [42];
- use both a hierarchical (multi-layer) and a contextual (BiLSTM or LSTM) approach, investigating various combinations and using differing layers for intent and slot prediction [106];
- an ensemble using BiLSTM and BiGRU fed to separate multilayer perceptrons (MLPs) whose outputs are fused then projected and a softmax applied to predict intent and slots concurrently is proposed in Reference [23].

For RNNs the input is typically token feature by token feature in temporal sequence. Reference [32] compared that to using context windows with superior results. However, Reference [125] showed inferior results with context windows.

3.2.4 Attention. One critical observation made of many purely recurrent models is that the sharing of the information between the two sub-tasks is implicit. That is, while the sub-tasks are addressed jointly, it is often only through back-propagation of a joint loss. Attention is an obvious technique for forcing an interaction between information from the two sub-tasks, in a learned way. Some attention constructions may still be seen as an implicit way of sharing information,

but stronger methods start to force explicit learning. In early papers a basic concept of attention used was the weighted sum of Bi-RNN hidden states (i.e., the context representation c_i in Figure 2 in Reference [52]) as an input to slot and intent prediction. In this way, attention is used to connect the encoded utterance separately to the two sub-tasks that are implicitly jointly learned via the joint loss. Then Reference [28] used a stronger, more explicit attention. The base circuit is a BiLSTM taking word vectors in sequence and using a different learned weighted sum of the intermediate states of the BiLSTM for each slot prediction (the slot attention) and the final state for intent detection. As is shown in their major equations, Equations (1)–(3), the new addition is a slot gate g that takes the current slot attention vector c^S and combines it with the current intent attention vector c^I via an attention-like gating operation. The output of the slot gate then explicitly feeds the slot prediction to derive the slot label y_i^S. Here h_i and h_j denote the BiLSTM hidden states, v and W are a trainable vector and matrix in the gate, and W_{hy}^S is a trainable matrix in the slot prediction layer. This circuit is an early example of intent2slot, a path through the circuit where intent prediction information is also fed explicitly to the slot prediction element. Another variation on intent2slot is provided in Reference [49],

$$c_i^S = \sum_{j=1}^{T} \alpha_{i,j}^S h_j, c_i^I = \sum_{j=1}^{T} \alpha_{i,j}^I h_j, \tag{1}$$

$$g = \sum v \cdot \tanh\left(c_i^S + W \cdot c^I\right), \tag{2}$$

$$y_i^S = \text{softmax}\left(W_{hy}^S\left(h_i + c_i^S \cdot g\right)\right). \tag{3}$$

Reference [69] also use an intent2slot architecture but with BERT encoding and using stack propagation. Rather than a gate like Reference [28], the intent detection is conducted for each word level and itself directly feeds the slot filling. As is denoted in their major equations, Equations (4) and (5), at each slot decoding step i, the utterance input is not only the encoded BERT encoded representation e_i but also the intent prediction y_i^I at the current timestep, which indicates the explicit contribution from intent classification to the final slot labeling for y_i^S. Here h_{i-1}^S and y_{i-1}^S are the decoder hidden state and slot prediction from previous timestep while W_h^S is the trainable matrix in the slot prediction layer. The final intent is taken by vote over all the timesteps. Reference [98] also uses an intent2slot gate with BERT embeddings,

$$h_i^S = f\left(h_{i-1}^S, y_{i-1}^S, y_i^I \oplus e_i\right), \tag{4}$$

$$y_i^S = \text{softmax}\left(W_h^S h_i^S\right), \tag{5}$$

Reference [114] in a sense provide the dual approach to that of Reference [28], providing an attended slot prediction as the main input into intent prediction. The attention is additive on the weighted hidden states of a BiLSTM encoder and the weighted sum of the predicted slot labels. We call this explicit feed of slot information to the intent slot2intent.

Then, Reference [121] extends the intent2slot gate of Reference [28] with a pair of slot gates, one carrying the global intent information to the slot task, and one taking it to each slot location individually. Reference [123] also apply intent2slot but only to tokens determined to be not labelled "O."

Self-attention was introduced to the BiLSTM architecture to force a stronger learning "at the semantic level" between the slots and the intent [48]. A first self-attention layer performs attention on word and convolutional character embeddings. This is concatenated with the word embeddings and fed to a BiLSTM layer. The final state informs intent detection, producing the intent embedding v^{int}. Another self-attention is then performed between the BiLSTM hidden states $H = [h_1, \ldots, h_T]$,

as is shown in Equation (6). This self-attended representation s_t is combined with the intent prediction embedding v^{int} via a MLP-based gate as in Equation (7), which explicitly contributes to the slot tagging of o_t in Equation (8). Similarly, Reference [10] starts with word and character embeddings from a BiLSTM layer and then performs multi-head self-attention on these, followed by a BiLSTM encoder whose final state informs intent prediction. Another multi-head self-attention on the second BiLSTM hidden states, combined with the masked intent prediction, feed a CRF for slot prediction,

$$s_t = \text{self-attention}(h_t, H), H = [h_1, \ldots, h_T], \tag{6}$$

$$h_t^* = \text{MLP}\Big(\Big[s_t, v^{int}\Big]\Big), \tag{7}$$

$$o_t = h_t \odot h_t^*. \tag{8}$$

In Reference [12], a BiLSTM layer takes the word inputs. State attention is performed as follows. For slots each hidden state is combined with the softmax of a weighted sum of all the hidden states passed through a feed forward network. Intent detection takes the last hidden state in combination with a similar weighted sum of the intermediate states. A similar formulation is used for word attention by weighting sums of word vectors rather than hidden states. All these features are combined in a fusion layer to inform the two tasks. Reference [107] uses a standard encoder-decoder LSTM that incorporates a length variable attention, that is, attention of a sub-sequence of learned width over the hidden states.

Transformer. The transformer architecture [96], a non-recurrent model useful for capturing global dependencies via multi-head self-attention (among other strengths) appears in Reference [91] to construct contextual embeddings of the word tokens. In their model, attention is applied between all these to inform the intent prediction sub-task where it gives a superior result. In Reference [120] word embeddings are passed to a three-level transformer layer, then a global output informs intent detection, and token level output is passed to a CRF for slot detection. Differently to Reference [91], a special token is added to represent the whole utterance. Both these models only use the bidirectional encoder of Reference [96]. After the initial success of the architecture researchers looked to address some potential weaknesses, especially affecting slot filling. First, the positional encoding is not as strong as recurrence in representing sequence. This may lead to label dependency issues, so Reference [14] added back the decoder and performed a sequential slot label generation task as well as the intent and slot classification performed by the encoder. They further perform a prediction step in each transformer layer and embed the results with the layer output, all to enhance learning the probability distributions over intent and slots. Second, the Transformer is seen as agnostic to the importance of global versus local context, with an intuition that intent may be better served by global and slot filling by local. Hence, Reference [126] used a Gaussian Transformer that is designed to better highlight local context to aid the slot task. Alternatively, Reference [70] and Reference [39] performed separate encodings for slot and intent representations at the start of their circuits; each should learn different contextual balance to aid the separate tasks. The former using separate Transformer encoders while the latter used a BiLSTM for slots and a Transformer for intent.

3.2.5 Hierarchical Models. A hierarchical model passes information learned to be relevant through ordered levels. While this flow is explicit, it is often unidirectional. For example, Reference [45] supplies a hierarchical approach with slot, intent, and domain levels. Each element of the intent level is represented as the vector sum of the components in the slot layer coming from the same utterance.

In the capsule network solution of Reference [117] relevant feedback is provided from the highest level back to the lowest, an approach that sought to explicitly capture the words-slots-intent

hierarchy. A capsule represents a group of neurons whose output can be used for predictions at the next level; word capsules can be used to make slot label predictions, and so on. The hierarchy is learned using a routing-by-agreement mechanism: The prediction is only endorsed when there is strong agreement from the incoming capsule. The authors also propose a mechanism whereby a strong intent message at the highest level can be fed back to the earlier levels to help them in their task. This explicit and direct feedback is stronger than the implicit or indirect joint learning typically found in RNN models. The work was extended to a multi-task setting with extra mid-level capsules for Named Entity Recognition (NER) and Part-Of-Speech (POS) labels, with mixed results [86].

3.2.6 Bi-directional Models. A model where there is a pipeline from one sub-task to the other may be seen as unidirectional, for example, a model that predicts intent first and then incorporates that intent prediction (or some precursor) in slot labelling. A bi-directional model has an explicit path from slot processing into intent prediction and also one from intent processing into slot prediction. This can form two parallel paths through the circuit, often with a fusion layer or a joint loss. This is different to the bidirectionality seen in RNNs, where the temporal sequence of inputs is fed in a forward and backward order to separate RNN cells, whose outputs are combined. The first such bi-directional circuit was proposed in Reference [102]. There, each path is a separate BiLSTM, encoding the utterance into a sequence of hidden states $(h_1^i, h_2^i, \ldots, h_n^i)$, where $i = 1$ corresponds to the intent BiLSTM and $i = 2$ refers to the slot BiLSTM. The hidden states from each path are shared with the other to form the cell output S_t^i, as is shown in Equations (9) and (11), which is another form of explicit influence between the tasks. An optional LSTM decoder is then supplied on each side to generate the intent (Equation (10)) and slots (Equation (12)). The $\hat{y}_n^1$ in Equation (10) contains the predicted probabilities for all intent labels at the last timestep n, whereas the y_t^2 in Equation (12) represents the slot output at each timestep t. Interestingly, the loss is not a joint loss, but the circuit alternates between predicting intent for a batch, and back-propagating intent loss, and then predicting slots for the same batch and back-propagating slot loss. They call this asynchronous training.

$$S_t^1 = \phi\left(S_{t-1}^1, h_{n-1}^1, h_{n-1}^2\right), \tag{9}$$

$$y_{intent}^1 = \underset{\hat{y}_n^1}{\text{argmax}} P\left(\hat{y}_n^1 | s_{n-1}^1, h_{n-1}^1, h_{n-1}^2\right), \tag{10}$$

$$S_t^2 = \varphi\left(h_{n-1}^2, h_{t-1}^1, S_{t-1}^2, y_{t-1}^2\right), \tag{11}$$

$$y_t^2 = \underset{\hat{y}_t^2}{\text{argmax}} P\left(\hat{y}_t^2 | h_{t-1}^1, h_{t-1}^2, s_{t-1}^2, y_{t-1}^2\right). \tag{12}$$

Bi-directional paths also appear in Reference [4]. Starting with GloVe word embeddings, an intent path converts them to convolutional features that are concatenated and then projected. The slot path passes the word vectors through a BiLSTM with a CRF on top with the results also projected. Three types of fusion of the paths (after reshaping/broadcasting) were tested: addition, average, and concatenation.

Reference [22] also considers bi-directionality. They start with a BiLSTM encoder. A weighted sum of intermediate states for each step (the slot contexts) feeds a slot sub-net, while the weighted final hidden state (the intent context) feeds an intent sub-net. These two interact in either a slot2intent fashion (slot affects intent) or intent2slot. The outputs then feed a softmax intent classifier and a CRF, respectively. In slot2intent mode a learned combination of the slot contexts and intent context then feed the intent sub-net, where they are combined with the intent context for prediction. In intent2slot mode the intent context is combined with the slot contexts to form a

slot informed intent context. This is then fed to the slot sub-net where it is combined with the slot contexts to feed the CRF for prediction. As may be expected intent2slot gave better slot results and slot2intent gave better intent results. That only one can be applied is a weakness of the architecture.

Strong, direct bi-directionality with explicit intent2slot and slot2intent paths is used in the Bi-ED circuit of Reference [33]. The intent2slot path uses attention between an initial intent prediction on a BERT sentence embedding and the BERT word token embeddings. Dually, slot2intent uses attention between initial slot predictions on BERT word embeddings and the BERT utterance embedding. A similar usage of preliminary probabilities occurs in Reference [87].

Increased focus was paid to Transformers in 2021 and explicit bi-directionality has borne fruit there. References [70] and [39] extract separate slot and intent representations then add sections within Transformer layers to facilitate interaction between the two. Reference [126] uses a Gaussian Transformer for encoding to correct the balance between local and global context. These encodings then pass through intent2slot and slot2intent paths similarly to Reference [33] and Reference [87], using preliminary predictions that they call Intuitive and secondary that they call Rational. Their fusion layer learns to balance between the Intuitive and Rational predictions.

3.2.7 Memory Networks. Reference [55] considers that even with the inclusion of feedback that the circuit of Reference [117] is still overly unidirectional. To overcome this, they consider the use of memory networks. As they see the typical interaction as a pipeline from words to slots to intent, they alternate interaction from slots to intent and vice versa via multiple blocks of memory nets. The network begins with GloVe embeddings and max pooled convolutional character embeddings. These feed the first memory block, which constructs slot features, intent features, and hidden states. Further memory blocks in the stack take the previous block's hidden states as inputs. Memory blocks perform three operations, striving to capture local context and global sequential patterns:

- Deliberate Attention: A slot memory (with number of cells equal to number of slot labels) and intent memory (ditto for number of intent labels) are randomly initialised then updated. At each word position each memory is updated as a weighted sum of the other memory and of the block hidden states for the current word. Diffusion of influence between slots and intents thus takes place and can inform the hidden states for the next word.
- Local Calculation: An LSTM network receives the input embeddings or previous block's hidden states. It calculates slot and intent representations as interactions between its inputs and the slot and intent memories.
- Global Recurrence: A BiLSTM layer that encodes global sequential interactions.

After the stacked blocks a final prediction takes place. Slots are labelled via a CRF on the final hidden states. Intent is via an average of the final hidden states.

3.2.8 Meta-studies of Flow Architectures. In an approach that considers both feature creation and architecture [68] gives an interesting generalisation of multi-task learning architectures then apply it to the joint task. For example, a three sub-task parallel architecture would take samples with training labels for each sub-task; develop universal features, task specific features, and grouped features; concatenate them; and then feed them to task specific decoders. Series architectures are also given. Their base circuit uses word and character embeddings and is a standard BiLSTM encoder feeding an LSTM decoder for slots and a softmax classifier on the final hidden states for intent. No attention or slot gating occur. The base circuit is then adjusted to match some of the series and parallel architectures. Varieties of series architectures from multi-task circuit design are also tested in Reference [25].

3.2.9 Graph Networks. Graph networks can be used to address shortcomings of limited context windows suffered by RNNs and CRFs, as they can learn global relationships between words and labels. For example, Reference [119] uses a graph S-LSTM network to overcome perceived shortcomings of RNNs, being lack of parallelisation (due to sequential nature), weak local context use, and lack of long-range detection. The graph has as nodes the word representations and sentence representation from an LSTM, and hence the network simultaneously works on the whole sentence. Only word nodes within a context window are connected by edges. The sentence node is connected to all word nodes. Messages are passed between the nodes to enable global coordination. The final node states for each slot go through a convolution unit and self-attention before being used for slot filling. The final sentence node state is used directly for intent detection.

Reference [89] sees shortcomings of linear chain CRFs as being limited context and only applicable to the slot sequence. They construct a graph-based CRF **graph convolutional network (GCN)** that learns relationships among words, slot labels, and intent labels. BERT embeddings are passed through a BiLSTM that feeds the GCN for prediction. A weighted joint loss is back-propagated.

Incorporating syntactic information, Reference [90] uses a GCN to build a matrix representing the dependency parses of the training sentences, with words as nodes and labelled dependency arcs as edges in the graph. A gate filters the most useful edge labels. The matrix acts on BERT token embeddings that pass through simple softmax classifiers with good results.

3.2.10 Importing Methods from Analogous Fields. An interesting analogy, proposed in Reference [5], is that the relationship between intent and slots is similar to that between the query and image in visual question answering. Thus they borrow an idea from the latter field—Multimodal Low Rank Bilinear fusion—between the features of each part. Similarly, Reference [63] consider an analogy with the joint task of clinical domain detection and entity recognition in medical literature. Coming from the IoT field they also propose a pipeline structure where intent is detected first and then slots determined in a closed domain setting. Staying within NLP, Reference [62] recast the joint task as Question Answering and use a Transformer to ask binary intent questions ("is the intent asking about _____?") and information gathering slot questions ("what cuisine is mentioned?").

3.3 Feature Creation and Enhancement

Feature creation is a critical part of the design of circuits in NLU as it ideally should capture, at least, semantic information of the individual tokens, their context, and of the entire sentence. Then any other extension to the feature set that may be used to enhance the result may be considered, including internal (syntactic, word context) or external (metadata, sentence context) information.

3.3.1 Token Embedding. The earliest models used features familiar from methods like POS tagging and included one-hot word embedding, n-grams, affixes, and so on [40]. Another approach was to incorporate entity lists from sites such as IMDB (movie titles) or Trip Advisor (hotel names) [9].

Neural models enable the embedding of diverse natural language without such feature engineering. The first neural features were convolutional embeddings of the utterance words in Reference [108], which fed to a statistical model after Reference [40]. The first to use RNN-based token embeddings was Reference [84], but they also combined those into a CNN-based sentence embedding. Instead of just using the tokens as input, Reference [59] used at each step a convolution of the current word and the previously predicted slot labels. Reference [103] used multiple convolutional features of the embedding words but also maintained the order of the words within the convolutions. These were then fed to an RNN layer.

The gamut of word embedding methods have been used including word2vec ([67, 103]), fastText [25], GloVe ([4, 5, 17, 55, 65, 68, 91, 122]), ELMo ([43, 85, 119]), BERT ([33, 34, 38, 43, 63, 69, 123]), and References [8, 11] (pre-print only)). References [23, 26] used concatenated GloVe and word2vec embeddings to capture more word information.

While BERT displays impressive performance, Reference [98] identified a limitation (logical dependency for slot filling) and countered it by feeding it to an intent2slot gate, an attention layer, and a CRF.

Reference [30] tested 10 different word contextualisation embeddings from four different method groups (feed forward, CNN, attention, and LSTM) with different depths.

A combination of character and word embedding has been widely used [10, 26, 42, 55, 68]. However, Reference [18] use only character embedding.

Pre-computed syntactic features, for example POS tags for each token using the nltk library [23], have been included with word embeddings.

A different approach from the service robotics field identifies the importance of a key verb in an instruction in informing the slot labels [121]. The key verb is deduced from a dependency parsing. A feature is constructed from the training data to encode *a priori* dependencies between words and key verbs. The circuit takes the key verb feature and concatenates it with each word's one hot encoding. These are passed to a BiLSTM layer to produce token embeddings. Reference [90] modifies BERT embeddings using a syntactic GCN based on a graph built from the dependency parses of the training sentences.

A recent approach from intent classification proposed training triples of samples—an anchor sample, a positive sample in the same class, and a negative sample from a different class [78]. Combining convolutional and BERT encodings of each one and mapping them to Euclidean space with Siamese shared weights, an intermediate loss of the anchor-positive distance minus the anchor-negative distance is minimised. The Euclidean mapping of the anchor is used for classification. This latter approach feeds in to the emerging field of contrastive learning and methods from there should be deployed in the NLU field.

3.3.2 Sentence Embedding. The final hidden state in an RNN was frequently used as the sentence embedding [52, 103, 127]. Sentences were also embedded by using a special token for the whole sentence in References [32, 120], as a max pooling of the RNN hidden states [122], as a learned weighted sum of Bi-RNN hidden states [52], as an average pooling of RNN hidden states [59], as a convolutional combination of the input word vectors [4, 124], and as self-attention over BERT word embeddings [123]. Reference [59] also applies a sparse attention mechanism that evaluates word importance over a batch and applies weights within each sample utterance for the intent detection.

Extra sentence encapsulation tokens were used in Reference [18], which used an extra <TAGG> token after the end-of-sentence <EOS> token and saw better intent prediction performance. Similarly, Reference [65] encodes both a <BOU> and <EOU> token at the beginning and end of the utterance in their BiLSTMs.

In a paper from the intent detection only field [95], the authors tried to simplify short query input to search engines by using a dependency parser to generate syntactically well-formed queries. They simply kept the top level predicate and its dependants for the query simplification and combined it with the sentence input for further classification using AdaBoost. The essence is to provide the extracted key word pieces as auxiliary information, which proved to decrease the intent classification error rate. Because some semantic and syntactic information contained in the sentence is filtered out, when this simplified syntactic structure of a sentence was used alone as input for classification, a decrease in performance was reported.

3.4 Target Variations

The targets are typically the annotated intent and slot labels. However, Reference [118] constructs a single tag for each token that incorporates the slot tag and the sentence intent. Their circuit then just performs a single sequence labelling task and the sentence intent is deduced by a majority vote of the intent portion of the predicted tags. Related approaches include Reference [107], which uses the same single tag set, and Reference [69], which performs the intent detection at token level though separate to the slot prediction, and the final intent is taken by vote. An intent detection paper [46] used an IOB intent tagging where the B and I labels are attached to sub-sentences key to the intent, effectively turning intent detection into a slot labelling task. Reference [87] gives an alternative slot tagging scheme where each utterance has two labellings: One labels the tokens that start spans with the slot type, and all else "O"; the other labels the tokens that end spans similarly.

Embedding of targets was experimented with in Reference [45], which works with learned embeddings of slot labels, intents, and domains where the sum of slot label embeddings for an utterance is close to the intent embedding in vector space. A network can then be trained to map tokens to vectors close to the slot labels and intent for the utterance.

Similarly, Reference [41] proposes a vector embedding of the entire semantic frame (intent, slot labels, slot values) as the target. In training the utterance and the semantic frame are input and vectorised. A semantic frame vector is output. The distance between the output vector and input frame vector is minimised. In testing the text is input and a vector is output and the nearest semantic frame vector is chosen. Finally, Reference [65] proposed an extra token tag for intent keywords, for example the word "play" in an utterance with intent *PlayMusic*. In one of their models only intent keywords and non-Other slot tokens contribute to intent detection.

3.5 Issues Addressed and Solutions Proposed

Table 5 summarises by paper and year the issues addressed in the joint task and the approaches to address these issues. We expand on this table in this section.

3.5.1 Narrowness of Approach. The use of features constructed only from the tokens in the sentences may be too narrow an approach. External knowledge about the words' places in the language, the syntactic structure of the sentence, or of co-occurrence statistics amongst word and labels may aid the task.

Knowledge bases. Knowledge bases are constructs containing information or statistical priors that may be useful to the task at hand. They may be constructed independent of the task or as a preliminary step using information from the training data. They have been used for feature construction, as features themselves, and to be consulted via attention. The first to use an extra knowledge base to inform the joint task was Reference [13]. They use a K-SAN input, being a structured knowledge network. Two K-SANs are constructed, one taking a dependency parse of the utterance (syntactic) and the other an Abstract Meaning Representation graph (semantic). Each representation is tested separately. A CNN encodes the representation into a vector, while a separate CNN encodes the sentence itself into another vector. Attention is applied between the two vectors, and the results combined to give a "knowledge guided representation" of the utterance. This is included as an input to a GRU cell along with the word encodings in sequence. A second RNN just takes the utterance words as input. A weighted sum of the hidden states of the two RNNs is used for prediction.

Reference [101] incorporated the ConceptNet[1] framework as a knowledge base source. (Head, Relation, Tail) triples are extracted for each word in the utterance. The TransE model [7] for

[1]https://github.com/commonsense/conceptnet5/wiki.

Table 5. Joint Task Papers Reviewed with Addressed Issue, Approach, and Techniques

Paper (Year)	Addressed issue	Approach
[40] (2008)	Joint solution of related tasks	Tri-layer CRF, extra layer for classification
[104] (2010)	Small training sets	MEM and CRF, joint task versus pipeline
[9] (2012)	Small training sets	Tri-level HMM, bolstered features
[108] (2013)	Automated feature creation	CNN features into TriCRF
[29] (2014)	Incorporate constituency parse	RecNN on word vecs and parse tree
[84] (2015)	Context from multi-turn dialogue	RNN (token) and CNN (sentence) features, MLP
[127] (2016)	Hierarchical task relationship	RNN, LSTM
[32] (2016)	Seq2seq, joint model, architectures	BiLSTM
[13] (2016)	Incorporate language knowledge	K-SAN attention network, GRU
[122] (2016)	Apply RNN to intent	GRU
[52] (2016)	Employ encoder-decoder	Encoder-decoder with attention
[53] (2016)	Real time analysis	LSTM, MLP
[125] (2017)	NLP in navigation dialogue	BiLSTM encoder decoder, seq2seq
[59] (2017)	No long term memory, linearity	LSTM, sparse attention
[42] (2017)	Error propagation, information sharing between tasks	Word and character RNN embedding
[110] (2017)	Noisy NLU outputs	Dialogue act unit after NLU
[28] (2018)	Relationship between slot & intent attention	Slot gate, BiLSTM
[67] (2018)	Multiple utterance dialogue	Utterance to utterance attention
[106] (2018)	Use hierarchy and context	Two layer (Bi)LSTM
[103] (2018)	Capture local semantic information	CNN, BiLSTM encoder decoder
[23] (2018)	Domain dependence	Ensemble model, GRU
[83] (2018)	Slow training time	Progressive multitask model using user information
[47] (2018)	Correlation of different tasks	Multi-task model incl. POS tag
[48] (2018)	Sharing semantic information	Self-attention
[118] (2018)	Tagging strategy	Token tags include intent and slot
[124] (2018)	Spatial (context) & serial (order) information	Encoder-decoder, CNN
[102] (2018)	slot2intent and intent2slot	Bi-directional architecture
[85] (2018)	Unsupervised learning	ELMo on unused utterances, BiLSTM
[114] (2018)	Use slot labelling output for intent	Cross attention, BiLSTM, CRF
[45] (2018)	Hierarchical vector approach	Learn vectors representing elements of frame
[41] (2018)	Model relationship between text and its semantic frame	Vector representation of frame
[74] (2018)	Rare, OOV words	Paraphrasing input utterances
[117] (2019)	Hierarchical structure	Capsule network with rerouting (feedback)
[55] (2019)	Unidirectional information flow	Memory network
[81] (2019)	Poor generalisation in deployment	Sparse word embedding (prune useless words)
[75] (2019)	Many-valued slots perform poorly	Delexicalisation
[101] (2019)	Language knowledge base, multiturn	Attention over knowledge base & multiturn history
[49] (2019)	Implicit sharing between tasks	BiLSTM, multi-task (DA)
[30] (2019)	Speed	Non-recurrent and label recurrent networks
[31] (2019)	Multi-turn dialogue, using context	Token attention, previous history

(Continued)

Table 5. Continued

Paper (Year)	Addressed issue	Approach
[10] (2019)	Capturing intent-slot correlation	Multi-head self-attention, masked intent
[11] (2019)	Poor generalisation	BERT
[4] (2019)	Learning joint distribution	CNN, BiLSTM, cross-fusion, masking
[91] (2019)	Lack of annotation, flexibility	Language transfer, multi-tasking, modularisation
[121] (2019)	Key verb-slot correlation	Key verb in features, BiLSTM, attention
[120] (2019)	Learning joint distribution	Transformer architecture
[18] (2019)	Efficient model of temporal dependency	Character embedding and RNN
[17] (2019)	Lack of annotation, small datasets	Augmented dataset
[12] (2019)	Learning joint distribution	Word embedding attention
[22] (2019)	Learning joint distribution	Bi-directional architecture, feedback
[123] (2019)	Poor generalisation	BERT encoding, multi-head self-attention
[69] (2019)	Weak influence of intent on slot	Use intent prediction in slot task
[27] (2019)	Multi-intent samples	Multi-label classification methods
[26] (2019)	Multi-turn history, learning joint distribution	RNN, CRF
[68] (2019)	Optimal architecture	BiLSTM, different architectures
[8] (2019)	Non-recurrent model, transfer learning	BERT, language transfer
[79] (2019)	Low-resource languages	Transfer methods with SLU test case
[65] (2019)	Natural language	Locate intent keywords, non-other slots
[107] (2020)	Good performance in one sub-task	Joint intent/slot tagging, length variable attention
[5] (2020)	Learning joint distribution	Multimodal Low-rank Bilinear Attention Network
[25] (2020)	Learning joint distribution	Stacked BiLSTM
[119] (2020)	Limitations of sequential analysis	Graph representation of text
[100] (2020)	Non-convex optimisation	Convex combination of ensemble of models
[98] (2020)	BERT issue with logical dependency	CRF and self-attention over BERT
[63] (2020)	Model transfer, IoT	Pipeline structure from medical analogue
[43] (2020)	Unseen labels	Few-shot meta-learning
[6] (2020)	Unseen labels, language transfer	Few-shot meta-learning
[89] (2020)	Linear chain CRF limitations	GCN-based CRF
[86] (2020)	Extend capsule network	Capsule network with MTL
[15] (2021)	Multi-turn	Bi-direction, attention
[14] (2021)	Non-autoregression in Transformer	Sequential decoder task
[34] (2021)	Data scarcity	WordNet knowledge base
[33] (2021)	Explicit interaction	Context and prediction attention
[38] (2021)	Explicit interaction, multitask	Bi-direction, sentiment labelling
[39] (2021)	Diff. info for each sub-task	BiLSTM + Transformer
[51] (2021)	Noisy unannotated data	Meta-learning
[57] (2021)	Multilingual chat	mBERT, multilingual network
[62] (2021)	Scarce data	Transfer from QnA
[70] (2021)	Explicit interaction	Bi-directional Transformer
[87] (2021)	Explicit interaction	Bi-directional Bi-GRU
[90] (2021)	Syntactic clues	Syntactic GCN
[126] (2021)	Explicit interaction, global/local balance	Bi-directional Gaussian Transformer

embedding multi-relational data is used to encode the knowledge. Attention is applied between words and the knowledge base encoding. Similarly, Reference [34] uses BERT embeddings of similar concepts for each input token from the WordNet[2] knowledge base as a supplementary input.

In Reference [72] a graph representation captures the interaction between multiple intents and slots. For multi-intent a score is calculated for each intent and those above a threshold are returned. The graphs use graph attention networks. Tokens are encoded by a BiLSTM, and then multiple intents are predicted. The slot path takes the token embeddings through an LSTM that provides a feature for each token that interacts with the intent predictions and the slot-intent graph to make slot predictions.

3.5.2 Multi-turn Dialogue. Typically in NLU, only the current single utterance is analysed. Temporal information or previous utterance context or previous dialogue action are not considered. A single sentence in a conversation can be ambiguous, but the ambiguity can be eliminated if previous utterances are considered. This was shown to give a significant reduction in error rate in Reference [3], which included the context from previous queries for the intent classification and slot filling of the current query. Each sub-task is treated separately, so this is not a joint model.

There are several datasets available that contain a multi-turn dialogue around a single intent or set of related intents. In these cases the previous turns' history can be incorporated. Reference [84] fed a sentence embedding along with the predicted intent and domain labels of previous turns into the intent prediction for the current turn. Alternatively, Reference [67] calculated attention between the BiGRU embeddings of successive utterances that make up a single sample and contribute to a single intent. Similarly, Reference [101] uses attention between the BiLSTM encoding of each utterance and the previous utterances in the history. In a pattern seen elsewhere in the joint task [15] perform two separate encodings, learning two weighted sums of BiGRU embeddings of previous turns, one for intent and one for slot usage. These are concatenated with the token embeddings of the current turn.

Reference [31] looks at multiple contextual inputs in multi-turn dialogues for the current utterance. For the current utterance, they apply token2token attention and sentence2token attention at the input. Information from previous turns, including intents, slots, and dialogue actions can then be attached.

While it is sensible for the research to focus on single utterance analysis it should be noted that SLU devices are often listening to all dialogue and that filtering out-of-domain utterances and incorporating lead in dialogue can be useful to the joint task.

Multi-task learning. Looking for synergies with related tasks has been an approach in the two sub-tasks and has been actively applied in the joint task. The full semantic frame contains three levels—domain, intent, and slots. Simultaneously solving the domain with the other layers has been explored [32, 84]. An extra task was introduced in Reference [83] that predicted tags for known user information from metadata (for example location, timestamp). The metadata task is preliminary and informs the BiLSTM word embedding. The results of the preliminary task feed the regular joint task training.

Reference [47] proposes that adding a further sequential task (POS tagging) will aid the joint tasks. A single LSTM layer takes word embeddings and performs an intent and slot prediction at each step, feeding those predictions with the LSTM hidden state to a next-word POS tagger. A joint loss across all tasks is calculated. The results show that the extra task helps improve intent detection. Conversely, Reference [38] proposes a *preliminary* token level sentiment labelling task.

[2] https://wordnet.princeton.edu/.

Echoing pre-trained models the network first learns to predict these labels in an LSTM then the hidden states are used as inputs to a joint model.

Reference [110] claims that noisy SLU output can be mitigated by making it part of an end-to-end network including dialogue action prediction in the dialogue manager, with errors back-propagating from the dialogue manager refining the NLU prediction. The hidden states of a BiLSTM SLU model also feed a second BiLSTM that performs the dialogue action prediction. A joint loss across all tasks is back-propagated. In related work, Reference [49] also tied together an SLU network and a network to predict the next dialogue action. They use a stronger NLU segment to improve overall results. A joint loss across intent, slots, and actions was back-propagated, and performance exceeded the SLU model alone. Dialogue action in a multi-turn dataset was explored in Reference [31]. In Reference [25] dialogue action is the first task in a multi-task pipeline rather than the last.

Using multiple auxiliary tasks, Reference [86] incorporated POS and NER tagging simultaneously with slot tagging and intent detection using a capsule network; however, the results were generally poorer when both NER and POS were included rather than just one and mixed for different datasets indicating a generalisability issue. The method of using SLU as fine-tuning with pre-training on another task, or vice versa, has shown improvements in the SLU performance. However, the results of Reference [86], echoing those of Reference [58] on slot tagging, indicate a parsimonious approach to adding extra tasks simultaneously more often yields a better result.

3.5.3 *Generalisability.*

Domain dependence. An issue found is that a model trained successfully on one domain or dataset does not perform as well on a different domain or dataset, implying it has simply learned statistical properties of the training dataset. One issue suggested in Reference [23] is that the language in the datasets is not particularly "natural." Though their ensemble model with syntactic POS features performed well on ATIS it is unclear whether it generalised to a second dataset. A domain-invariant model proposed in Reference [23] used an ensemble of word embeddings in an ensemble circuit with a BiGRU unit and a BiLSTM unit. While together they outperform each unit used alone, the circuit did not transfer well to a new dataset. This approach of using multiple methods in one circuit for generalisability appears to rely too much on chance than good design. Post-deployment, some state-of-the-art architectures show a drop-off in performance [81]. Some issues that cause drop-off in performance are personalised language of users not matching the training data, and the cost of annotated training sets (and hence their limited size and spread). Focusing on the vocabulary, they propose a sparse vocabulary embedding that they apply to two existing architectures and show improved results. The embedding uses lasso regularisation to penalise words useless to the tasks. They apply the method to the networks of Reference [52] and Reference [28] and find that while using sparse vocabulary that intent accuracy increases but slot f1 decreases. They qualitatively discuss these results with observations on what words/structures help the two sub-tasks and the joint task. A pre-trained model should address the poor generalisability of models that perform their own embedding. Following this claim, Reference [123] uses a two-step decoder where the first step decodes intent that feeds the intent classifier and also the second decoder that works on slot labelling. The intent decoder performs multi-head self-attention on BERT encodings. In the slot decoder, a BERT embedding for a word is concatenated with the attended intent in training only if it is a "real slot," i.e., non-"O"; otherwise, it is concatenated with a random vector. Each concatenation feeds a softmax classifier for the token. A joint loss is back-propagated. The results are good for ATIS and SNIPS.

Non-English data and transfer learning. NLU is eventually required in many languages, most of which do not have the large annotated training datasets required. An aspect of generalisability of

models is thus whether they can be used outside the language on which they are trained. Papers have used the same architecture for both English and non-English datasets to give comparative studies across languages: Reference [40] used ATIS and a Korean banking dataset; Reference [122] used ATIS and Chinese questions collected from Baidu Knows; Reference [67] works only with a Chinese dataset where word boundaries are not clearly identified. With the advent of **multilingual BERT (mBERT)** (BERT trained on a multilingual corpus) in 2021 [57] compared a multilingual model versus multiple monolingual models on the joint task finding that at this stage the latter outperforms. This question and the related linguistic concept of code switching, where a speaker switches frequently between different languages offers areas for further work in the joint task. Other papers considered the transfer of the model from English to other languages to address lack of annotated data in those languages. For example, Reference [91] consider a simple weight transfer from an English model for use in German. Reference [8] (pre-print only) consider transfer learning from English to Italian. Transfer to low-resource languages is studied in Reference [79], in this case from English to Spanish and Thai. The circuit is a basic BiLSTM with CRF. They evaluate three different cross-lingual transfer methods: (1) translating the training data, (2) using cross-lingual pre-trained embeddings, and (3) using a multilingual machine translation encoder as contextual word representations. They find that using cross-lingual transfer well outperforms training on limited data from the low-resource language. The work is extended by Reference [56], Reference [6], and Reference [71] but moves into cross-lingual transfer theory and out of the scope of this survey. This issue of generalisability is still very much open and in demand by end users. Methods discussed in Section 3.5.4 for using few-shot methods to boost performance of existing models in new domains or datasets warrant further investigation.

3.5.4 Limited Training Data. Annotated training data are costly in time and resources to produce. With new domains and applications for SLU appearing, with existing domains changing, and with colloquial language shifting, there is a need for methods to perform well with limited training data.

Small datasets. An early statistical model [104] tests two-pass (pipeline, intent then slot) versus one-pass (simultaneous solving) for a small training set. They show that intent classification is much better in the two-pass model while token level slot f1 suffers slightly. Reference [88] proposed using an RNN network to learn the word/label dependency distributions from available training data. For intent, the intent label is attached to each word in an utterance. Synthetic samples are then generated for use in training. They showed that this can lead to better results for slot tagging using a CRF on three datasets but that the results for intent were inconsistent. Dataset augmentation is explored in Reference [17] where new training samples are generated from existing ones via three methods: labelled word replacement from an external synonym lexicon, random replacement of outside words with a synonym, and "sequence order mutation"—change of order of spans for utterances with one labelled span. They showed that augmentation can improve the slot f1 result, and more so for smaller datasets, but has little effect on intent accuracy. There is a further literature on dataset augmentation for SLU that we will not cover here.

Lack of annotated data. As new domains appear, it takes time and cost to develop annotated datasets for training. One approach is to train on user metadata as a preliminary step [83]. They show they can achieve higher slot f1 scores on smaller training sets and with fewer epochs than only using the intent and slot annotations. Alternatively, Reference [85] constructs an unlabelled utterance dataset collected from ASR interactions with their agent. They train an ELMo-style word embedding on this dataset. For the joint task, they find their embedding outperforms fastText. As well as language transfer, Reference [6] also address the transfer to new label domains with

minimal samples available via a few-shot meta-learning approach. Reference [51] also use few-shot meta-learning on datasets with noise added, proposing a new robustness benchmark test.

Unseen labels. Reference [43] addresses the issue of unseen test classes by applying two few-shot algorithms, model agnostic meta-learning and prototypical networks, in combination with three word embeddings—GloVe, BERT, and ELMo. They find the prototypical network algorithm performs best, that joint training significantly improves slot filling span-based f1, and that ELMo and BERT share the spoils from the word embeddings.

3.5.5 The OOV Issue. Out-of-vocabulary (OOV) words in the test set may lead to lower test performance. Similarly, the use of rare words in the training set may introduce unwanted bias. This issue is related to generalisability and also to changing vocabulary from user to user or over time. Delexicalisation was investigated in Reference [122], which sets all words that only appear once in the training set to an unknown UNK token. Then new words in the test set are also set to the UNK token. They also replace all numbers with a generic DIGIT token. This is also applied in Reference [48] and Reference [119]. Paraphrasing of input utterances to cater for rare or OOV words, or for unusually phrased requests, was used in Reference [74]. The paraphrasing is performed by an encoder-decoder RNN and is performing a kind of translation. The paraphrase can be applied to any downstream model. Another approach proposes BERT embeddings as a sop to rare or OOV words [11]. BERT uses word-piece encoding to provide a meaningful embedding for all words. Reference [75] addresses the issue of networks having trouble with slots with large semantic variability—that is, there are many values the slot can take during training and many unseen values during testing/deployment. They call these **out-of-distribution (OOD)** slots. They propose a new delexicalisation method. This replaces values in OOD slot locations with default values in pre-processing.

3.5.6 Obfuscation and Speed. Taking a contrary view, Reference [30] considers how joint modelling may obfuscate, or hide, information and may also be unnecessarily slow. They propose a modularised network with separated tasks after a common word contextualisation pre-processing. The modularisation enables easier analysis of results. They perform speed analysis within their model suite. Then Reference [100] proposed a convex combined multiple model approach to counter limitations of non-convex optimisation, one of which is slow speed of convergence due to being stuck near non-optimal solutions. Each network in the circuit has the same structure but different initialised weights. A convex combination of label predictions from each network is used as the label prediction for each slot and the intent. Both a local loss function for each network and a global loss function on the combination are back-propagated. The networks are BiLSTMs with a context layer. The convex combination outperforms single classifiers. The speed improvements are significant.

3.5.7 Real-time Learning. In Reference [53], the authors consider real-time analysis where the whole utterance is not analysed but a prediction is made at each timestep. In this RNN, the intent is predicted at each step and used as context to the slot prediction (as well as a next word language model). Thus, the current slot prediction is conditional on the input words to that point, the previous slot predictions, and the previous intent predictions. The recurrent unit is an LSTM, but the current intent and slot predictions use MLPs on the current hidden state.

3.5.8 Label Dependency. There are dependencies between slot labels, meaning that some slots appear more commonly with some other slots in the same utterance. For example, in a travel dataset it is highly probable that *B-FromCity* and *B-ToCity* are in the same sentence. Capturing such label dependencies would help find the best slot combinations and generate better prediction

results. This is an issue covered in the slot filling literature and the methods used there including CRFs [111, 112] and encoder-decoder seq2seq models [44, 128] have been used in the joint task. We note particularly that the use of CRFs after a deep learning solution became popular again from 2018 (see Table 4) to address this issue [22, 26, 114, 120]. In Reference [98], a CRF is used to counter a label dependency limitation for slot filling in using BERT due to its non-recurrent nature. However, Reference [10] claims earlier models neither perform slot filling realistically enough (so reflecting the language priors) nor explore intent-slot correlation well. They propose to use a CRF for the former and a masked intent prediction as an input to the CRF for the latter. The mask is "a conditional probability distribution of slot given intent, obtained from training data." Similarly, Reference [4] uses a CRF with masking, prior conditional probabilities of slot/intent co-occurrence obtained from training data, for slot prediction. Label dependency issues may occur in Transformers due to their non-recurrent nature. Reference [14] address this with a sequential slot label generation task in the decoder. Reference [39] also identify the different type of information to be useful for each subtask and perform separate embeddings—a BiLSTM with its stronger positional information for slot filling and a transformer for intent semantics. The embeddings later interact via attention.

3.5.9 Handling Multi-labels. Most annotated queries in training datasets express only one intent. In real life, however, queries may contain more than one intent. For example, the query "find Beyonce's movie and music" has two intents, "find_movie" and "find_music." An NLU system should be able to handle multi-intent queries.

Approaches in the joint task include simply removing multi-label samples from the dataset [106] or using both the first label as the only label and merging labels to a compound label [17]. Reference [72] considered multi-intent datasets, including their own extension of SNIPS to multi-intent. For multi-intent a score is calculated for each intent and those above a threshold are returned.

A study of both sentence level and token level multi-intent detection took place in Reference [27]. For ATIS, they split the compound multi-labels, giving about 2% of the dataset with multi-labels. They also use an internal dataset with 52% of the samples having multi-labels. Although the assignment method is unclear, a sentence may be assigned multi-labels during prediction, and these are then assigned to individual tokens in the sentence to aid with slot filling.

4 DATASETS

A summary of the most commonly used datasets is presented in Table 6. Here we cover the most commonly used datasets, ATIS and SNIPS, popular due to their easy availability and ubiquity of use allowing comparison between models. We then briefly cover the other datasets.

4.1 The Air Travel Information System

ATIS was introduced in 1990 in Reference [35], and its history is instructive in understanding some of the conventions of the field. The domain is air travel information including "information about flights, fares, airlines, cities, airports, and ground services." The first release, ATIS-0, collected 740 evaluable samples. Each sample contained a sound file of a single utterance question, a transcription of the question, a set of tuples constituting the answer, and the SQL query that produced the tuples. Tokens were generated according to **Standard Normal Orthographic Representation (SNOR)** rules: white-space-separated lexical tokens using case-insensitive alphabetic text; spelled letters represented with the letter followed by a fullstop (e.g., "a. b. c."); no non-alphabetic characters (except apostrophes for contractions and possessives and hyphens for hyphenated words and fragments). The average length of the SNOR translated utterances was 11.3 tokens.

Table 6. Major Datasets Used in the Literature, Single Turn in English Unless Noted, Train-Val-Test Gives the Number of Utterances

Name	Public	Train-Val-Test	Num Intent	Num Slots	Domain, Notes
ATIS	Y	4478/500/893	21	128	air travel
SNIPS-NLU	Y	13084/700/700	7	72	personal assist.
FRAMES	Y	20006/—/6598	24	136	hotel, multiturn
CQUD	N	3286	43	20	Chinese, question answering
TREC	Y	5500/—/500	6(50)	—	question classification
TRAINS	N	5355/—/1336	12	32	problem solving, multiturn
Microsoft Cortana	N	10k/1k/15k	10–20	15–63	personal assist., multidomain
Facebook	Y	30521/4181/8621	12	11	multi-lingual task oriented
SRTS FrameNet	N	2803/—/312	12	61	robotics
Alexa	N	264000/—/—	246	3409	17 domains
DSTC2	Y	4790/1579/4485	13	9	multiturn, restaurant search
DSTC4	Y	5648/1939/3178	87	68	multiturn, tourism dialogue
DSTC5	Y	27528/3441/3447	84	533	dialogue with social robots
CMRS	N	2901/969/967	5	11	Chinese, room reservations
CU-Move	N	57584/—/—	5	38	in-vehicle dialogue
AMIE	N	3418/—/—	10	7	in-vehicle dialogue
TeleBank	N	2238/—/—	25	17	Korean, banking
CONDA	Y	26921/8974/8974	4	6	in-game chat
MTOP	Y	73174/10453/20907	117	78	11 domains, 6 languages
MIT Movie_Eng	Y	8798/97/2443	—	25	movies, slot only
MIT Restaurant	Y	6894/766/1521	—	17	restaurants, slot only

Extensions to the dataset were made available in subsequent years ATIS-1 [66], ATIS-2 [36], and ATIS-3 [19] in late 1993 to mid-1994. The set that evolved to become the standard ATIS for NLU analysis was drawn from the annotated samples in ATIS-2 and -3. Reference [115] was the first to use the combined set for language understanding. Then Reference [76] used the same set but tweaked the annotation to something more closely resembling the ATIS set used today. The set contains 4,978 training samples and 893 test samples. In more recent years with the advent of neural net models, 500 of the training samples are set aside as a validation set. Work toward formalising the ATIS dataset took place in Reference [94], using the same samples as in Reference [115] and Reference [76]. The intents listed in Reference [94] are not the current ones, as they list 17 intents each of which have non-zero frequency in train and test set. In later releases, some joint intents are included to give 21 intents. Also, in later releases, the SNOR rules are relaxed. For example, punctuation is allowed ("st. louis"), utterances are all lowercase, and numbers are allowed for times and years but not dates. We note that in the version of the dataset used today that the intents are highly imbalanced with 75% of the samples in a single intent.

Reference [94] performs an AdaBoost classification on word n-gram features for intent classification and then separately a CRF method to label slots and then performs a classification of error types into six types for intent and five types for slots. They suggest research directions based on these errors as follows: use of parsers to identify head words or clauses, *a priori* information (knowledge bases), and methods to enable long-distance pattern identification, as opposed to more local, shorter patterns. They also measure the high misannotation rate (2.5% for intent and 8.4% for slots). In 2018, Reference [1] performed the next analysis specifically to question the usefulness of

ATIS. They ran a set of different methods from a boosted tree ensemble to a BiLSTM net on ATIS slot tagging with and without named entity tag labelling. They use the same data as in Reference [76], which removes issues with position labels (B,I,O) by collapsing semantic spans as single tokens. For example, "san jose" is a single token not two. While this weakens their approach, the results are worth looking at.

They chose their five best models and cluster the predicted slots according to (1) **agree/correct (AC)** (all models get the slot correct and agree on the answer), (2) **non-agreement/error (NE)** (all models got the slot wrong but there is no agreement on the errors), (3) **agree/error (AE)** (all models got the wrong slot, but they all made the same error), and (4) **non-agreement/correct (NC)** (models do not agree on the solution, but at least one is correct).

These clusters suggest future directions for research. While AC is "solved," AE and NE are open problems (aspects of the dataset not captured by the models), and NC are useful for model comparison between those that got them right and those that did not. They also highlight issues with the dataset—bad annotations, ambiguity "where slots could be labelled with different labels," and repetition errors where "only the first mention of an entity is labelled," e.g., in "show flight and prices Kansas city to Chicago on next Wednesday arriving in Chicago by 7pm" Chicago is only labelled once. They estimate that about 2.5% of the utterances are erroneously slot-tagged and conclude that ATIS is at the end of its useful life for analysis. The next deep analysis of the ATIS dataset took place in Reference [64] where the authors extensively reviewed the shortcomings of the dataset. They have subsequently re-annotated the dataset fixing what they deem errors. Even without this re-annotated version of ATIS, results reported in the literature show that the test intent accuracy being achieved is now above 99% and slot f1 above 98%. It appears that the models to date have successfully captured the joint distributions of words, slots, and intents in the dataset. Further models may only make improvements at the edges and, while useful, may be hidden by what appear non-significant increase in the evaluation measures.

4.2 SNIPS

The SNIPS Natural Language Understanding dataset and its creation are fully described in Reference [16]. It contains 15,884 utterances in seven balanced intent classes. In training, there are 72 slot labels and a vocabulary size of 11,241 words. The average sequence length is 9.05. Unlike ATIS, SNIPS covers different domains—weather, restaurants, and entertainment. Reference [55] showed an interesting visualisation that the slot labels used for different domains form largely disjoint sets. These differences have made it a useful counterpoint for experimentation in NLU, and models addressing both ATIS and SNIPS successfully show they can handle imbalanced data. However, the reported test results for SNIPS too are very high—intent accuracy above 99% and slot f1 around 98%.

4.3 Other Datasets

Microsoft have several non-publicly available sets that have been used by Microsoft researchers. For example, FRAMES is multi-turn dialogues around hotel bookings. The Microsoft Cortana personal voice assistant datasets have at least six domains—weather, calendar, communication, reminder, alarm, and places. Other software houses with datasets include Facebook (public) and Alexa (private). Some competitions have applicable datasets, for example the DSTC 2, 3, and 5 competitions have been used in papers. These often contain multi-turn dialogues. Also the Chinese competition-based CCKS dataset has been used for research. The TRAINS dataset, a collection of problem-solving dialogues, has been used in four papers. Datasets from diverse but relevant fields have been FrameNet from robotics, CU-Move and AMIE from in-vehicle communication, CONDA [105] from in-game dialogue, and from question answering CQUD (from Baidu Knows), Yahoo,

and TREC (only intent annotated). Non-English datasets have been generated: Reference [2] derived an Italian dataset starting by translating SNIPS and then using Italian words for tokens like cities or movie names; ATIS has been translated into multiple languages as MultiATIS++ [109]. For multilingual transfer learning [79] provide their own tri-lingual dataset and MTOP provides nearly parallel utterances across 11 domains and 6 languages [50].

4.4 Discussion

ATIS and SNIPS have reached near to the end of their useful lives as benchmarks for the joint task. Very high test results show that the methods developed in this survey can successfully learn the joint distributions of intent and slot labels, and slot labels with each other, in a supervised learning setting. They appear to be set to continue being the benchmarks due to the ability to compare a new approach to previous ones, though this should be tempered by the use of non-standard experimental set up discussed in Section 6. They are useful for study, because they are single utterance, have reasonable numbers of intents and slots, are task focused (so have a clear intent), and have reasonable utterance lengths. Challenges include mis-annotation, OOV issues, and perhaps the level of unnaturalness of the language. In their defence, they provide differences—class imbalance versus imbalance and single versus multi-domain—and a model that scores well on both can claim to have some generalised ability. However, as noted in the literature, the greater generalisability of such supervised learning models to new domains is in question. It is probable that more naturally conversational data should be tested. To avoid costly annotation this should be largely unannotated, encouraging research in zero- or few-shot methods. Such methods can still be tested on ATIS and SNIPS (as in Reference [43]). Metrics for measuring the efficacy of such models in the absence of annotation need to be considered. We further note that all the few- and zero-shot papers reviewed use annotated datasets for evaluation, and hence still need to be transferred to new unseen datasets.

5 EVALUATION METRICS

5.1 Intent Classification

5.1.1 Intent Accuracy. For intent classification the widely used metric of accuracy is used for evaluation. Accuracy is the ratio of the number of correct predictions of intent to the total number of sentences. Some papers instead use error rate, the ratio of wrongly classified samples to the total number of samples, equivalent to accuracy subtracted from 100%, to measure intent classification performance [61]. Some utterances in the ATIS have more than one intent label. Most researchers, since they are not doing multiple label detection, consider the combined label as a new label type, e.g., atis_airfare#atis_flight_time. Reference [119] notes that "some researchers [48, 52] count an utterance as a correct classification if any ground-truth label is predicted. Others [22, 28] require that all of these intent labels have to be correctly predicted if an utterance is to be counted as a correct classification."

5.1.2 Intent Precision, Recall, and f1. Less frequently micro-averaged precision, recall, and f1 are used to evaluate intent prediction [49, 110]. For an intent class C, **True positives (TP)** are intents that are correctly classified as of class C, **False positives (FP)** are intents that belong to other classes but are incorrectly classified as class C, and **False negatives (FN)** are intents of class C that are incorrectly classified as other classes. These are then used in the standard equations for precision, recall, and f1. A variation used for multi-label identification is precision and recall at the top-k predictions. Here precision is the ratio of correct labels in the top-k predictions divided by k and recall is the ratio of correct labels in top-k predictions over the total number of correct labels [116].

5.1.3 Tests of Significance. Several standard tests are used to investigate whether the difference in intent metric between two models is significant. Welch's t-test has been used with the p-value threshold set to 0.05 [24, 26]. Others use the Student's t-test for similar purposes [85]. McNemar's test is also applied to test paired binary classified data to evaluate how well two tests agree with each other. Reference [40] used this for a classification of ATIS intents into two domains.

5.2 Slot Labelling Evaluation

5.2.1 Span-based Slot Precision, Recall, and f1. A span (sometimes called a chunk) refers to a sequence of words labelled from the same meta-class. For example the labelling B-MISC I-MISC I-MISC is a span of meta-class MISC. For a meta-class C, we can thus define at the span level: TP is the number of spans of meta-class C that are wholly correctly predicted; FP is the number of spans of a different meta-class that are incorrectly predicted as of meta-class C; FN is the number of spans of meta-class C that are incorrectly predicted, partially or wholly, to another meta-class. Micro-averaged and macro-averaged precision and recall and f1 can then be calculated. In most papers, slot f1 is reported as the span-based micro-averaged f1 over all meta-classes excluding "O." The conlleval.py[3] script is regularly used to calculate this micro-averaged precision, recall, and f1 [18, 20, 55].

5.2.2 Token-based Slot Precision, Recall, and f1. Token-based slot measures are used in Reference [49]. In this evaluation metric, TP, FP, and FN are calculated at the token level, with the same definitions as in Section 5.1.2. B-MISC and I-MISC are treated as separate unrelated classes, so some label dependency information captured by the span-based metric is lost.

5.2.3 Slot Accuracy. Slot accuracy is the ratio of the number of correctly labelled slots to the total number of slots. This is used in Reference [113] where it is referred to as word labelling accuracy.

5.3 Semantic Accuracy

A sentence is correctly analysed if both the intent is correctly predicted and all the slots (including O labels) are correctly predicted. Semantic accuracy is then the number of correctly analysed sentences divided by the number of sentences.

6 EXPERIMENTAL SETUP

The standard experiment trains on annotated utterances, creates features, and learns to predict an intent and slot labels for each utterance. A held-out, unseen test set is used for evaluating performance. This experimental setup varies for different papers. Further, many do not clearly state their setup with respect to datasets and hyper-parameters. In those papers that do specify the setup for datasets, most utilised a train-test split, usually as 80–20%. The training data may be split further to make a validation set. Alternatively, papers use 5-fold or 10-fold cross-validation for evaluation. Results are sometimes reported to be averaged over a number of runs. For parameter tuning, dropout rates ranged from 0.003 to 0.5 and the size of hidden states was normally between 100 and 200, with as low as 64. Some models use the Adam optimiser with a learning rate between 0.0001 and 0.01 or started with 0.02 and halved it after 10 epochs [97] or halved the learning rate when the loss reduced below a threshold [73]. Some papers mention that they set parameters randomly in the beginning and then apply 5-fold validation when tuning parameters. Word embedding dimensions vary from 64 to 1,024. The problem of lack of reporting also occurred with the number of epochs. When stated, most models were trained for less than 50 epochs, with some of these

[3] https://github.com/sighsmile/conlleval.

using early stopping. Reference [28] applied 10 epochs for ATIS and 20 epochs for SNIPS in all of their experiments. While Reference [30] allowed unlimited number of epochs with a stopping criteria, Reference [69] used 300 epochs with no early stopping. Further, some models report the final epoch results while other report the best results. Considering that many papers did not clearly state their experimental setup, it may bring difficulties in model result replication. Therefore, it is recommended that papers include detailed information about setup in the experiments section.

7 PERFORMANCE SUMMARY

To summarise performance in the joint task, we list the models and their reported test results for the ATIS and SNIPS for the standard evaluation metrics (if available) in Table 7. Papers are included in this table if at least one of their results is better than the benchmarks reported in papers from the previous calendar year. Several interesting patterns can be observed based on the results available: (1) The overall improvement on slot f1 (from 87.3 in 2016 to 98.78 in 2019) and semantic accuracy (from 73.2 in 2016 to 93.6 in 2020) for SNIPS over time is much more significant than ATIS (from 93.96 in 2014 to 98.75 in 2019 for f1 and from 78.9 in 2016 to 91.6 in 2020 for semantic accuracy), while intent accuracy moves in the opposite way (from 98.32 in 2016 to 99.76 in 2019 for ATIS and from 96.7 in 2016 to 99.98 in 2019 for SNIPS). (2) For those models that reported slot f1 and intent accuracy on both datasets, 16 of 29 perform better in slot f1 for ATIS and in intent accuracy for SNIPS. (3) Before 2019, the best performance for semantic accuracy come from ATIS while a shift to SNIPS starts from 2019 ending with the best semantic accuracy in that dataset.

As mentioned in the previous section, there is a wide variety in the number of epochs from which NLU models derive their reported performance for comparison with others. Reference [28] made an early attempt at performance comparison under the same epoch settings (10 epochs for ATIS and 20 epochs for SNIPS). Reference [30] clearly specified the epochs to converge for each of the models being compared. We view these as two alternative ways to compare performance with regards to the epochs and suggest that they can provide an auxiliary aspect in model comparison and selection for future research. Thus, similarly to Reference [28], we extracted those models that use 10 epochs for ATIS and 20 epochs for SNIPS and provide the test results in Table 8. These results are from papers who follow this etiquette or from the reproduction in Reference [28] or are replicated by us using the GitHub code supplied by the authors when indicated by ††. In the latter case, we also confirm the consistent calculation of intent and semantic accuracy as well as span-based slot f1. A similar pattern can be observed: that a significant improvement in slot f1 and semantic accuracy for SNIPS has been made since 2019 and that most of the models perform better in slot f1 for ATIS and intent accuracy for SNIPS. These patterns may be related to the various distributions of slot and intent labels and the different nature of domains of the two datasets with regard to different architectures of the models.

In summary, we note that the results are now high enough for the two most commonly used datasets that any fruitful new developments may be lost in results that appear to not significantly increase the metrics for these datasets. Just as SNIPS grew to become standard, and offered different aspects to ATIS (balanced data, multi-domain), it is probable that a new dataset should become part of the SLU reporting canon. It should address the issues of unlabelled data and emerging domains, as these problems should be the focus of newer models.

8 CONCLUSIONS

Section 1 presented three questions concerning joint intent detection and slot filling. Following the survey we now address these questions.

Table 7. NLU Performance on ATIS and SNIPS-NLU Datasets (%)

Paper, Model	ATIS			SNIPS		
	Slot f1	Int acc	Sem acc	Slot f1	Int acc	Sem acc
[40] (2008) Joint 2	94.42	93.07	—	—	—	—
[108] (2013) CNN TriCRF	95.42	94.09	—	—	—	—
[29] (2014) RecNN+Viterbi	93.96	95.4	—	—	—	—
[84] (2015) RNN Joint + NE	96.83	95.4	—	—	—	—
[32] (2016) in Reference [28] (2018)*	94.3	92.6	80.7	87.3	96.9	73.2
[13] (2016) K-SAN Syntax	95.38	—	84.32	—	—	—
[122] (2016) W+N	96.89	98.32	—	—	—	—
[52] (2016) in Reference [28] (2018)*	94.2	91.1	78.9	87.8	96.7	74.1
[28] (2018) Slot-Gate (Full Atten)* †	94.8	93.6	82.2	88.8	97.0	75.5
[28] (2018) Slot-Gate (Intent Atten)* †	95.2	94.1	82.6	88.3	96.8	74.6
[103] (2018) Attention and aligned	97.76	97.17	—	—	—	—
[23] (2018)	98.02	98.43	—	—	—	—
[48] (2018) †	96.52	98.77	—	—	—	—
[47] (2018)	94.81	98.54	—	—	—	—
[102] (2018) †	96.89	98.99	—	—	—	—
[114] (2018) ACJIS Model	96.43	98.57	—	—	—	—
[85] (2019) ELMo	95.62	97.42	87.35	93.9	99.29	85.43
[22] (2019) SF First (with CRF) i †	95.8	97.8	86.8	91.4	97.4	80.6
[117] (2019) Capsule i †	95.2	95.0	83.4	91.8	97.3	80.9
[30] (2019) CNN 3L, 5 kern, label recur	96.95	98.36	—	94.22	99.1	—
[30] (2019) CNN 3L, 5 kern, label rec.*	95.27	97.37	—	92.3	97.57	—
[10] (2019)	96.54	98.91	—	93.94	99.71	—
[120] (2019)	95.1	97.2	—	93.3	98.9	—
[18] (2019) BiLSTM-CRF	95.6	96.6	86.2	94.6	97.4	87.2
[55] (2019) CM-Net with GloVe	96.2	99.1	—	97.15	99.29	—
[55] (2019) CM-Net with BERT	—	—	—	97.31	99.32	—
[69] (2019) †	95.9	96.9	86.5	94.2	98	86.9
[69] (2019) Model+BERT	96.1	97.5	88.6	97	99	92.9
[26] (2019) HCNN+CRF,word+char emb	97.32	99.09	—	94.38	98.24	—
[8] (2019)	95.7	97.8	88.2	96.2	99	91.6
[123] (2019)	98.75	99.76	—	98.78	99.98	—
[68] (2019) Base	95.4	96.1	—	94.8	98	—
[68] (2019) Base+BERT	95.8	96.6	—	94.5	97.6	—
[11] (2019) BERT †	96.1	97.5	88.2	97	98.6	92.8
[11] (2019) BERT+CRF †	96	97.9	88.6	96.7	98.4	92.6
[98] (2020) SASGBC	96.69	98.21	91.6	96.43	98.86	92.57
[89] (2020) fully-E@EMG-CRF	96.4	99.0	89.6	97.2	99.7	93.6
[33] (2021)	96.3	98.6	88.6	97.2	99.2	92.8
[38] (2021)	96.4	98.2	88.5	97.6	99.3	93.0
[70] (2021) BERT	97.1	98.8	93.1	96.1	98.0	88.8
[90] (2021)	97.3	98.3	90.2	98.3	98.9	90.2

* denotes ATIS 10 epoch, SNIPS 20 epoch, i denotes epoch count implied, † indicates GitHub available, — denotes not reported.

Table 8. NLU Performance on ATIS and SNIPS-NLU Datasets (%) Using ATIS 10 Epoch and SNIPS 20 Epoch (i Denotes Epoch Count Implied)

Paper, Model	ATIS			SNIPS		
	Slot f1	Int acc	Sem acc	Slot f1	Int acc	Sem acc
[32] (2016) in Reference [28] (2018)	94.3	92.6	80.7	87.3	96.9	73.2
[52] (2016) in Reference [28] (2018)	94.2	91.1	78.9	87.8	96.7	74.1
[28] (2018) Slot-Gate (Full Att)†	94.8	93.6	82.2	88.8	97.0	75.5
[28] (2018) Slot-Gate (Int. Att)†	95.2	94.1	82.6	88.3	96.8	74.6
[48] (2018) ††	94.82	97.00	84.00	88.93	97.71	76.43
[102] (2018) ††	95.14	96.08	84.87	88.46	96.71	75.39
[22] (2019) SF First (with CRF) i†	95.8	97.8	86.8	91.4	97.4	80.6
[117] (2019) Capsule i†	95.2	95.0	83.4	91.8	97.3	80.9
[30] (2019) CNN3L,5 kern,lab rec	95.27	97.37	—	92.3	97.57	—
[69] (2019) ††	93.15	95.9	80.4	90.88	97.14	79.71
[11] (2019) BERT ††	95.54	97.54	87.35	96.91	98.43	92.43
[11] (2019) BERT+CRF ††	96.03	97.76	88.47	96.60	98.57	92.14
[33] (2021) Bi-ED	**96.3**	**98.6**	**88.6**	**97.2**	**99.2**	**92.8**

†† reproduced by this article, — denotes not reported.

Q1: How do joint models balance the two aspects, intent classification and slot filling?

In NLU intent detection, features are constructed to represent the input sentences. The features look to capture semantic information in the words. Many approaches have been made to extend the feature set using internal (syntactic, word context) or external (metadata, sentence context) information. These features are passed to classifiers from classical and deep learning methods that look to discover a well-defined decision boundary between the features. A triple loss may be used to better define the decision boundary. Attention mechanisms have also been integrated in models for identifying which parts of sentences should contribute to the classification. Slot filling is a sequence labelling problem, and in early years drew from methods for statistically modelling the dependencies within sequences, like CRFs and HMMs. The strength of RNNs in this area was noted and applied, leading to improved performance. Interestingly, the use of CRFs returned as a post-RNN step, due to their efficacy at handling label dependency issues that were sub-optimally addressed by RNNs (impossible slot sequences for example). For feature creation, the semantic information within the words is used. Context windows from small to long range within the sentence are also used, with attention being one approach for eliciting useful context. Slot tagging has experimented with external knowledge bases for extra performance. Long-range dependency has been addressed with deep CRFs and Transformer architectures.

Multi-task learning has been used by both tasks to look for synergistic learning from other related tasks. The joint task itself is an example of this approach. A joint model needs to learn the joint distributions of intent and slot labels, while also paying regard to the distributions of slot labels within utterances. Approaches to the joint task range from implicit learning of these distributions, through explicit learning of the conditional distribution of slot labels over the intent label, or vice versa, to fully explicit learning of the full joint distributions.

The RNN architecture has been exploited, as it not only provides a state at each temporal token step but also a final state that encapsulates the sentence. One critical observation made of many purely recurrent models was that the sharing of the information between the two sub-tasks is implicit. Attention was used to make this more explicit. Self-attention between the word tokens has been used to learn label dependency and as a stronger alternative to learned weighted sum attention. Transformer encoders are a prevalent non-recurrent architecture that perform self-attention

amongst tokens, address long-range dependency, and can form a sentence representation from the transformed token representations or by using a special sentence token. This clearly points to the BERT pre-trained model architecture that has been used as input to classifiers as well as in more integrated architectures.

Another method to make influence between the tasks explicit is hierarchical models. Capsule models pass slot deductions of sufficient confidence to an intent detection capsule and vice versa. Memory networks also use a form of explicit feedback. An explicit influence between the two sub-tasks comes even further to the forefront in bi-directional models (see Section 3.2.6). Reference [22] proves slot2intent and intent2slot influence improve results but did not fuse the two approaches. Fusion approaches here include alternating between slot2intent and intent2slot [102], post-processing fusion [4], and Reference [33], in which bi-directional, direct, and explicit influence is central to the model architecture. Even within the joint task some researchers highlight non-optimal handling of label dependency in slot labelling. Adding CRFs to a deep joint model is the common solution to this problem. Graph networks have been designed to garner knowledge from the training data of word-slot, slot-slot, word-intent, and slot-intent correlations. The use of sentence context in multi-turn dialogues would seem to provide even more fuel for explicit influence by incorporating intent to intent dependency albeit unidirectionally in time through a conversation. Multi-dialogue also offers the interesting, aligned problems of identifying out-of-domain utterances and change of intent within a conversation.

Q2: Has syntactics been fully exploited or does semantics override this consideration?

The early experiments using classical syntactic features were ambivalent about their efficacy for the NLU tasks. In intent detection [95] used a simplified syntactic representation of a sentence as input but found this reduced performance and had to be used alongside the full input sentence for any gain. In slot filling the use of syntactic features gave a small uplift in NLU for text data [20]. The features they constructed included head words from dependency parsing and POS tags over windows and were designed to capture long-range influence beyond the n-grams they used. As can now be expected, deep learning methods would be much more effective at solving this issue. The earliest attempt at a neural model to address the joint task did in fact use a syntactic structure on top of word vectors [29]. Their recursive neural networks work over the constituency parse tree of the utterance, with leaves corresponding to the words (represented by word vectors). As described in Section 3.2.2, this recursively constructed a state at the root that was used for intent classification. However, the result did not improve on what had been achieved for the separate tasks. POS tags features have been used as pre-computed syntactic features in deep learning to address issues like short query length. The use of features constructed only from the tokens in the sentences may be too narrow an approach. External knowledge about the words' places in the language, or even just in the dataset, may be useful. For example, the use of *a priori* dependencies between key verbs obtained from dependency parsing and other words used by Reference [121] was a novel approach. This external syntactic knowledge may be encoded in knowledge bases as explored by Reference [13], which uses attention between a syntactic and a semantic K-SAN, and Reference [101], which embed (Head, Relation, Tail) triples for each word in the utterance as a knowledge base, then use attention between word embeddings and the base. The inclusion of knowledge embedded in graph representations or networks that perform tasks on such graphs has borne fruit in the very recent literature [90]. Further research in this area could include other types of graphical representations and incorporate information not just from the current training set or external knowledge bases but some combination of the two or data from several training sets. In some ways, this points to the implicit capture of features of the syntax of a language by the representations produced by large, bi-directionally contextual, pre-trained models like BERT and the Generative Pre-Trained Transformer (GPT) series. While the results produced by using these

models have been impressive there is incremental gains to be made by explicit representation using knowledge bases. Finally, as suggested in Reference [23], the language in the standard NLU datasets is not particularly "natural," usually following a set of standard query structures relevant to the service provided by the device and almost developing a query shorthand. This should lend itself to a knowledge base that represents these patterns well, as in Reference [61], for example, but the field should be extending itself to more general and natural human language patterns.

Q3: Can models successfully trained on a supervised dataset from one domain be made more generalisable to different domains or languages or to unseen data?

The issue of generalisability is still very much open and in demand by end users. Deployed models may show a drop-off from experimental performance. New intents and slots may appear and changes in language used may degrade performance. Then a model for a new domain with no or little training data may be required, and NLU is eventually required in many languages, most of which do not have the annotated training datasets required. One issue is that the standard datasets are fully annotated and have their own particular patterns and language that many models have now successfully learned. Ensemble models have been used to address this though with mixed success, as have delexicalisation and other methods to generalise the language in training. Ultimately though, the use of pre-trained models has better addressed the poor generalisability of models that perform their own embedding. Cross-lingual generalisation is covered in the Section 3.5.3 and includes methods like same training architecture for datasets in different languages, same architecture and weight transfer, and translation of datasets (including substitution of named entities with target language examples). The work in Reference [79] showing the better performance of cross-lingual pre-trained embeddings is another feather in the cap of pre-training for generalisability. Some research in few- and zero-shot training methods has begun in recent years to address the development of models in new domains and this promising area should be given further attention. For unseen data and emerging intents, clustering has been used to identify new intents. Creation of new intent representations by adding anomalous differences to existing intent representations has been used successfully. Unseen slot labels were addressed with pointer networks or slot description representations. Meta-learning algorithms are a more recent development addressing unseen labels and are an active area of research.

9 FUTURE DIRECTIONS

There are two major future directions for joint Natural Language Understanding.

NLU Dataset. As mentioned in Section 4.1, most of the existing NLU research uses ATIS and SNIPS, which are single turn, small in size, and have several quality issues in their annotation. It is now easy to achieve very high performance in both intent classification and slot filling. To test the performance of the joint learning aspect for NLU models, it is worth the further development of datasets and benchmark tasks incorporating multi-turn tasks, multi-language and code switching, and generalisability to different subject domains and paradigms. These paradigms move from task focused utterances (as in ATIS and SNIPS) to conversational, instructional (IoT, robotic, or vehicle instruction), or third party (surveying in-game chat).

Joint Learning Evaluation. It is clearly shown that joint learning between intent classification and slot filling is better than the individual training models. What is less explored is any aspect of *how* each sub-task improves performance in the other. Recent work in separating the input encoding, and the paths through the circuit, for slot and intent, should add clarity to this research. Experiments investigating the effect of adjusting aspects of the network on either side can be measured on the separate or joint metrics. These can be reported via correlation analysis metrics, like p value or Receiver Operator Characteristic (ROC) curve, and should deliver deeper understanding of the interactions.

REFERENCES

[1] Frédéric Béchet and Christian Raymond. 2018. Is ATIS too shallow to go deeper for benchmarking spoken language understanding models? In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)*. ISCA, 1–5.

[2] Valentina Bellomaria, Giuseppe Castellucci, Andrea Favalli, and Raniero Romagnoli. 2019. Almawave-SLU: A new dataset for SLU in Italian. In *Proceedings of the 6th Italian Conference on Computational Linguistics*. AILC.

[3] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya. 2013. Easy contextual intent prediction and slot detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8337–8341.

[4] Anmol Bhasin, Bharatram Natarajan, Gaurav Mathur, Joo Hyuk Jeon, and Jun-Seong Kim. 2019. Unified parallel intent and slot prediction with cross fusion and slot masking. In *Natural Language Processing and Information Systems*, Elisabeth Métais, Farid Meziane, Sunil Vadera, Vijayan Sugumaran, and Mohamad Saraee (Eds.). Springer, Cham, 277–285.

[5] Anmol Bhasin, Bharatram Natarajan, Gaurav Mathur, and Himanshu Mangla. 2020. Parallel intent and slot prediction using MLB fusion. In *Proceedings of the 14th International Conference on Semantic Computing (ICSC'20)*. IEEE, San Diego, 217–220.

[6] Hemanthage S. Bhathiya and Uthayasanker Thayasivam. 2020. Meta learning for few-shot joint intent detection and slot-filling. In *Proceedings of the 5th International Conference on Machine Learning Technologies (ICMLT'20)*. Association for Computing Machinery, New York, NY, 86–92.

[7] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., Lake Tahoe, CA, 2787–2795.

[8] Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint BERT-based model. arXiv:1907.02884. Retrieved from https://arxiv.org/abs/1907.02884.

[9] Asli Celikyilmaz and Dilek Hakkani-Tur. 2012. A joint model for discovery of aspects in utterances. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 330–338.

[10] Mengyang Chen, Jin Zeng, and Jie Lou. 2019. A self-attention joint model for spoken language understanding in situational dialog applications. arXiv:1905.11393. Retrieved from https://arxiv.org/abs/1905.11393.

[11] Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. arXiv:1902.10909. Retrieved from https://arxiv.org/abs/1902.10909.

[12] Sixuan Chen and Shuai Yu. 2019. WAIS: Word attention for joint intent detection and slot filling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, 9927–9928.

[13] Yun-Nung Chen, Dilek Hakanni-TÃijr, Gokhan Tur, Asli Celikyilmaz, Jianfeng Guo, and Li Deng. 2016. Syntax or semantics? Knowledge-guided joint semantic frame parsing. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'16)*. IEEE, 348–355.

[14] Lizhi Cheng, Weijia Jia, and Wenmian Yang. 2021. An effective non-autoregressive model for spoken language understanding. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21)*. Association for Computing Machinery, New York, NY, 241–250.

[15] Lizhi Cheng, Wenmian Yang, and Weijia Jia. 2021. A result based portable framework for spoken language understanding. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'21)*. IEEE, Los Alamitos, CA, 1–6.

[16] 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. arXiv:1805.10190. Retrieved from https://arxiv.org/abs/1805.10190.

[17] Slawomir Dadas, Jaroslaw Protasiewicz, and Witold Pedrycz. 2019. A deep learning model with data enrichment for intent detection and slot filling. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 3012–3018. https://doi.org/10.1109/SMC.2019.8914542

[18] Fatima Daha and Saniika Hewavitharana. 2019. Deep neural architecture with character embedding for semantic frame detection. In *Proceedings of the IEEE 13th International Conference on Semantic Computing (ICSC'19)*. IEEE, 302–307.

[19] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Proceedings of the Workshop on Human Language Technology*. Association for Computational Linguistics, 43–48.

[20] Anoop Deoras and Ruhi Sarikaya. 2013. Deep belief network based semantic taggers for spoken language understanding. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'13)*. ISCA, 2713–2717.

[21] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. Survey on evaluation methods for dialogue systems. *Artif. Intell. Rev.* 54 (2021), 755–810.

[22] 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*. Association for Computational Linguistics, 5467–5471.

[23] Mauajama Firdaus, Shobhit Bhatnagar, Asif Ekbal, and Pushpak Bhattacharyya. 2018. A deep learning based multitask ensemble model for intent detection and slot filling in spoken language understanding. In *Neural Information Processing*, Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa (Eds.). Springer, Cham, 647–658.

[24] Mauajama Firdaus, Shobhit Bhatnagar, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Intent detection for spoken language understanding using a deep ensemble model. In *Lecture Notes in Computer Science*. Springer, Cham, 629–642. https://doi.org/10.1007/978-3-319-97304-3_48

[25] Mauajama Firdaus, Hitesh Golchha, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A deep multi-task model for dialogue act classification, intent detection and slot filling. *Cogn. Comput.* 13 (2020), 626–645.

[26] Mauajama Firdaus, Ankit Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2019. A multi-task hierarchical approach for intent detection and slot filling. *Knowl.-Bas. Syst.* 183 (2019), 104846.

[27] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 564–569. https://doi.org/10.18653/v1/N19-1055

[28] 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)*. Association for Computational Linguistics, 753–757.

[29] Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'14)*. IEEE, 554–559.

[30] Arshit Gupta, John Hewitt, and Katrin Kirchhoff. 2019. Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, 46–55. https://doi.org/10.18653/v1/W19-5906

[31] Arshit Gupta, Peng Zhang, Garima Lalwani, and Mona Diab. 2019. CASA-NLU: Context-aware self-attentive natural language understanding for task-oriented chatbots. 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'19)*. Association for Computational Linguistics, 1285–1290. https://doi.org/10.18653/v1/D19-1127

[32] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung 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 Conference of the International Speech Communication Association (INTERSPEECH'16)*. ISCA, 715–719.

[33] Soyeon Caren Han, Siqu Long, Huichun Li, Henry Weld, and Josiah Poon. 2021. Bi-directional joint neural networks for intent classification and slot filling. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'21)*. ISCA, 4743–4747.

[34] Ting He, Xiaohong Xu, Yating Wu, Huazhen Wang, and Jian Chen. 2021. Multitask learning with knowledge base for joint intent detection and slot filling. *Appl. Sci.* 11, 11 (2021). https://doi.org/10.3390/app11114887

[35] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language (HLT'90)*. Association for Computational Linguistics, 96–101. https://doi.org/10.3115/116580.116613

[36] Lynette Hirschman. 1992. Multi-site data collection for a spoken language corpus—MAD COW. In *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP'92)*. International Speech Communication Association, 903–906.

[37] Lixian Hou, Yanling Li, Chengcheng Li, and Min Lin. 2019. Review of research on task-oriented spoken language understanding. *J. Phys.: Conf. Ser.* 1267 (July 2019), 012023.

[38] Zhiqi Huang, Fenglin Liu, Peilin Zhou, and Yuexian Zou. 2021. Sentiment injected iteratively co-interactive network for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. IEEE, 7488–7492. https://doi.org/10.1109/ICASSP39728.2021.9413885

[39] Yanfei Hui, Jianzong Wang, Ning Cheng, Fengying Yu, Tianbo Wu, and Jing Xiao. 2021. Joint intent detection and slot filling based on continual learning model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. IEEE, 7643–7647. https://doi.org/10.1109/ICASSP39728.2021.9413360

[40] Minwoo Jeong and Gary Geunbae Lee. 2008. Triangular-chain conditional random fields. *IEEE Trans. Aud. Speech Lang. Process.* 16, 7 (2008), 1287–1302.

[41] Sangkeun Jung, Jinsik Lee, and Jiwon Kim. 2018. Learning to embed semantic correspondence for natural language understanding. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 131–140. https://doi.org/10.18653/v1/K18-1013

[42] Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. ONENET: Joint domain, intent, slot prediction for spoken language understanding. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'17)*. IEEE, 547–553.

[43] Jason Krone, Yi Zhang, and Mona Diab. 2020. Learning to classify intents and slot labels given a handful of examples. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics, 96–108. https://doi.org/10.18653/v1/2020.nlp4convai-1.12

[44] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder LSTM for semantic slot filling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2077–2083. https://doi.org/10.18653/v1/D16-1223

[45] Jihwan Lee, Dongchan Kim, Ruhi Sarikaya, and Young-Bum Kim. 2018. Coupled representation learning for domains, intents and slots in spoken language understanding. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'18)*. IEEE, 714–719.

[46] Michał Lew, Aleksander Obuchowski, and Monika Kutyła. 2021. Improving intent detection accuracy through token level labeling. In *Proceedings of the 3rd Conference on Language, Data and Knowledge (LDK'21), Open Access Series in Informatics*,Vol. 93, Dagmar Gromann, Gilles Sérasset, Thierry Declerck, John P. McCrae, Jorge Gracia, Julia Bosque-Gil, Fernando Bobillo, and Barbara Heinisch (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 30:1–30:11. https://doi.org/10.4230/OASIcs.LDK.2021.30

[47] Changliang Li, Cunliang Kong, and Yan Zhao. 2018. A joint multi-task learning framework for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*. IEEE, Calgary, Canada, 6054–6058.

[48] Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3824–3833. https://doi.org/10.18653/v1/D18-1417

[49] Changliang Li, Yan Zhao, and Dong Yu. 2019. Conditional joint model for spoken dialogue system. In *Proceedings of the International Conference on Cognitive Computing (ICCC'19)*, Ruifeng Xu, Jianzong Wang, and Liang-Jie Zhang (Eds.). Springer, Cham, 26–36.

[50] Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2950–2962. https://doi.org/10.18653/v1/2021.eacl-main.257

[51] Shang-Wen Li, Jason Krone, Shuyan Dong, Yi Zhang, and Yaser Al-Onaizan. 2021. Meta learning to classify intent and slot labels with noisy few-shot examples. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'21)*. IEEE, 1004–1011.

[52] Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'16)*. ISCA, 685–689. https://doi.org/10.21437/Interspeech.2016-1352

[53] Bing Liu and Ian Lane. 2016. Joint online spoken language understanding and language modeling with recurrent neural networks. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'16)*. Association for Computational Linguistics, 22–30.

[54] Jiao Liu, Yanling Li, and Min Lin. 2019. Review of intent detection methods in the human-machine dialogue system. *J. Phys.: Conf. Ser.* 1267 (July 2019), 012059. https://doi.org/10.1088/1742-6596/1267/1/012059

[55] Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. CM-net: A novel collaborative memory network for spoken language understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 1051–1060. https://doi.org/10.18653/v1/D19-1097

[56] Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 1297–1303.

[57] Cedric Lothritz, Kevin Allix, Bertrand Lebichot, Lisa Veiber, Tegawendé F. Bissyandé, and Jacques Klein. 2021. Comparing multilingual and multiple monolingual models for intent classification and slot filling. In *Natural Language Processing and Information Systems*, Elisabeth Métais, Farid Meziane, Helmut Horacek, and Epaminondas Kapetanios (Eds.). Springer, Cham, 367–375.

[58] Samuel Louvan and Bernardo Magnini. 2019. Leveraging non-conversational tasks for low resource slot filling: Does it help? In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, 85–91. https://doi.org/10.18653/v1/W19-5911

[59] Mingbo Ma, Kai Zhao, Liang Huang, Bing Xiang, and Bowen Zhou. 2017. Jointly trained sequential labeling and classification by sparse attention neural networks. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'17)*. ISCA, 3334–3338.

[60] Françoise Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 4749–4752.

[61] Alaa Mohasseb, Mohamed Bader-El-Den, and Mihaela Cocea. 2018. Classification of factoid questions intent using grammatical features. *ICT Express* 4, 4 (December 2018), 239–242. https://doi.org/10.1016/j.icte.2018.10.004

[62] Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2021. Language model is all you need: Natural language understanding as question answering. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. IEEE, 7803–7807.

[63] Pin Ni, Yuming Li, Gangmin Li, and Victor Chang. 2020. Natural language understanding approaches based on joint task of intent detection and slot filling for IoT voice interaction. *Neural Comput. Appl.* 32 (2020), 16149–16166.

[64] Jingcheng Niu and Gerald Penn. 2019. Rationally reappraising ATIS-based dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5503–5507.

[65] Eda Okur, Shachi H. Kumar, Saurav Sahay, Asli Arslan Esme, and Lama Nachman. 2019. Natural Language Interactions in Autonomous Vehicles: Intent Detection and Slot Filling from Passenger Utterances. arXiv:1904.10500 [cs.CL]. Retrieved from https://arxiv.org/abs/org/1904.10500.

[66] David S. Pallett, Nancy L. Dahlgren, Jonathan G. Fiscus, William M. Fisher, John S. Garofolo, and Brett C. Tjaden. 1992. DARPA february 1992 ATIS benchmark test results. In *Proceedings of the Workshop on Speech and Natural Language (HLT'91)*. Association for Computational Linguistics, 15–27.

[67] Lingfeng Pan, Yi Zhang, Feiliang Ren, Yining Hou, Yan Li, Xiaobo Liang, and Yongkang Liu. 2018. A multiple utterances based neural network model for joint intent detection and slot filling. In *Proceedings of the Evaluation Tasks at the China Conference on Knowledge Graph and Semantic Computing (CCKS'18)*. CEUR-WS.org, 25–33.

[68] Shiva Pentyala, Mengwen Liu, and Markus Dreyer. 2019. Multi-task networks with universe, group, and task feature learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 820–830. https://doi.org/10.18653/v1/P19-1079

[69] 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 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 2078–2087. https://doi.org/10.18653/v1/D19-1214

[70] 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. IEEE, 8193–8197. https://doi.org/10.1109/ICASSP39728.2021.9414110

[71] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3853–3860. https://doi.org/10.24963/ijcai.2020/533

[72] 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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 1807–1816. https://doi.org/10.18653/v1/2020.findings-emnlp.163

[73] Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'15)*. ISCA, 135–139.

[74] Avik Ray, Yilin Shen, and Hongxia Jin. 2018. Robust spoken language understanding via paraphrasing. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)*. ISCA, 3454–3458. https://doi.org/10.21437/Interspeech.2018-2358

[75] Avik Ray, Yilin Shen, and Hongxia Jin. 2019. Iterative delexicalization for improved spoken language understanding. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'19)*. ISCA, 1183–1187. https://doi.org/10.21437/Interspeech.2019-2955

[76] Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'07)*. ISCA, 1605–1608.

[77] Evgeniia Razumovskaia, Goran Glavaš, Olga Majewska, Edoardo M. Ponti, Anna Korhonen, and Ivan Vulić. 2021. Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems. arXiv:2104.08570. Retrieved from https://arxiv.org/abs/2104.08570.

[78] Fuji Ren and Siyuan Xue. 2020. Intention detection based on siamese neural network with triplet loss. *IEEE Access* 8 (2020), 82242–82254.

[79] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 3795–3805. https://doi.org/10.18653/v1/N19-1380

[80] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A survey of available corpora for building data-driven dialogue systems: The journal version. *Dialog. Discourse* 9, 1 (2018), 1–49.

[81] Yilin Shen, Wenhu Chen, and Hongxia Jin. 2019. Interpreting and improving deep neural SLU models via vocabulary importance. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'19)*. ISCA, 1328–1332. https://doi.org/10.21437/Interspeech.2019-3184

[82] Yilin Shen, Xiangyu Zeng, and Hongxia Jin. 2019. A progressive model to enable continual learning for semantic slot filling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 1279–1284. https://doi.org/10.18653/v1/D19-1126

[83] Yilin Shen, Xiangyu Zeng, Yu Wang, and Hongxia Jin. 2018. User information augmented semantic frame parsing using progressive neural networks. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)*, B. Yegnanarayana (Ed.). ISCA, 3464–3468.

[84] Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng. 2015. Contextual spoken language understanding using recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'15)*. IEEE, 5271–5275.

[85] Aditya Siddhant, Anuj Kumar Goyal, and Angeliki Metallinou. 2019. Unsupervised transfer learning for spoken language understanding in intelligent agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, 4959–4966. https://doi.org/10.1609/aaai.v33i01.33014959

[86] Ieva Staliūnaitė and Ignacio Iacobacci. 2020. Auxiliary capsules for natural language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*. IEEE, 8154–8158.

[87] Rui Sun, Lu Rao, and Xingfa Zhou. 2021. Bidirectional information transfer scheme for joint intent detection and slot filling. In *Proceedings of the 17th International Conference on Computational Intelligence and Security (CIS'21)*. IEEE, 333–337. https://doi.org/10.1109/CIS54983.2021.00076

[88] Yik-Cheung Tam, Yangyang Shi, Hunk Chen, and Mei-Yuh Hwang. 2015. RNN-based labeled data generation for spoken language understanding. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'15)*. ISCA, 125–129.

[89] Hao Tang, Donghong Ji, and Qiji Zhou. 2020. End-to-end masked graph-based CRF for joint slot filling and intent detection. *Neurocomputing* 413 (2020), 348–359. https://doi.org/10.1016/j.neucom.2019.06.113

[90] Shimin Tao, Ying Qin, Yimeng Chen, Chunning Du, Haifeng Sun, Weibin Meng, Yanghua Xiao, Jiaxin Guo, Chang Su, Minghan Wang, Min Zhang, Yuxia Wang, and Hao Yang. 2021. Incorporating complete syntactical knowledge for spoken language understanding. In *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction*, Bing Qin, Zhi Jin, Haofen Wang, Jeff Pan, Yongbin Liu, and Bo An (Eds.). Springer Singapore, Singapore, 145–156.

[91] Quynh Ngoc Thi Do and Judith Gaspers. 2019. Cross-lingual transfer learning for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*. IEEE, 5956–5960.

[92] Gokhan Tur, Asli Celikyilmaz, Xiaodong He, Dilek Hakkani-Tür, and Li Deng. 2018. Deep learning in conversational language understanding. In *Deep Learning in Natural Language Processing*, Li Deng and Yang Liu (Eds.). Springer Singapore, Singapore, 23–48. https://doi.org/10.1007/978-981-10-5209-5_2

[93] Gokhan Tur and Renato De Mori. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons, New York, NY.

[94] Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in ATIS? In *Proceedings of the IEEE Spoken Language Technology Workshop*. IEEE, 19–24.

[95] Gokhan Tur, Dilek Hakkani-Tur, Larry Heck, and S. Parthasarathy. 2011. Sentence simplification for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11)*. IEEE, 5628–5631.

[96] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 5998–6008.

[97] Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'16)*. IEEE, 6060–6064. https://doi.org/10.1109/ICASSP.2016.7472841

[98] Congrui Wang, Zhen Huang, and Minghao Hu. 2020. SASGBC: Improving sequence labeling performance for joint learning of slot filling and intent detection. In *Proceedings of the 6th International Conference on Computing and Data Engineering (ICCDE'20)*. Association for Computing Machinery, New York, NY, 29–33.

[99] Xiaojie Wang and Caixia Yuan. 2016. Recent advances on human-computer dialogue. *CAAI Trans. Intell. Technol.* 1, 4 (October 2016), 303–312. https://doi.org/10.1016/j.trit.2016.12.004

[100] Yu Wang, Yue Deng, Yilin Shen, and Hongxia Jin. 2020. A new concept of multiple neural networks structure using convex combination. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 11 (2020), 9539–9546.

[101] Yufan Wang, Tingting He, Rui Fan, Wenji Zhou, and Xinhui Tu. 2019. Effective utilization of external knowledge and history context in multi-turn spoken language understanding model. In *Proceedings of the IEEE International Conference on Big Data (Big Data'19)*. IEEE, Los Angeles, USA, 960–967.

[102] Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 309–314. https://doi.org/10.18653/v1/N18-2050

[103] Yufan Wang, Li Tang, and Tingting He. 2018. Attention-based CNN-BLSTM networks for joint intent detection and slot filling. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Maosong Sun, Ting Liu, Xiaojie Wang, Zhiyuan Liu, and Yang Liu (Eds.). Springer, Cham, 250–261.

[104] Ye-Yi Wang. 2010. Strategies for statistical spoken language understanding with small amount of data—An empirical study. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'10)*. ISCA, 2498–2501.

[105] Henry Weld, Guanghao Huang, Jean Lee, Tongshu Zhang, Kunze Wang, Xinghong Guo, Siqu Long, Josiah Poon, and Caren Han. 2021. CONDA: A CONtextual dual-annotated dataset for in-game toxicity understanding and detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2406–2416. https://doi.org/10.18653/v1/2021.findings-acl.213

[106] Liyun Wen, Xiaojie Wang, Zhenjiang Dong, and Hong Chen. 2018. Jointly modeling intent identification and slot filling with contextual and hierarchical information. In *Natural Language Processing and Chinese Computing*, Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong (Eds.). Springer, Cham, 3–15.

[107] Cong Xu, Qing Li, Dezheng Zhang, Jiarui Cui, Zhenqi Sun, and Hao Zhou. 2020. A model with length-variable attention for spoken language understanding. *Neurocomputing* 379 (2020), 197–202.

[108] Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*. IEEE, 78–83.

[109] Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 5052–5063. https://doi.org/10.18653/v1/2020.emnlp-main.410

[110] Xuesong Yang, Yun-Nung Chen, Dilek Hakkani-Tür, Paul Crook, Xiujun Li, Jianfeng Gao, and Li Deng. 2017. End-to-end joint learning of natural language understanding and dialogue manager. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17)*. IEEE, 5690–5694.

[111] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT'14)*. IEEE, 189–194. https://doi.org/10.1109/SLT.2014.7078572

[112] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014. Recurrent conditional random field for language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14)*. IEEE, 4077–4081.

[113] Dong Yu, Shizhen Wang, and Li Deng. 2010. Sequential labeling using deep-structured conditional random fields. *IEEE J. Select. Top. Sign. Process.* 4, 6 (2010), 965–973. https://doi.org/10.1109/JSTSP.2010.2075990

[114] Shuai Yu, Lei Shen, Pengcheng Zhu, and Jiansong Chen. 2018. ACJIS: A novel attentive cross approach for joint intent detection and slot filling. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'18)*. IEEE, 1–7.

[115] Yulan He and Steve Young. 2003. A data-driven spoken language understanding system. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 583–588.

[116] Chenwei Zhang, Wei Fan, Nan Du, and Philip S. Yu. 2016. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *Proceedings of the 25th International Conference on World Wide Web (WWW'16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1373–1384. https://doi.org/10.1145/2872427.2874810

[117] 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*. Association for Computational Linguistics, 5259–5267. https://doi.org/10.18653/v1/P19-1519

[118] Dongjie Zhang, Zheng Fang, Yanan Cao, Yanbing Liu, Xiaojun Chen, and Jianlong Tan. 2018. Attention-based RNN model for joint extraction of intent and word slot based on a tagging strategy. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'18)*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer, Cham, 178–188.

[119] Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Hou-Feng Wang. 2020. Graph LSTM with context-gated mechanism for spoken language understanding. In *Proceedings of the AAAI Annual Conference on Artificial Intelligence (AAAI'20)*. AAAI Press, 9539–9546.

[120] Linhao Zhang and Houfeng Wang. 2019. Using bidirectional transformer-CRF for spoken language understanding. In *Natural Language Processing and Chinese Computing*, Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan (Eds.). Springer, Cham, 130–141.

[121] Shuyou Zhang, Junjie Jiang, Zaixing He, Xinyue Zhao, and Jinhui Fang. 2019. A novel slot-gated model combined with a key verb context feature for task request understanding by service robots. *IEEE Access* 7 (2019), 105937–105947.

[122] Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 2993–2999.

[123] Zhichang Zhang, Zhenwen Zhang, Haoyuan Chen, and Zhiman Zhang. 2019. A joint learning framework with BERT for spoken language understanding. *IEEE Access* 7 (2019), 168849–168858.

[124] Xinlu Zhao, E. Haihong, and Meina Song. 2018. A joint model based on CNN-LSTMs in dialogue understanding. In *Proceedings of the International Conference on Information Systems and Computer Aided Education (ICISCAE'18)*. IEEE, 471–475.

[125] Yang Zheng, Yongkang Liu, and John H. L. Hansen. 2017. Intent detection and semantic parsing for navigation dialogue language processing. In *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC'17)*. IEEE, 1–6.

[126] Peilin Zhou, Zhiqi Huang, Fenglin Liu, and Yuexian Zou. 2021. PIN: A novel parallel interactive network for spoken language understanding. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR'21)*. IEEE, Los Alamitos, CA, 2950–2957. https://doi.org/10.1109/ICPR48806.2021.9411948

[127] Qianrong Zhou, Liyun Wen, Xiaojie Wang, Long Ma, and Yue Wang. 2016. A hierarchical LSTM model for joint tasks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Maosong Sun, Xuanjing Huang, Hongfei Lin, Zhiyuan Liu, and Yang Liu (Eds.). Springer, Cham, 324–335.

[128] Su Zhu and Kai Yu. 2017. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17)*. IEEE, 5675–5679. https://doi.org/10.1109/ICASSP.2017.7953243

Received 4 October 2021; revised 6 June 2022; accepted 20 June 2022