Recent Advances in End-to-End Spoken Language Understanding

Natalia Tomashenko, Antoine Caubrière, Yannick Estève, Antoine Laurent, Emmanuel Morin

Recent Advances in End-to-End Spoken Language Understanding

Natalia Tomashenko[1], Antoine Caubrière[2], Yannick Estève[1], Antoine Laurent[2], and Emmanuel Morin[3]

[1] LIA, University of Avignon, France
{natalia.tomashenko, yannick.esteve}@univ-avignon.fr
[2] LIUM, University of Le Mans, France
{antoine.caubriere,antoine.laurent}@univ-lemans.fr
[3] LS2N, University of Nantes
emmanuel.morin@univ-nantes.fr

Abstract. This work deals with spoken language understanding (SLU) systems in the scenario when the semantic information is extracted directly from the audio speech signal by means of a single end-to-end neural network model. We consider two SLU tasks: named entity recognition (NER) and semantic slot filling (SF). For these tasks, in order to improve the model performance, we explore various strategies including speaker adaptive training and sequential pretraining schemes.

Keywords: Spoken language understanding (SLU) · Acoustic adaptation · End-to-end SLU · Slot filling · Named entity recognition.

1 Introduction

Spoken language understanding (SLU) is an important component of dialog systems. Traditional SLU systems consist of at least two components: (1) an automatic speech recognition (ASR) system that transcribes acoustic speech signal into word sequences and (2) a natural language understanding (NLU) system which predicts, given the output of the ASR system, named entities, semantic or domain tags, and other language characteristics depending on the considered task. In classical approaches, these two systems are often built and optimized independently.

Recent advances in deep learning have impacted many research and industrial domains and in particular have boosted the progress in conversational artificial intelligence (AI) and its applications. Most of the state-of-the art SLU and conversational AI systems employ neural network models [11]. Nowadays there is a great interest of the research community in end-to-end systems for various speech and language technologies. A few recent papers [22, 17, 26, 12, 5, 19] present ASR-free end-to-end approaches for SLU tasks and show promising results. These methods aim to learn SLU models from acoustic signal without intermediate text representation. Paper [5] proposed an audio-to-intent architecture for semantic classification in dialog systems. An encoder-decoder framework [28] is used in paper [26] for domain and intent classification, and in [17] for domain, intent, and argument recognition. A different approach based on the model trained with the Connectionist Temporal Classification (CTC) criterion [14] was proposed in [12] for named entity recognition (NER) and slot filling. End-to-end methods

are motivated by the following factors: (1) possibility of better information transfer from the speech signal due to the joint optimization on the final objective function, and, in particular, leveraging errors from the ASR system and focusing on the most important information; and (2) simplification of the overall system and elimination of some of its components. However, deep neural networks and especially end-to-end models often require more training data to be efficient. For SLU, this implies the demand of big semantically annotated corpora. In this work, we explore different ways to improve the performance of end-to-end SLU systems.

2 SLU tasks

In SLU for human-machine conversational systems, an important task is to automatically extract semantic concepts or to fill in a set of *slots* in order to achieve a goal in a human-machine dialogue. In this paper, we consider two SLU tasks: named entity recognition (NER) and semantic slot filling (SF). In the NER task, the purpose is to recognize information units such as names, including person, organization and location names, dates, events and others. In the SF task, the extraction of wider semantic information is targeted. These last years, NER and SF where addressed as word labelling problems, through the use of the classical BIO *(begin/inside/outside)* notation [23]. For instance, "*I would like to book three double rooms in Paris for tomorrow*" will be represented for the NER and SF task as the following BIO labelled sentences:

- NER: "*I::∅ would::∅ like::∅ to::∅ book::∅ three::B-amount double::∅ rooms::∅ in::∅ Paris::B-location/city for::∅ tomorrow::B-time/date*".
- SF: "*I::B-command would::I-command like::I-command to::I-command book::I-command three::B-room/number double::B-room/type rooms::I-room/type in::∅ Paris::B-location/city for::∅ tomorrow::B-time/date*".

In this paper, similarly to [12], the BIO representation is abandoned in profit to a chunking approach. For instance for NER, the same sentence will be presented as:

- NER: "*I would like to book* <amount *three* > *double rooms in* <location/city *Paris* > *for* <time/date *tomorrow* >".

In this study, we train an end-to-end neural model to reproduce such textual representation from speech. Since our neural model emits characters, we use specific characters corresponding to each opening tag (one by named entity category or one by semantic concept), while the same symbol is used to represent the closing tag.

3 Model training

End-to-end training of SLU models is realized through the recurrent neural network (RNN) architecture and CTC loss function [14] as shown in Figure 1. A spectrogram of power normalized audio clips calculated on 20ms windows is used as the input features for the system. As shown in Figure 1, it is followed by two 2D-invariant (in the time and-frequency domain) convolutional layers, and then by five BLSTM layers with sequence-wise batch normalization. A fully connected layer is applied after BLSTM layers, and

the output layer of the neural network is a softmax layer. The model is trained using the CTC loss function. The neural architecture is similar to the Deep Speech 2 [1] for ASR.

The outputs of the network depend on the task. For ASR, the outputs consist of graphemes of a corresponding language, a *space* symbol to denote word boundaries and a *blank* symbol. For NER, in addition to ASR outputs, we add outputs corresponding to named entity types and a closing symbol for named entities. In the same way, for SF task, we use all ASR outputs and additional tags corresponding to semantic concepts and a closing symbol for semantic tags.

In order to improve model training, we investigate speaker adaptive training (SAT), pretraining and transfer learning approaches. First, we formalize the $\star$-*mode*, that proved its effectiveness in all our previous and current experiments.

3.1 CTC loss function interpretation related to $\star$-*mode*

The CTC loss function [14] is relevant to train models for ASR without Hidden Markov Models. The $\star$-*mode* can be seen as a minor modification of the CTC loss function.

CTC loss function definition By means of a many-to-one $\mathcal{B}$ mapping function, CTC transforms a sequence of the network outputs, emitted for each acoustic frame, to a sequence of final target labels by deleting repeated output labels and inserting a *blank* (*no label*) symbol. The CTC loss function is defined as:

$$\mathcal{L}_{CTC} = -\sum_{(\mathbf{x},\mathbf{l})\in Z} \ln P(\mathbf{l}|\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is a sequence of acoustic observations, $\mathbf{l}$ is the target output label sequence, and Z the training dataset. $P(\mathbf{l}|\mathbf{x})$ is defined as:

$$P(\mathbf{l}|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{l})} P(\pi|\mathbf{x}), \tag{2}$$

where π is a sequence of initial output labels emitted by the model for each input frame. To compute $P(\pi|\mathbf{x})$ we use the probability of the output label π_t emitted by the neural model for frame t to build this sequence. This probability is modeled by the value $y^t_{\pi_t}$ given by the output node of the neural model related to the label π_t. $P(\pi|\mathbf{x})$ is defined as $P(\pi|\mathbf{x}) = \prod_t^T y^t_{\pi_t}$, where T denotes the number of frames.

CTC loss function and $\star$-*mode* In the framework of the $\star$-*mode*, we introduce a new symbol, "$\star$", that represents the presence of a label (the opposite of the *blank* symbol) that does not need to be disambiguated. We expect to build a model that is more discriminant on the important task-specific labels. For example, for the SF SLU task important labels are the ones corresponding to semantic concept opening and closing tags, and characters involved in the word sequences that support the value of these semantic concepts (*i.e* characters occurring between an opening and a closing concept tag). In the CTC loss function framework, the $\star$-*mode* consists in applying another kind of mapping function before $\mathcal{B}$. While $\mathcal{B}$ converts a sequence π of initial output labels

into the final sequence $\mathbf{l}$ to be retrieved, we introduce the mapping function $\mathcal{S}$ that is applied to each final target output label. Let C be the set of elements $\mathbf{l}_i$ included in subsequences $\mathbf{l}_a^b \subset \mathbf{l}$ such as $\mathbf{l}_a$ is an opening concept tag and $\mathbf{l}_b$ the associated closing tag; i, a and b are indexes that handle positions in sequence $\mathbf{l}$, and $a \leq i \leq b$. Let V be the vocabulary of all the symbols present in sequences $\mathbf{l}$ in Z, and let consider the new symbol $\star \notin V$. Let define $V^\star = V \cup \{\star\}$, and L (resp. $L^\star$) the set of all the label sequences that can be generated from V (resp. $V^\star$).

Considering n as the number of elements in $\mathbf{l}$, m an integer such as $m \leq n$, we define the mapping function $\mathcal{S} : L \rightarrow L^\star, \mathbf{l} \mapsto \mathbf{l}'$ in two steps:

$$\begin{array}{ll} 1.\ \forall \mathbf{l}_j \in \mathbf{l} & \begin{cases} \mathbf{l}_j \notin C \Rightarrow \mathbf{l'}_j = \star \\ \mathbf{l}_j \in C \Rightarrow \mathbf{l'}_j = \mathbf{l}_j \end{cases} \\ 2.\ \forall \mathbf{l'}_j \in \mathbf{l}' & \mathbf{l'}_{j-1} = \star \Rightarrow \mathbf{l'}_j = \emptyset \end{array} \tag{3}$$

By applying $\mathcal{S}$ on the last example sentence used in Section 2 for NER, this sentence is transformed to:

- sent: "*I would like to book* <amount *three* > *double rooms in* <location/city *Paris* > *for* <time/date *tomorrow* >".
- $\mathcal{S}$(sent): "* <amount *three* > * <location/city *Paris* > * <time/date *tomorrow* >".

To introduce $\star$*-mode* in the CTC loss function definition, we modify the formulation of $P(\mathbf{l}|\mathbf{x})$ in formula (2) by introducing the $\mathcal{S}$ mapping function applied to $\mathbf{l}$:

$$P(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1} \circ \mathcal{S}(\mathbf{l})} P(\pi|\mathbf{x}). \tag{4}$$

3.2 Speaker adaptive training

Differences between training and testing conditions may significantly reduce recognition accuracy in ASR systems and degrade performance of other speech-related technologies. Adaptation is an efficient way to reduce the mismatches between the models and the data from a particular speaker or channel. For many decades, acoustic model adaptation has been an essential component of any state-of-the-art ASR system. For end-to-end approaches, speaker adaptation is less studied, and most of the first end-to-end ASR systems do not use any speaker adaptation and are built on spectrograms [1] or filterbank features [2]. However, some recent works [7] demonstrated the effectiveness of speaker adaptation for end-to-end models.

For SLU tasks, there is also an emerging interest in the end-to-end models which have a speech signal as input. Thus, acoustic, and particularly speaker, adaptation for such models can play an important role in improving the overall performance of these systems. However, to our knowledge, there is no research on speaker adaptation for end-to-end SLU models, and the existing works do not use any speaker adaptation.

One way to improve SLU models which we investigate in this paper is speaker adaptation. We apply i-vector based speaker adaptation [25]. The proposed way of integration of i-vectors into the end-to-end model architecture is shown in Figure 1. Speaker i-vectors are appended to the outputs of the last (second) convolutional layer, just before the first recurrent (BLSTM) layer. In this paper, for better initialization, we first train a

model with *zero pseudo i-vectors* (all values are equal to 0). Then, we use this pretrained model and fine-tune it on the same data but with the real i-vectors. This approach was inspired by [6], where an idea of using zero auxiliary features during pretraining was implemented for language models and in our preliminary experiments it demonstrated better results than direct model training with i-vectors.

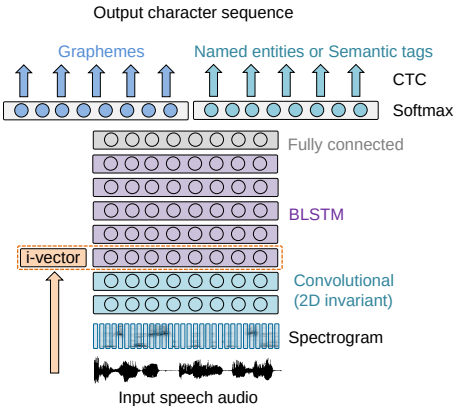

Fig. 1. Universal end-to-end deep neural network model architecture for ASR, NER and SF tasks. Depending on the current task, the set of the output characters (targets) consists of: (1) ASR: graphemes for a given language; (2) NER: graphemes and named entity tags; and (3) SF: graphemes and semantic SF tags.

3.3 Transfer learning

Transfer learning is a popular and efficient method to improve the learning performance of the target predictive function using knowledge from a different source domain [20]. It allows to train a model for a given target task using available out-of-domain source data, and hence to avoid an expensive data labeling process, which is especially useful in case of low-resource scenarios.

In this paper, for SF, we investigate the effectiveness of the transfer learning paradigm for various source domains and tasks: (1) ASR in the target and out-of-domain languages; (2) NER in the target language; (3) slot filling (SF). For all the tasks, we used similar model architectures (Section 4.2 and Figure 1). The difference is in the text data preparation and output targets. For training ASR systems, the output targets correspond to alphabetic characters and a *blank* symbol. For NER tasks, the output targets include all the ASR targets and targets corresponding to named entity tags. We have several symbols corresponding to named entities (in the text these characters are situated before the beginning of a named entity, which can be a single word or a sequence of several words) and a one tag corresponding to the end of the named entity, which is the same for all named entities.

Similarly, for SF tags, we use targets corresponding to the semantic concept tags and one tag corresponding to the end of a concept. Transfer learning is realized through the

chain of consequence model training on different tasks. For example, we can start from training an ASR model on audio data and corresponding text transcriptions. Then, we change the softmax layer in this model by replacing the targets with the SF targets and continue training on the corpus annotated with semantic tags. Further in the paper, we denote this type of chain as $ASR \rightarrow SF$. Models in this chain can be trained on different corpora, that can make this method especially useful in low-resource scenario when we do not have enough semantically annotated data to train an end-to-end model, but have sufficient amount of data annotated with more general concepts or only transcribed data. For NER, we also investigates the knowledge transfer from ASR.

Table 1. Corpus statistics for ASR, NER and SF tasks.

Task	**Corpora**	**Size,h**	**#Speakers**
ASR train	EPAC, ESTER 1,2, ETAPE, REPERE, DECODA, MEDIA, PORTMEDIA	404.6	12518
NER train	EPAC, ESTER 1,2, ETAPE, REPERE	323.8	7327
NER dev	ETAPE (dev)	6.6	152
NER test	ETAPE (test), Quaero (test)	12.3	474
SF train	1. MEDIA (train),	16.1	727
	2. PORTMEDIA (train)	7.2	257
SF dev	MEDIA (dev)	1.7	79
SF test	MEDIA (test)	4.8	208

4 Experiments

4.1 Data

Several publicly available corpora have been used for experiments (see Table 1).

ASR data The corpus for ASR training was composed of corpora from various evaluation campaigns in the field of automatic speech processing for French, as shown in Table 1. The EPAC [9], ESTER 1,2 [10], ETAPE [15], REPERE [13] contain transcribed speech in French from TV and radio broadcasts. These data were originally in the microphone channel and for experiments in this paper were downsampled from 16kHz to 8kHz, since the test set for our main target task (SF) consists of telephone conversations. The DECODA [3] corpus is composed of dialogues from the call-center of the Paris transport authority. The MEDIA [8, 4] and PORTMEDIA [18] are corpora of dialogues simulating a vocal tourist information server. The target language in all experiments is French. For experiments with transfer learning from ASR built in a different source language (English in our case) to SF in the target language, we used the TED-LIUM corpus [24]. This publicly available dataset contains 1495 TED talks in English that amount to 207 hours speech data from 1242 speakers, 16kHz. For experiments, we downsampled the audio data to 8kHz.

NER data To train the NER system, we used the following corpora: EPAC, ESTER 1,2, ETAPE, and REPERE. These corpora contain speech with text transcriptions and named entity annotation. The named entity annotation is performed following the methodology

of the Quaero project [16]. The taxonomy is composed of 8 main types: *person, function, organization, location, product, amount, time*, and *event*. Each named entity can be a single word or a sequence of several words. The total amount of annotated data is 112 hours. Based on this data, a classical NER system was trained using *NeuroNLP2*[4] to automatically extract named entities for the rest 212 hours of the training corpus. This was done in order to increase the amount of the training data for NER. Thus, the total amount of audio data to train the NER system is about 324 (112+212) hours. The development part of the ETAPE corpus was used for development, and as a test set we used the ETAPE test and Quaero test datasets.

SF data The following two French corpora, dedicated to semantic extraction from speech in a context of human/machine dialogues, were used in the current experiments: MEDIA and PORTMEDIA (see Table 1). The corpora have manual transcription and conceptual annotation. A concept is defined by a label and a value, for example with the concept *date*, the value *2001/02/03* can be associated [29, 8]. The MEDIA corpus is related to the hotel booking domain, and its annotation contains 76 semantic tags: *room number, hotel name, location, date, room equipment*, etc. The PORTMEDIA corpus is related to the theater ticket reservation domain and its annotation contains 35 semantic tags which are very similar to the tags used in the MEDIA corpus. For joint training on these corpora, we used a combined set of 86 semantic tags.

4.2 Models

We used the *deepspeech.torch* implementation[5] for training speaker independent models, and our modification of this implementation to integrate speaker adaptation. The open-source *Kaldi* toolkit [21] was used to extract 100-dimensional speaker i-vectors. All models had similar topology (except for the number of outputs) shown in Figure 1 for SAT models. Speaker independent models were trained in the same way, but without i-vector integration. Input features are spectrograms. They are followed by two 2D-invariant (in the time and-frequency domain) convolutional layers[6], and then by five 800-dimensional BLSTM layers with sequence-wise batch normalization. A fully connected layer is applied after BLSTM layers, and the output layer of the neural network is a softmax layer. The size of the output layer depends on the task (see Section 4.3). The model is trained using the CTC loss function.

4.3 Tasks

The target tasks for us are NER and SF. For each of this task, other tasks can be used for knowledge transfer. To train NER, we use ASR for transfer learning. To train SF, we use ASR on French and English, NER and another auxiliary SF task for transfere learning. Hence, we consider the following set of tasks:

[4] https://github.com/XuezheMax/NeuroNLP2

[5] https://github.com/SeanNaren/deepspeech.pytorch

[6] With parameters: kernel size=(41, 11), stride=(2, 2), padding=(20, 5)

- ASR_F – French ASR with 43 outputs {French characters, *blank* symbol}.
- ASR_E – English ASR with 28 outputs {English characters, *blank* symbol}.
- NER – French NER with 52 outputs {43 outputs from ASR_F, 8 outputs corresponding to named entity tags, 1 output corresponding to the closing tag for all named entities}.
- SF_1 – target SF task with 130 outputs {43 outputs from ASR_F , 86 outputs for semantic slot tags, 1 output for the closing tag}; trained on the training part of the MEDIA corpus.
- SF_{1+2} – auxiliary SF task; trained on the MEDIA plus PORTMEDIA training corpora.

For the target tasks NER and SF_1, we also considered $\star$-mode (Section 3.1), denoted respectively $NER^\star$ and $SF_1^\star$.

4.4 Results for NER

Performance of NER was evaluated in terms of *precision*, *recall*, and *F-measure*. Results for different training chains for speaker-independent (SI) and speaker adaptive training models (SAT) are given in Table 2. We can see, that pretraining with ASR_F task does not lead to significant improvement in performance. When the $NER^\star$ is added to the training chain, it improves all the evaluation measures. In particular, F-measure is increased by 1.9% absolute. For each training chain, we trained a corresponding chain with speaker adaptation. Results for SAT models are given in the right part of Table 2. We can see, that for all training chains, SAT models outperform SI models. The best result with SAT (F-measure 71.8%) outperforms the best SI result by 1.1% absolute.

Table 2. NER results on the test dataset in terms of Precision (P,%), Recall (R,%) and F-measure (F, %) for SI and SAT models.

Model training	SI			SAT		
	P	**R**	**F**	**P**	**R**	**F**
NER	78.9	60.7	68.6	80.9	60.9	69.5
$ASR_F \rightarrow NER$	80.5	60.0	68.8	80.2	61.7	69.7
$ASR_F \rightarrow NER \rightarrow NER^\star$	82.1	62.1	70.7	**83.1**	**63.2**	**71.8**

4.5 Results for SF

SF performance was evaluated in terms of *F-measure*, *concept error rate* (CER) and *concept value error rate* (CVER). Training performance on the MEDIA development dataset in terms of *character error rate* (CER) is shown in Figure 2 for different transfer learning chains for SI and SAT models. The blue curves SF_1 corresponds to the SI baseline model when the model was directly trained on the target SF task without pretraining. All curves of other colours correspond to different sequential transfer learning chains. We can observe, that all considered transfer learning schemes substantially improve the training performance. By comparing SF_1 and SF_{1+2}, we can conclude that training on the auxiliary task improves the performance. However, when we further trained this model on the target task (chain: $SF_{1+2} \rightarrow SF_1$), the performance continued

to improve. This demonstrates, that in given conditions, the sequence transfer learning provides better improvement than just joint training. The best SI model is obtained through the following training chain: $ASR_F \rightarrow SF_{1+2} \rightarrow SF_1$. These results are confirmed further in Table 3. Also, we can see that SAT gives an additional improvement in performance for all the models. For better models the improvement from SAT is less noticeable, than for the worse ones.

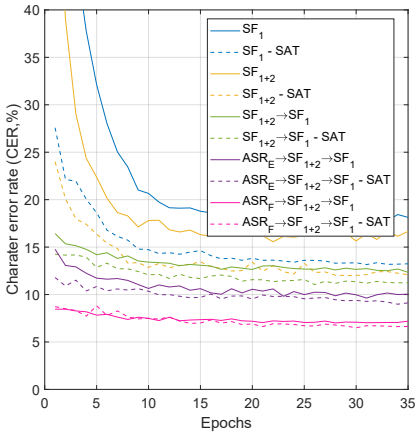

Fig. 2. Training performance on the MEDIA development dataset in terms of character error rate (CER) for training SF models. For each type of the model chain, a solid line corresponds to a speaker-independent model, and a dash line of the same colour denotes a speaker adaptive training (SAT) version of a given model.

Results for different training chains for speaker-independent (SI) models on the test set are given in Table 3 (#1–8). The first line SF_1 shows the baseline result on the test MEDIA dataset for the SF task, when a model was trained directly on the target task using in-domain data for this task (training part of the MEDIA corpus). The second line SF_{1+2} corresponds to the case when the model was trained on the auxiliary SF task. Other lines in the table correspond to different training chains described in Section 3.3. In $\#4$, we can see a chain that starts from training an ASR model for English. We can observe that using a pretrained ASR model from a different language can significantly (16.2% of relative CER reduction) improve the performance of the SF model (#4 vs #3). This result is noticeable since it shows that we can take benefit from linguistic resources from another language in case of lack of data for the target one. Using an ASR model trained in French (#5) provides better improvement: 36.0% of relative CER reduction (#5 vs #3). When we start the training process from a NER model (#6) we can observe slightly better results. Further, for the best two model training chains (#5 and 6) we trained corresponding models in $\star$-mode (#7 and 8). Results with speaker adaptation for four best models are shown in the right part of Table 3 (#9–12). We can see that SAT models show better results than SI ones. For CVER, we can observe a similar tendency. The results for the best models using beam search and a 4-gram LM are shown in brackets in blue. The LM was built on the texts including "$\star$". Finally, Table 4

Table 3. SF performance results on the MEDIA test dataset for end-to-end SF models trained with different transfer learning approaches. Results are given in terms of F-measure (F), CER and CVER metrics (%); $\mathbf{SF}_1$ – target task; $\mathbf{SF}_{1+2}$ – auxiliary task; **F** and **E** refer to the languages. For the best models, the results in blue correspond to decoding using beam search with a LM.

Model training	**SI**				**SAT**			
	#	**F**	**CER**	**CVER**	**#**	**F (LM)**	**CER (LM)**	**CVER (LM)**
SF_1	1	72.5	39.4	52.7				
SF_{1+2}	2	73.2	39.0	50.1				
$SF_{1+2} \rightarrow SF_1$	3	77.4	33.9	44.9				
$ASR_E \rightarrow SF_{1+2} \rightarrow SF_1$	4	81.3	28.4	37.3				
$ASR_F \rightarrow SF_{1+2} \rightarrow SF_1$	5	85.9	21.7	28.4	9	87.5	19.4	25.4
$NER \rightarrow SF_{1+2} \rightarrow SF_1$	6	86.4	20.9	27.5	10	87.3	19.5	26.0
$ASR_F \rightarrow SF_{1+2} \rightarrow SF_1^\star$	7	85.9	21.2	27.9	11	**87.7 (89.2)**	18.8 (16.5)	25.5 **(20.8)**
$NER \rightarrow SF_{1+2} \rightarrow SF_1^\star$	8	87.1	19.5	27.0	12	87.6 (89.2)	**18.6 (16.2)**	**24.6 (20.8)**

Table 4. SF performance results on the MEDIA test dataset for different systems.

Systems in literature:	**CER**	**Systems in this paper:**	**CER**
Pipeline: ASR+SLU, [27]	19.9	—gready mode	18.6
End-to-end, [12]	27.0	—beam search with LM	16.2

resumes our best results (in greedy and beam search modes) and shows the comparison results on the MEDIA dataset from other works [27, 12]. We can see, that the reported results significantly outperform the results reported in the literature for the current task.

Error analysis In the training corpus, different semantic concepts have different number of samples, that may impact the SF performance. Figure 3 demonstrates the relation between the concept error rate (CER) of a particular semantic concept and its frequency in the training corpus. Each point in Figure 3 corresponds to a particular semantic concept. For rare tags, the distribution of errors has larger variance and means than for more frequent tags. In addition, we are interested in the distribution of different types of SF errors (*deletions*, *insertions* and *substitutions*), which is shown in the form of a confusion matrix in Figure 4. For better representation, we first ordered the concepts in descending order by the total number of errors. Then, we chose the first 36 concepts which have the biggest number of errors. The total amount of errors of the chosen 36 concepts corresponds to 90% of all the errors for all concepts in the test MEDIA dataset. The diagonal corresponds to the correctly detected concepts and other elements (except for the last row and last column) correspond to the substitution errors. The final raw represents insertion errors and the final column – deletions. Each element in the matrix shows the total number of the corresponding events ('correctly recognized concept', 'substitution', 'deletion' or 'insertion') normalized by the total number of such events in the row. The most frequent errors are deletions (50% of all errors), then substitutions (32.3%) and insertions (17.7%).

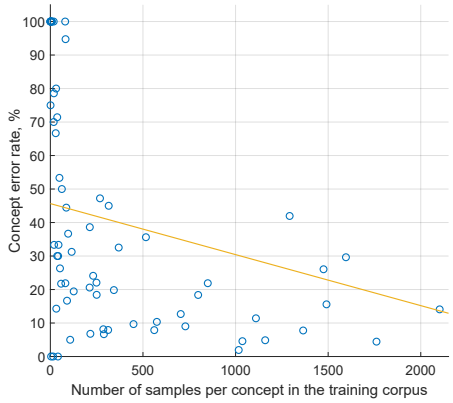

Fig. 3. Concept error rate (CER,%) results on the MEDIA test dataset for different concepts depending on the number of corresponding concepts in the training corpus. The CER results are given for the SAT model (#12), decoding with beam search and a 4-gram LM.

5 Conclusions

In this paper, we have investigated several ways to improve the performance of end-to-end SLU systems. We demonstrated the effectiveness of speaker adaptive training and various transfer learning approaches for two end-to-end SLU tasks: NER and SF. In order to improve the quality of the SF models, during the training, we proposed to use knowledge transfer from an ASR system in another language and from a NER in a target language. Experiments on the French MEDIA test corpus demonstrated that using knowledge transfer from the ASR in English improves the SF model performance by about 16% of relative CER reduction for SI models. The improvement from the transfer learning is greater when the ASR model is trained on the target language (36% of relative CER reduction) or when the NER model in the target language is used for pretraining. Another contribution concerns SAT training for SLU models – we demonstrated that this can significantly improve the model performance for NER and SF.

References

1. Amodei, et al.: Deep speech 2: End-to-end speech recognition in English and Mandarin. In: International conference on machine learning. pp. 173–182 (2016)
2. Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y.: End-to-end attention-based large vocabulary speech recognition. In: ICASSP. pp. 4945–4949. IEEE (2016)
3. Bechet, F., Maza, B., Bigouroux, N., Bazillon, T., El-Beze, M., et al.: DECODA: a call-centre human-human spoken conversation corpus. In: LREC. pp. 1343–1347 (2012)
4. Bonneau-Maynard, H., Ayache, C., Bechet, F., et al.: Results of the French Evalda-Media evaluation campaign for literal understanding. In: LREC (2006)
5. Chen, Y.P., Price, R., Bangalore, S.: Spoken language understanding without speech recognition. In: ICASSP (2018)
6. Deena, S., et al.: Semi-supervised adaptation of RNNLMs by fine-tuning with domain-specific auxiliary features. In: INTERSPEECH. pp. 2715–2719. ISCA (2017)

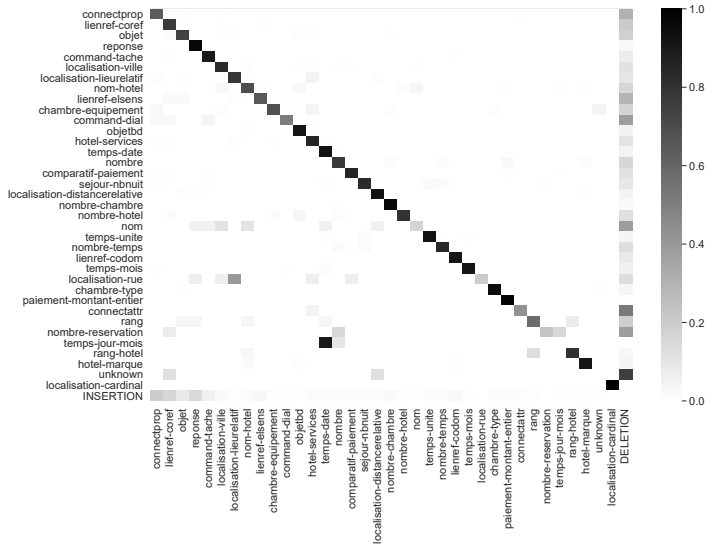

Fig. 4. Confusion matrix for concepts on the MEDIA test dataset. The last row and last column represent insertion and deletion errors correspondingly. The CER results are given for the SAT model (#12), decoding with beam search and a 4-gram LM.

7. Delcroix, M., Watanabe, S., Ogawa, A., Karita, S., Nakatani, T.: Auxiliary feature based adaptation of end-to-end asr systems. In: INTERSPEECH. pp. 2444–2448 (2018)
8. Devillers, L., et al.: The french MEDIA/EVALDA project: the evaluation of the understanding capability of spoken language dialogue systems. In: LREC (2004)
9. Estève, Y., Bazillon, T., Antoine, J.Y., Béchet, F., Farinas, J.: The EPAC corpus: Manual and automatic annotations of conversational speech in french broadcast news. In: LREC (2010)
10. Galliano, S., et al.: The ESTER 2 evaluation campaign for the rich transcription of french radio broadcasts. In: Interspeech (2009)
11. Gao, J., Galley, M., Li, L., et al.: Neural approaches to conversational AI. Foundations and Trends in Information Retrieval pp. 127–298 (2019)
12. Ghannay, S., Caubrière, A., Estève, Y., et al.: End-to-end named entity and semantic concept extraction from speech. In: SLT. pp. 692–699 (2018)
13. Giraudel, A., Carré, M., Mapelli, V., Kahn, J., Galibert, O., Quintard, L.: The REPERE corpus: a multimodal corpus for person recognition. In: LREC. pp. 1102–1107 (2012)
14. Graves, A., et al.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
15. Gravier, G., Adda, G., Paulson, N., et al.: The ETAPE corpus for the evaluation of speech-based TV content processing in the french language. In: LREC (2012)
16. Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O., Quintard, L.: Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 92–100 (2011)
17. Haghani, P., et al.: From audio to semantics: Approaches to end-to-end spoken language understanding. arXiv preprint arXiv:1809.09190 (2018)
18. Lefèvre, F., et al.: Robustness and portability of spoken language understanding systems among languages and domains: the PortMedia project [in French]. pp. 779–786 (2012)

19. Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V.S., Bengio, Y.: Speech model pre-training for end-to-end spoken language understanding. arXiv preprint arXiv:1904.03670 (2019)
20. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering pp. 1345–1359 (2010)
21. Povey, D., Ghoshal, A., et al.: The Kaldi speech recognition toolkit. In: ASRU (2011)
22. Qian, Y., Ubale, R., et al.: Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system. In: ASRU. pp. 569–576 (2017)
23. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Natural language processing using very large corpora, pp. 157–176. Springer (1999)
24. Rousseau, A., Deléglise, P., Esteve, Y.: Enhancing the TED-LIUM corpus with selected data for language modeling and more ted talks. In: LREC. pp. 3935–3939 (2014)
25. Saon, G., Soltau, H., Nahamoo, D., Picheny, M.: Speaker adaptation of neural network acoustic models using i-vectors. In: ASRU. pp. 55–59 (2013)
26. Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B., Bengio, Y.: Towards end-to-end spoken language understanding. arXiv preprint arXiv:1802.08395 (2018)
27. Simonnet, E., et al.: Simulating asr errors for training SLU systems. In: LREC 2018 (2018)
28. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
29. Vukotic, V., Raymond, C., Gravier, G.: Is it time to switch to word embedding and recurrent neural networks for spoken language understanding? In: Interspeech (2015)