Transfer Learning of Transformers for Spoken Language Understanding

Jan Švec(✉), Adam Frémund, Martin Bulín, and Jan Lehečka

Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic
{honzas,afremund,bulinm,jlehecka}@kky.zcu.cz

Abstract. Pre-trained models used in the transfer-learning scenario are recently becoming very popular. Such models benefit from the availability of large sets of unlabeled data. Two kinds of such models include the Wav2Vec 2.0 speech recognizer and T5 text-to-text transformer. In this paper, we describe a novel application of such models for dialog systems, where both the speech recognizer and the spoken language understanding modules are represented as Transformer models. Such composition outperforms the baseline based on the DNN-HMM speech recognizer and CNN understanding.

Keywords: Wav2Vec model · Speech recognition · T5 model · Spoken language understanding

1 Introduction

Speech processing techniques historically relied on the noisy channel model and the generative paradigm. In automatic speech recognition (ASR), this class of models was represented by the GMM-HMM type of speech recognizers [3]. The same generative paradigm was followed also in the downstream tasks of speech processing in spoken dialog systems, especially in the spoken language understanding module (SLU) [5]. The rise of deep neural networks together with the massive popularity of automatic differentiation toolkits such as TensorFlow or PyTorch introduced the discriminative model paradigm first into the field of SLU [6]. The advent of deep neural networks emerged as a hybrid DNN-HMM architecture, where only the Gaussian mixtures were replaced by the deep neural networks keeping the rest of the decoder stack the same (pronunciation lexicon, language model, and decoding strategy).

The research was accelerated with the publication of the Transformer architecture [12], which was introduced first for the high-level linguistic tasks such as machine translation. The Transformer architecture is capable to capture the inherent knowledge present in large datasets. But for many tasks, including ASR

This research was supported by the Czech Science Foundation (GA CR), project No. GA22-27800S and by the grant of the University of West Bohemia, project No. SGS-2022–017.

and SLU, the number of available labeled data is always insufficient for training the powerful parameter-rich Transformer model. The problem was solved by using a transfer learning approach, that uses models trained on large unlabelled data in a self-supervised manner as a base for subsequent training of the parameters on small labeled data. The representatives of this class of models are BERT (Bidirectional Encoder Representations from Transformers) [4] or T5 (Text-To-Text Transfer Transformer) [10] models. The use of the Transformer architecture has also spread to the field of ASR, changing the modeling paradigm from generative models to discriminative models. The end-to-end training of ASR has started to become the state-of-the-art [9]. The recently published Wav2Vec 2.0 combines the Transformer architecture with the transfer learning approach and with the direct processing of a speech signal using convolutional layers [2].

In this paper, we present the results of using the transfer learning approach both in the ASR and the SLU modules of the Czech spoken dialog system. We use the Wav2Vec 2.0 speech recognizer instead of the traditional DNN-HMM hybrid ASR and the fine-tuned T5 model as a replacement for the discriminative SLU module based on carefully designed convolutional architecture.

2 Transfer Learning for Spoken Dialog Systems

The Transformer architecture is a building block of many modern approaches in speech and language processing. The rise of transfer-learning techniques leads to novel methods with state-of-the-art performance. The training process consists of two steps: (1) pre-training a generic model and (2) fine-tuning the pre-trained model on in-domain data. We used these paradigms for both models involved in spoken language understanding. We choose the Wav2Vec 2.0 model as a speech recognizer and the Text-To-Text Transfer Transformer (T5) as an SLU module. In this section, the pre-training procedure of the generic models is described. The fine-tuning step of speech recognizer is then described in Sect. 3.2 and fine-tuning of the SLU in Sect. 4.2.

2.1 Wav2Vec 2.0 Transformer

One of the most studied self-supervised end-to-end automatic speech recognition (ASR) model architectures is Wav2Vec 2.0 [2]. It is a deep neural network pre-trained to reconstruct the corrupted signals. The input raw audio signal is processed by a multi-layer convolutional neural network into a sequence of latent-speech representations which are fed into a multi-layer Transformer [12] (Fig. 1). The output of the Transformer is a sequence of frame-level contextualized speech representations which are then processed by the connectionist temporal classification (CTC) layer [1] decoding the most probable sequence of graphemes.

Because there is no Wav2Vec 2.0 model available for Czech language, which we are experimenting with, we decided to pre-train our own model. We gathered as much public and in-house unlabeled audio data as possible. Together, we collected more than 80 thousand hours of Czech speech. The collection includes recordings from radio (22k hours), unlabeled data from VoxPopuli dataset [13] (18.7k hours), TV shows (15k hours), shadow speakers (12k hours), sports (5k hours), telephone data (2k hours), and a smaller amount of data from several other domains.

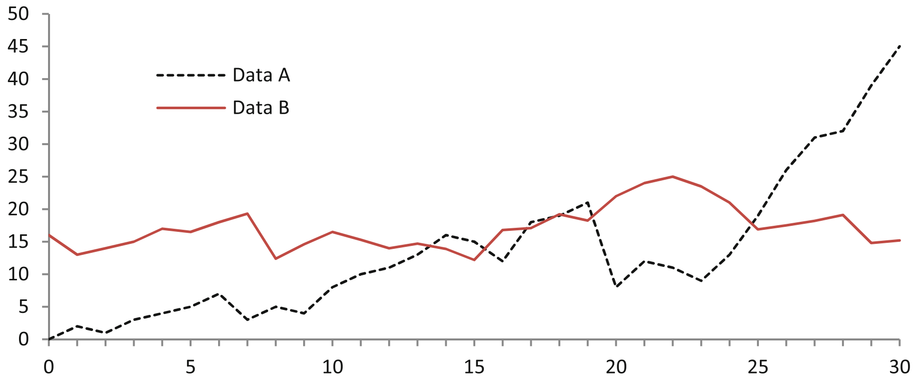

Fig. 1. Architecture of the Wav2Vec 2.0 model. Figure taken from [2].

Since the feature extraction of the input signal is limited by the memory of GPUs in use, we sliced all records not to exceed 30 s, which we found to be a reasonable input size for batching.

We followed the same pre-training steps as for the base Wav2Vec 2.0 model in [2]. We pre-trained the model for 400 thousand steps with a batch size of about 1.6 h, corresponding to more than 11 epochs over the dataset. We released our pre-trained model under the nickname *ClTRUS* (abbreviation for **C**zech **l**anguage **TR**ransformer from **U**nlabeled **S**peech) for public non-commercial use[1].

2.2 Text-to-Text Transfer Transformer

The Text-to-Text Transfer Transformer (T5) model is a self-supervised trained variant of the generic textual Transformer architecture [10]. The T5 model is able to construct the internal representation of input on many linguistic layers: starting from phonetic and syntactic through semantic to the pragmatic layer. The T5 model is pre-trained in a self-supervised manner by generating a text restoration task from unlabelled training data. An example of the pre-training input/output text pair is shown in Fig. 2.

The T5 model tries to recover missing tokens in the input sentence masked with sentinel tokens `<X>` and `<Y>`. The masked tokens are used as training targets and the output sentence is terminated using another sentinel token `<Z>`.

[1] Available at https://huggingface.co/fav-kky/wav2vec2-base-cs-80k-ClTRUS.

This way, the T5 learns not only the knowledge required to understand the input sentence but also the knowledge necessary to generate meaningful output sentences.

The original Google's T5-base English model[2] was trained from Common Crawl data[3]. We replicated the same pre-processing procedure to obtain the Czech data and we pre-trained our own T5 model for Czech language. The pre-processing steps correspond with the steps presented in [11] for building the Colossal Clean Crawled Corpus (C4) on which Google's T5 model was pre-trained. Such rules are generally applied while processing web text:

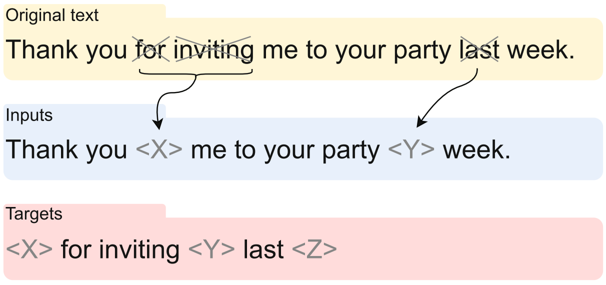
Original text
Thank you ~~for inviting~~ me to your party ~~last~~ week.
Inputs
Thank you <X> me to your party <Y> week.
Targets
<X> for inviting <Y> last <Z>

Fig. 2. Example of processing the original text and creating input/output text pairs. Figure taken from [10].

- Only lines ending in terminal punctuation are retained. Short pages and lines are discarded.
- Pages with dirty and obscene words are removed.
- Lines with the word "JavaScript" and the curly braces { } are removed (remains of incorrect crawling of the webpage).
- The pages in the corpus were de-duplicated. The resulting corpus contains each three-sentence span just once.

For the Czech language, we have collected the CommonCrawl corpus at the end of August 2020. It contains 47.9 GB of clean text, 20.5 M unique URLs and 6.7 B running words. The Czech T5 training procedure followed the original procedure described in [11].

We used the `t5-base` architecture consisting of 220M parameters, 2×12 transformer block in the encoder and the decoder. The dimensionality of hidden layers and embeddings was 768. The attention mechanism uses 12 attention heads with inner dimensionality 64.

[2] https://github.com/google-research/text-to-text-transfer-transformer.
[3] https://commoncrawl.org/.

3 Speech Recognition

Speech recognition is the first module in a dialog system that transforms an input audio signal into a textual representation usable in the SLU module. The accuracy of speech recognition is the key metric of the dialog system because the entities which are not recognizable in this module cannot be used in downstream dialog processing. In this paper, we compare the performance of the DNN-HMM baseline speech recognizer and the Transformer-based Wav2Vec 2.0 (W2V) recognizer.

3.1 DNN-HMM Baseline

The traditional DNN-HMM baseline speech recognizer consists of a set of different models, esp. an acoustic model, pronunciation lexicon, and language model. We used two separated class-based languages models for the HHTT and TIA corpora trained from the training set with classes populated by the related entities (names, station names, etc.). The acoustic model was a generic hybrid DNN-HMM model with PLP parameterization. The pronunciation lexicon was generated by the Czech phonetic rules. The DNN-HMM system allows generating of not only the 1-best word hypothesis but also the word-lattice of multiple hypotheses. This allows the SLU module to compensate for the lower recognition accuracy if it is designed to process such input structure.

3.2 Wav2Vec 2.0 Recognizer

The W2V recognizer consists of a stack of convolutional layers transforming the raw input waveform into a latent speech representation. The latent speech representation is then processed in the Transformer model to obtain a contextual representation of the speech. This representation is then classified using a single dense multi-class classification layer with softmax activation and trained using the CTC loss. In this paper, we used an unsupervised scenario where the recognizer was not trained on the in-domain data. The fine-tuning data consisted of 5k hours of labeled data including approximately 430 h of telephone speech with no overlap with the in-domain data.

The W2V recognizer is a grapheme-based recognizer containing no recognition lexicon nor language model. The W2V recognizer generates an orthographic transcription of the input audio. It is able to generate grapheme posterior probabilities, but the conversion of the output into the form of word-lattice is complicated. We, therefore, use only the 1-best hypothesis.

When analyzing the output of the W2V recognizer, the confusion tables revealed that a significant number of errors were caused by wrong word forms, while the meaning was fully kept. It was caused mainly by the use of Czech colloquial word forms. Therefore, we decided to apply an output normalization method mapping all possible word variants having the same semantic meaning into one correct form.

4 Spoken Language Understanding

The spoken language understanding (SLU) module of a dialog system converts the output of the speech recognizer into a semantic representation, which encodes the meaning of the utterance. In this work, we used the abstract semantic trees. Such trees consist of nodes labeled with semantic concepts. The trees are abstract because they do not encode an alignment between the words of the recognized hypothesis and the nodes of the tree. The abstract semantic trees could be represented in the textual parenthesized form (for examples, see Table 1).

4.1 CNN SLU Baseline

The baseline SLU method used in this paper is a hierarchical discriminative model based on a convolutional neural network (CNN SLU) [15]. This model is able to cope with the multiple hypotheses output from the speech recognizer in the form of a word lattice. The convolution is performed directly on the word lattice, where each convolution region is represented as a bag-of-word vector with the corresponding lattice score assigned to this region. The output of the convolutional layer is max-pooled to obtain a fixed-size vector representation of the input word lattice. Then, this vector representation is transformed using a stack of fully connected layers to obtain expansion probabilities of the semantic grammar.

The semantic grammar is parameterized by the utterance u and consists of a tuple $G_u = (\Theta, R_u, S)$, where Θ is a set of semantic concepts, R_u is a set of grammar rules dependent on the utterance u and $S \in \Theta$ is a root concept (starting symbol of the parsing algorithm). The rules R_u are in the form $A \rightarrow \beta\ [p]$, where $A \in \Theta$, $\beta \subseteq \{\nu\} \cup \Theta$, ν is a special symbol representing rules without lexical realization in the utterance u (see below) and p is the probability of concept A having a set of child nodes β:

$$p = P(A \rightarrow \beta|u) = P(\beta|A, u) \tag{1}$$

A standard best-first search algorithm is used to decode the most probable unordered semantic tree assigned to the utterance u by iteratively expanding the nodes of the semantic tree starting with the root symbol S_0 and using the rules from R_u. In the CNN reimplementation of the HDM, the output layer contains multiple softmax units (one unit for each concept A prediction posterior probability distribution $P(\beta|A, u)$). The parameters of the neural network are then optimized using the categorical cross-entropy loss function.

4.2 T5 SLU

For fine-tuning the T5-based SLU model we used the Tensorflow implementation of HuggingFace Transformers library [14] together with the `t5s`[4] library. This

[4] https://github.com/honzas83/t5s.

library simplifies the process of fine-tuning, predicting, and evaluating the model to easy-to-use Python modules. The input files (train, eval, test data sets) for fine-tuning are the tab-separated files (TSV) containing two columns: *input text* and *output text*. For training, it is also necessary to prepare a simple YAML configuration file, which specifies the name of the pre-trained T5 model, the SentencePiece model, the names of TSV files for fine-tuning and hyperparameters settings such as the number of epochs, batch sizes, and learning rate. The `t5s` library uses a variable sequence length and a variable batch size to fully utilize the underlying GPU used for training. In the experiments, we used the ADAM optimization with learning rate decay proportional to the inverse square root of the number of learning epochs.

As the T5 model allows to transfer the task of spoken language understanding into the text-to-text transformation, we used for fine-tuning pairs consisting of speech recognition outputs and relevant parenthesized abstract semantic trees. During the fine-tuning procedure, we tried many combinations of hyperparameter values (number of epochs, steps per epoch, and learning rate). We also experimented with training from ground-truth transcriptions only, recognized data only, and from the mixture of ground-truth and recognized data. The best combination of such hyperparameters was determined using the development set and subsequently, the test data were processed using these settings. The best results were obtained using the following values: the number of epochs: 2, steps per epoch: 1800, learning rate: $5 \cdot 10^{-4}$ and training from the mixture of ground-truth and recognized data.

Table 1. Examples of training pairs consisting of speech recognition output (input text) and relevant parenthesized abstract semantic trees (expected output). The first three examples come from the TIA corpus and the other three examples are from the HHTT corpus.

Utterance transcription	Abstract semantic tree
dobrý den mám čas pozítří od jedný do půl druhý	HELLO, ZJISTI(KALENDAR(TIME))
Lit. *hello I have time tomorrow from one to half past one*	
nejlepší by to bylo zítra od půl druhé do čtyř hodin	VYTVOR(SCHUZKY(TIME))
Lit. *it would be best tomorrow from half past one to four o'clock*	
je volno v zasedačce dnes od čtyř do sedmi	ZJISTI(KALENDAR(SUBJECT, TIME, VEC))
Lit. *is the meeting room available today from four to seven?*	
v kolik jedou vlaky na prahu kolem páté a šesté hodiny	DEPARTURE(TO(STATION), TIME)
Lit. *what time do the trains go to Prague around five or six o'clock?*	
no tak potom v šest sedmnáct s přestupem v chomutově	TIME, TRANSFER(STATION)
Lit. *well, then at six-seventeen with a transfer in Chomutov*	
ano dvacet jedna deset staví v rokycanech	ACCEPT(ARRIVAL(TIME, TO(STATION)))
Lit. *yes (the train) at ten past nine stops in Rokycany*	

5 Dataset Description

In the experiments, we use two Czech semantically annotated corpora: a Human-Human Train Timetable (HHTT) corpus [8] which contains inquiries and answers about train connections; and an Intelligent Telephone Assistant (TIA) [16] corpus containing utterances about meeting planning, corporate resources sharing and conference call management. These corpora contain unaligned semantic trees together with word-level transcriptions (for examples, see Table 1). We have split the corpora into train, development, and test data sets (72:8:20) at the dialog level so that the speakers do not overlap (Table 2). To perform the multi-task training experiment, we had to unify some semantic concepts. In the HHTT corpus, all time and date information was annotated as Time. On the opposite side the TIA corpus contained more granular annotation of dates and times include concepts like Time, Interval, Relative-Date etc. To avoid re-annotation of the first corpus, we have merged all time- and date-related concepts into a Time concept. We have also unified the concepts for agreement and disagreement so that the resulting corpus contains Accept and Reject concepts.

To evaluate the SLU performance we use the *concept accuracy* measure [17] defined as $cAcc = \frac{N-S-D-I}{N} = \frac{H-I}{N}$ where H is the number of correctly recognized concepts, N is the number of concepts in reference and S, D, I are the numbers of substituted, deleted and inserted concepts. The concept accuracy can measure the similarity of partially matching semantic trees. We also evaluate the *sentence accuracy* measure ($sAcc$), which measures the ratio of sentences with the predicted semantic tree exactly matching the reference semantic tree.

6 Experimental Evaluation

The comparison of the recognition word accuracy of the speech recognizers is presented in Table 3. First, we present the performance of two different DNN-HMM recognizers on the HHTT and TIA datasets. In this case, the recognizer

Table 2. Corpora characteristics.

	HHTT	TIA	multi-task
# different concepts	28	19	46
# different semantic trees (train)	380	253	630
# train sentences	5240	6425	11665
# train concepts	8967	13499	22466
# dev. sentences	570	519	1089
# dev. concepts	989	1106	2095
# test sentences	1439	1256	2695
# test concepts	2546	2833	5379

is tailored for the specific task by using a domain-dependent language model. Then, we recognized the same data using the W2V recognizer. Although the Wav2Vec recognizer does not use domain knowledge, we report the recognition accuracy for TIA and HHTT datasets separately. The comparison of W2V with the DNN-HMM shows that the W2V provides a significant performance boost on the TIA dataset but no improvement on the HHTT dataset. The error analysis on the HHTT dataset showed that a large number of errors come from the orthographic transcription produced by the W2V recognizer scored against a normalized ground-truth reference. Therefore, we applied the rule-based normalization mentioned in Sect. 3.2. We defined a set of 91 normalization rules in total, for example:

- na shledanou ← nashledanou, naschledanou, na schledanou (**Lit.** *good bye*)
- tři čtvrtě ← tvičtvrtě, tvištvrtě, tvi štvrtě (**Lit.** *three quarters*)
- děkuji ← děkuju (**Lit.** *thanks*)

Using these rules, we normalized the recognizer output as well as the ground truth transcription. The vocabulary size was reduced by 31 words in the case of the HHTT corpus and by 47 words for the TIA corpus respectively. The normalized outputs from the W2V recognizer have a significantly higher recognition accuracy than the DNN-HMM baseline.

Since we train a single SLU model using multi-task conditions, we also report the recognition accuracy on the union of HHTT and TIA datasets. Under the multi-task condition, the recognition accuracy on test data increased from 76.23% (DNN-HMM baseline) to 85.01% (normalized W2V recognizer). The lower accuracy on development data is caused by the selection of utterances which are probably more challenging to recognize.

In the next set of experiments, we compared the CNN SLU baseline with the T5 SLU model. We have to note, that the CNN SLU baseline is a special model designed for the SLU task and is able to process the input in the form

Table 3. Speech recognition word-level accuracy.

	% Acc	
	devel	test
DNN-HMM TIA	71.31	77.88
DNN-HMM HHTT	70.40	74.05
W2V TIA	83.70	86.08
W2V HHTT	68.93	73.66
W2V TIA normalized	86.39	89.14
W2V HHTT normalized	73.80	79.48
DNN-HMM TIA+HHTT	70.92	76.23
W2V TIA+HHTT	77.30	80.72
W2V TIA+HHTT normalized	80.96	85.01

Table 4. Spoken language understanding performance.

	% cAcc		% sAcc	
	devel	test	devel	test
DNN-HMM ASR + CNN SLU (baseline)	76.04	80.24	69.70	74.51
DNN-HMM ASR + T5 SLU	76.09	81.50	70.98	74.84
W2V ASR + T5 SLU	80.81	84.29	73.09	79.04
W2V ASR normalized + T5 SLU	81.19	85.37	73.55	79.33
Ground truth transcription + T5 SLU	87.54	87.69	81.27	83.41

of word lattice and also generate the probabilistic distribution over the set of semantic trees. In addition, we used ensembling of multiple models to filter-out different random initializations [16]. By contrast, the T5 SLU model is a text-to-text transformer working only with the 1-best input and 1-best output. From this point of view, the results shown in Table 4 are very promising – the much simpler fine-tuning and prediction of the T5 model is fully compensated by the knowledge extracted during self-supervised pre-training. The T5 SLU model provides better performance on the test data in both the *cAcc* and *sAcc* metrics when using the DNN-HMM speech recognizer in comparison with the baseline CNN SLU.

Since the T5 model is a generative model, the correctness of the generated parenthesized semantic trees is not guaranteed. The T5 generates its output in an autoregressive manner and therefore a small number of predicted semantic trees have an incorrect number of closing parentheses. We, therefore, perform a simple post-processing in which we add the missing parentheses to obtain valid semantic trees. The T5 model also does not have a fixed set of output semantic concepts and therefore it is able to generate semantic concepts not seen in the training data. We treat such concepts as erroneous predictions and count them as an insertion or substitution errors.

By moving from the DNN-HMM speech recognizer to the W2V recognizer, the performance of the T5 SLU significantly improves in both metrics (Table 4). The impact of W2V recognition output normalization is less obvious than in the speech recognition experiment. This is expected because the normalization rules only merge the semantically equivalent words.

The last row of Table 4 shows the ceiling for the *cAcc* and *sAcc* metrics obtained by using the ground-truth transcriptions instead of the speech recognizer. There is only a 2.32% margin in *cAcc* which indicates that the W2V speech recognizer graphemic hypotheses are semantically close to the human-made transcriptions. At the same time, the speech recognition accuracy is around 85% and we can state that the errors are not changing the semantics of the utterances.

7 Conclusion

In this paper, we presented the application of Transformer-based models in the spoken dialog systems. We combined the speech recognizer represented by the Wav2Vec 2.0 recognizer with CTC output and spoken language understanding implemented as the fine-tuned T5 model. The overall performance of the ASR/SLU pipeline outperforms the baseline method based on the traditional approach of DNN-HMM hybrid ASR and SLU based on convolutional networks.

The results presented in this paper are very promising and outline future research in the applications of transfer-learning Transformers in spoken dialog systems. First of all, the Wav2Vec 2.0 speech recognizer with the current architecture is not suitable for real-time usage required by natural speech interaction. The causes are mainly the full attention mechanisms used in the Transformer as well as the computational costs of the Wav2Vec model. Another open question is the composability of the Transformer models. In the current setup, the utterances are processed step-wise (speech $\rightarrow$ text $\rightarrow$ semantics). Future research should focus on the composition of multiple transformers (e.g. by using adapters [7]) and the joint end-to-end training. This way, the limitation on only the 1-best text hypothesis would be eliminated. Also, the text normalization process could be implicitly modeled in the composed model.

Acknowledgement. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Baevski, A., Rahman Mohamed, A.: Effectiveness of self-supervised pre-training for ASR. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7694–7698 (2020)
2. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: Wav2Vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. **33**, 12449–12460 (2020)
3. Deng, L., Li, X.: Machine learning paradigms for speech recognition: an overview. IEEE Trans. Audio Speech Lang. Process. **21**(5), 1060–1089 (2013). https://doi.org/10.1109/TASL.2013.2244083
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the NAACL: HLT, Vol. 1, pp. 4171–4186. ACL, Minneapolis, Minnesota (2019)
5. He, Y., Young, S.: Spoken language understanding using the hidden vector state model. Speech Commun. **48**(3), 262–275 (2006). https://doi.org/10.1016/j.specom.2005.06.002. www.sciencedirect.com/science/article/pii/S0167639305001421. Spoken Language Understanding in Conversational Systems
6. Henderson, M., Gašić, M., Thomson, B., Tsiakoulis, P., Yu, K., Young, S.: Discriminative spoken language understanding using word confusion networks. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 176–181 (2012). https://doi.org/10.1109/SLT.2012.6424218

7. Houlsby, N., et al.: Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning, pp. 2790–2799. PMLR (2019)
8. Jurčíček, F., Zahradil, J., Jelínek, L.: A human-human train timetable dialogue corpus. In: Proceedings of EUROSPEECH, Lisboa, pp. 1525–1528 (2005). ftp://ftp.cs.pitt.edu/web/projects/nlp/conf/interspeech2005/IS2005/PDF/AUTHOR/IS051430.PDF
9. Karita, S., Soplin, N.E.Y., Watanabe, S., Delcroix, M., Ogawa, A., Nakatani, T.: Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In: Proceedings Interspeech 2019, pp. 1408–1412 (2019). https://doi.org/10.21437/Interspeech.2019-1938
10. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
11. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer (2020). arXiv preprint arXiv:1910.10683
12. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Wang, C., et al.: VoxPopuli: a large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 993–1003. Association for Computational Linguistics (2021). https://aclanthology.org/2021.acl-long.80
14. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics (2020)
15. Švec, J., Chýlek, A., Šmídl, L.: Hierarchical discriminative model for spoken language understanding based on convolutional neural network. In: Proceedings Interspeech 2015, pp. 1864–1868 (2015). https://doi.org/10.21437/Interspeech.2015-72
16. Švec, J., Chýlek, A., Šmídl, L., Ircing, P.: A study of different weighting schemes for spoken language understanding based on convolutional neural networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6065–6069 (2016). https://doi.org/10.1109/ICASSP.2016.7472842
17. Švec, J., Šmídl, L., Ircing, P.: Hierarchical discriminative model for spoken language understanding. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8322–8326 (2013). https://doi.org/10.1109/ICASSP.2013.6639288