INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Towards End-to-End Spoken Dialogue Systems with Turn Embeddings

Ali Orkan Bayer, Evgeny A. Stepanov, Giuseppe Riccardi

Signals and Interactive Systems Lab - University of Trento, Italy
{aliorkan.bayer, evgeny.stepanov, giuseppe.riccardi}@unitn.it

Abstract

Training task-oriented dialogue systems requires significant amount of manual effort and integration of many independently built components; moreover, the pipeline is prone to error-propagation. End-to-end training has been proposed to overcome these problems by training the whole system over the utterances of both dialogue parties. In this paper we present an end-to-end spoken dialogue system architecture that is based on turn embeddings. Turn embeddings encode a robust representation of user turns with a local dialogue history and they are trained using sequence-to-sequence models. Turn embeddings are trained by generating the previous and the next turns of the dialogue and additionally perform spoken language understanding. The end-to-end spoken dialogue system is trained using the pre-trained turn embeddings in a stateful architecture that considers the whole dialogue history. We observe that the proposed spoken dialogue system architecture outperforms the models based on local-only dialogue history and it is robust to automatic speech recognition errors.

Index Terms: spoken dialogue systems, end-to-end learning, sequence-to-sequence models

1. Introduction

Conventionally, spoken dialogue systems (SDSs) are built by integrating several independent components, as shown in Figure 1. Traditional SDS components are the following: automatic speech recognition (ASR), spoken language understanding (SLU), dialogue manager (DM), natural language generation (NLG), and text-to-speech (TTS). Combination of independently trained models may lead to problematic error propagation. In addition, design and implementation of these systems may require significant amount of manual effort.

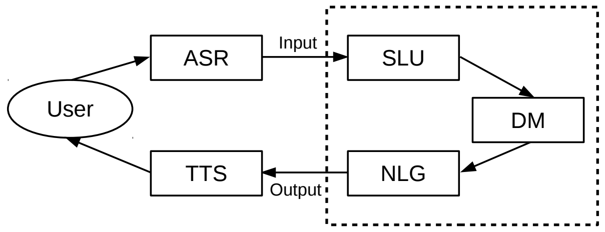

Figure 1: *Conventional spoken dialogue systems architecture. The dotted rectangle shows the components that are covered by the end-to-end model that is presented.*

End-to-end learning overcomes these problems by training SDSs using the data obtained from the available dialogues. End-to-end learning has been applied to dialogue systems such as short response generation [1], open domain conversational dialogue systems [2], task-oriented dialogue systems [3], and the joint training of understanding models with a dialogue manager to predict system actions [4].

The definition of the input and output in the end-to-end learning for spoken dialog system varies in the literature. In [4], for instance, the outputs of the end-to-end system are system actions that are learned from user turns. However, generally we can assume that two ends are user utterance and system response. Thus, in this paper, we consider one end to be ASR output and the other end to be the output of the NLG module; as shown by the dashed box in Figure 1. Therefore, the input to the proposed system is the ASR transcription of the user turn and the output is the generated next system turn.

End-to-end learning for dialogue systems uses sequence-to-sequence neural networks. Sequence-to-sequence models were introduced for the task of machine translation in [5, 6]. They consist of an encoder network which encodes the sentence in the source language and a decoder which generates the corresponding translation in the target language. End-to-end dialogue systems use a similar architecture [1, 2, 3] to predict the next system turn from the current user turn. [1] and [2] present a response generation system and an open domain conversational system respectively that are both text-based. [3] proposes a task-oriented SDS which lacks an SLU model; the system takes slot-value pairs as input.

In this paper, we propose a task-oriented spoken dialogue system architecture[1] that is based on turn embeddings – a robust representation of user turns. We utilize sequence-to-sequence neural networks to learn turn embeddings of user turns by predicting previous and next system turns. Also we train an SLU model jointly with the sequence-to-sequence model for capturing the semantic information in the turn embeddings. Finally, we build a dialogue model that uses the pre-trained turn embeddings of the user turns that are obtained by the sequence-to-sequence model as input, and predicts the next system turns. The dialogue model we propose uses the whole history of the dialogue by using a recurrent neural network (RNN) architecture.

2. Turn embeddings

In this section we define the neural network architecture for learning turn embeddings and describe how these networks are trained. Word embeddings have been used to map discrete word representations onto a continuous space. It has been shown that word embeddings improve the accuracy of natural language processing tasks by representing semantically similar words with similar vectors. One of the widely used approach to train word embeddings is the skip-gram model that is given in [7]. The skip-gram model learns word embeddings by predicting the preceding and succeeding words.

The aim of learning turn embeddings is to encode user turns with a distributed representation and to map similar user turns to similar vectors. Therefore, we use a similar strategy to the skip-gram model, i.e., we train turn embeddings by predicting the preceding and succeeding system turns. Each turn consists

[1] A complete functional SDS would require connecting the back-end.

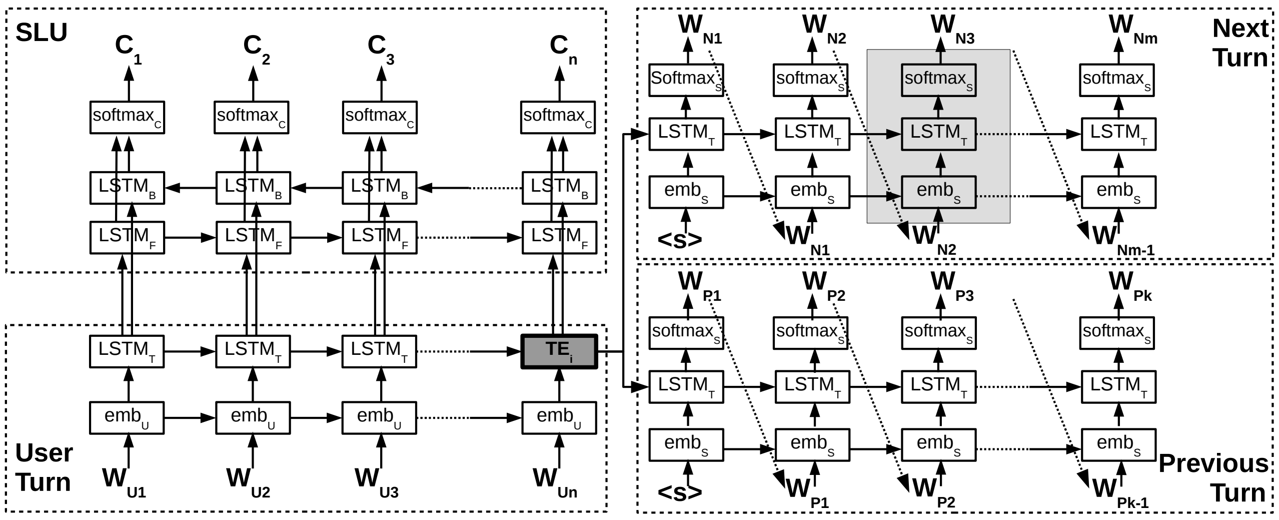

Figure 2: *The sequence-to-sequence model architecture for learning turn embeddings. The sequence-to-sequence model is trained by predicting the previous and the next system turns jointly with the semantic concepts; given the user turn. The LSTM cells of the encoder that processes the user turn and of all the decoders that predicts the system turns are shared. The decoders share the same embedding layer and the softmax layer; but a different embedding layer is used for the encoder. The turn embedding for a user turn, "TE_i", is obtained at the shaded bold LSTM cell by using its hidden state. The shaded rectangle denotes the building block of a decoder unit. The dotted arrows indicate that during decoding the output of the decoder unit is fed as input to the next decoder unit, however training is performed with reference transcriptions.*

of a sequence of words therefore, we use sequence-to-sequence models as the building block.

2.1. Sequence-to-sequence models

The basic sequence-to-sequence model that is described in [6] is used for the machine translation task. The model has an encoder-decoder architecture, where the source language utterance is encoded with an RNN encoder. The decoder uses the final state of the encoder as the initial state and outputs a word sequence in the target language. The authors use long-short term memory (LSTM) [8] cells to handle long-range dependencies better.

We have extended this architecture by using multiple decoders to predict not only the next system turn but other next and even previous system turns. The encoder and all of the decoders share the same LSTM cells that are defined by the following equations:

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + b_o) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \tag{4}$$

$$m_t = o_t \odot tanh(c_t) \tag{5}$$

The parameters of the LSTM cells are the weight matrices: W_{fx}, W_{fm}, W_{ix}, W_{im}, W_{ox}, W_{om}, W_{cx}, W_{cm}; and the bias vectors: b_f, b_i, b_o, b_c. x_t represents the input at time t. The forget gate, input gate, and output gate activations at time t is given by f_t, i_t, and o_t. c_t defines the cell activation vectors and m_t defines the cell output activation vectors at time t. σ is the sigmoid function and $\odot$ represents element-wise multiplication operation.

In addition to LSTM cells, the sequence-to-sequence model uses embedding layers to map one-of-n encoding of words onto a continuous space. All of the decoders share the same embedding layer, however for the encoder we use a different embedding layer since the vocabularies of user and system turns differ. The decoders also share a softmax layer to output word predictions for the system turn. The sequence-to-sequence model that we use for training turn embeddings is depicted in Figure 2. This model predicts the previous and the next system turns jointly with the semantic concepts given the user turn.

2.2. SLU model

SLU is one of crucial components of task-oriented spoken dialogue systems and maps the user utterance onto a task specific meaning representation that is determined by the application domain. SLU is usually modeled as a sequence labeling task in which each word is assigned a semantic concept label. The current state-of-the-art SLU systems use RNNs [9]. In [10] the RNN that uses bi-directional LSTM cells is shown to outperform other architectures on multi-domain SLU.

In this paper we use bi-directional LSTM cells for the SLU model. The SLU model uses two different LSTM cells; one for the forward and one for the backward direction that are defined by Equations 1-5. As shown in Figure 2 the bi-directional LSTM cells take the cell output activations of the encoder cells as input, and predict the semantic concept for the corresponding word. We train the SLU model jointly with the decoders that predict the system turns.

2.3. Training turn embeddings

Turn embeddings are trained jointly with all of the decoders that predict the system turns and the SLU model. During training the whole network is unrolled and the current user turn is fed into the encoder. The semantic concepts are predicted by the bi-directional LSTM cells. The system turns are generated one by one by the decoder networks by feeding the reference transcription of the machine turns into the decoder inputs. The parameters of the network is optimized by AdaGrad [11] algorithm with the cross entropy cost function. On a trained net-

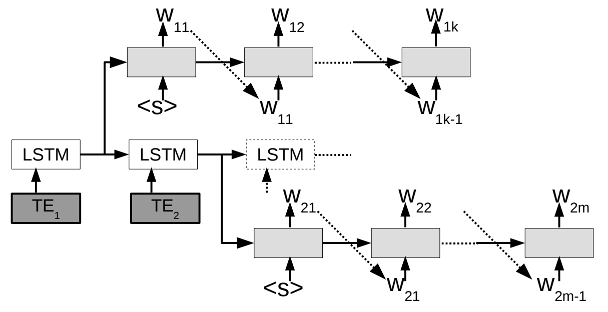

Figure 3: *The dialogue model architecture. The system takes the pre-trained turn embeddings, "TE_i", for all the user turns in a single dialogue as input and generates the corresponding system turns at each step by the same decoder unit that are used in sequence-to-sequence models. The recurrent architecture enables the system to generate systems turns based on the whole dialogue history.*

work, the turn embeddings are obtained at the final LSTM cell of the encoder network that is shaded in Figure 2. The c_t and m_t vectors that are given by Equation 4 and 5 are concatenated to obtain the turn embedding for the given user turn. In addition, the semantic concept labels for a given turn are obtained from the SLU model.

3. Dialogue model

The pre-trained turn embeddings are used to train a complete dialogue model which considers the whole dialogue history. For that purpose the dialogue model uses an RNN network with LSTM cells, where each cell takes the turn embedding for the user turn as input at that time step. Hence, each LSTM cell represents the dialogue state at that turn. The system turn corresponding to the user turn at a specific time is generated by using a decoder network that is initialized by the LSTM cell state at that time step. The decoders have the same architecture as the decoders in the sequence-to-sequence models. The LSTM cells in the decoders are shared by all of the decoders and by the dialogue model. The dialogue model that is defined is depicted in Figure 3.

The training of the dialogue model is performed as follows. The pre-trained turn embeddings for every user turn are obtained from the last LSTM cell of the encoder in the architecture given in Figure 2. The whole dialogue network is unrolled and the pre-trained turn embeddings are fed into the network. The corresponding system turns are generated by feeding the reference transcriptions of the system turns as inputs to the corresponding decoders. The predictions for each system turn is computed and the parameters of the whole network is optimized by using AdaGrad [11] algorithm with the cross entropy cost function. At each iteration the cost is calculated over the development set and the learning rate is adjusted and early stopping is applied to prevent overfitting.

4. Experimental work

The performance of the turn embeddings and the dialogue model is evaluated on the human-machine part of the Italian LUNA Corpus (LUNA HM) [12]. The dialogues are spoken conversations in the hardware/software help desk domain that were collected using Wizard of Oz (WOZ) technique: the human agent (wizard) reacting to user requests is following one of the ten most common scenarios identified by the help desk service provider. Responses to users were provided using Text-to-Speech Synthesis (TTS). LUNA HM corpus is split into training, development, and test partitions which contain 542, 71, and 110 dialogues respectively.

The performance of the turn embeddings are evaluated both on the understanding performance using concept error rate (CER) and on the next turn generation performance using BLEU [13] score. The performance of the dialogue model, on the other hand, is evaluated only by BLEU. The models are evaluated both on the reference transcription and on the ASR output of the test set.

4.1. ASR system

The ASR baseline for LUNA HM corpus is built using the Kaldi [14] speech recognition toolkit. The hidden Markov model (HMM) acoustic models are trained using mel-frequency cepstral coefficients that are transformed by linear discriminant analysis and maximum likelihood linear transform. These features are then spliced in the window of $[-3, +3]$. The acoustic models are trained by discriminative training and by using speaker adaptive training. The language model is a tri-gram model with Kneser-Ney smoothing that is trained on the LUNA HM corpus. The ASR has a word error rate of 21.3% on the test set.

4.2. Baseline systems

The baseline SLU system is trained by using conditional random fields (CRFs) [15]. To make it comparable with the neural network models we have not used additional features over the word features. The model uses a window of [-1, +1] to extract the word features and uses the value of the previous concept when predicting the current one.

The standard well-performing dialogue system baselines are based on information retrieval models that select a system response with respect to their similarity to training conversations (user or system turns) [16, 17]. For spoken conversation data, it has been observed that computing similarity to other user turns and selecting linked responses yields better results than computing similarity to system responses directly [16]. Consequently, we compute TF-IDF weighted cosine similarity between a test user utterance (as bag-of-words) and all training user utterances, and select the system response of the user turn with the highest score as a candidate (Nearest Neighbor model of [16]). Additionally, we also compute the cosine similarity over the pre-trained word embeddings on Italian Wikipedia [18]. The pre-trained word embeddings have a size of 300 and trained over a window of [-5,5] by using the skip-gram model [7]. Similarities computed considering just the current user turn are used as baselines for the evaluation of the next turn generation performance on the turn level in Table 1. To be able to approach to the stateful dialogue model, similarities are computed also by considering the previous system turn which are reported in Table 2 as baselines.

4.3. Training turn embeddings

Turn embeddings are learned using the proposed sequence-to-sequence model. The sequence-to-sequence networks with the following architectures are evaluated: disjoint SLU model ("SLU"), disjoint next system turn generation model ("ST[+1]"), joint SLU model with the next system turn generation model ("SLU + ST[+1]"), joint SLU model with the previous and the next system turn generation model ("SLU + ST[-1,+1]"), joint SLU model with two previous and two next system turns generation model ("SLU + ST[-2,+2]").

The size of the LSTM cells and the size of the embeddings are 256 for both the user turns and the system turns. The vocabulary size for the user turns is 2300 and the vocabulary for the system turns is around 300 after replacing each ticket number the system generates with a special token. At every iteration the cost is computed on the development set and the learning rate is halved when the cost increases on the development set. Early stopping is performed when no further improvement is obtained on the development set. During the testing phase the system turns are obtained by greedy decoding (with decoding with a beam size of 1). The performance of the sequence-to-sequence models with the relevant baselines are given in Table 1. For each network architecture, 10 models are trained with different random initializations, only the best performing models on the reference transcription of the development set is reported. The models are implemented using Tensorflow library [19].

Table 1: *Performance of the sequence-to-sequence models on SLU and next system turn (NST) generation performance. The baselines are presented with the 'BL:' label. Evaluations are done on the reference transcriptions, "Ref.", and on the ASR transcriptions, "ASR", of the test set.*

Model	**SLU (CER)**		**NST (BLEU)**	
	Ref.	ASR	Ref.	ASR
BL: CRF	26.0%	30.5%	NA	NA
BL: TF-IDF	NA	NA	30.4	**30.7**
BL: Word emb.	NA	NA	**30.7**	27.0
ST[+1]	NA	NA	24.3	23.5
SLU	25.2%	28.6%	NA	NA
SLU + ST[+1]	23.4%	28.0%	28.3	26.1
SLU + ST[-1,1]	**23.0%**	27.3%	**28.8**	**27.0**
SLU + ST[-2,2]	23.7%	**27.1%**	28.3	26.9

As can be seen from Table 1 the neural network SLU models outperforms the CRF baseline and more robust to ASR noise. We observe that training the SLU jointly with system turns improves the SLU performance. "SLU + ST[-1,1]" model, which considers the preceding and succeeding turns, performs the best among other neural network models. Adding more context does not improve the performance. The performance of sequence-to-sequence models are below the baselines for next turn generation performance. However, we use the turn embeddings obtained form these models to train stateful dialogue models at a later stage, and do not use them to generate turns.

Turn embeddings are qualitatively compared to dialogue acts (DAs) as defined in the recent international ISO standard for DA annotation – Dialogue Act Markup Language (DiAML) [20]. The DiAML annotation scheme consists of 56 DA tags (communicative functions), organized into 9 dimensions, such as General, Social Obligations Management, Time Management, etc. The user turns in the LUNA HM test set are automatically annotated for ISO dimensions and DA tags using models trained on LUNA human-human corpus by [21]; and manually corrected. We visualize the user turns obtained from ASR hypotheses by a t-Distributed Stochastic Neighbor Embedding (t-SNE) [22] plot in Figure 4. We only show the most frequent DA tags: Answer, Inform, Confirm and show the whole Social dimension as a single class. We observe that turn embeddings cluster user turns with respect to their DAs.

4.4. Dialogue model

The training of the dialogue model (DiaM) is performed as follows. The turn embeddings for the training and the development set are obtained by using each sequence-to-sequence model that are presented in the previous section. The dialogue model uses the same size of LSTM cells and the same size of embeddings as the sequence-to-sequence models. During training the learning rate is adjusted by calculating the cost on the development set and early stopping is applied. The system turn is obtained by using greedy decoding. As done for the turn embedding models each dialogue model is trained with 10 different random initializations and the performance (BLEU score) of these models are presented in Table 2 with the relevant baselines. The best performing models on the reference transcription of the development set are reported with the average scores of the 10 different random initializations in parentheses.

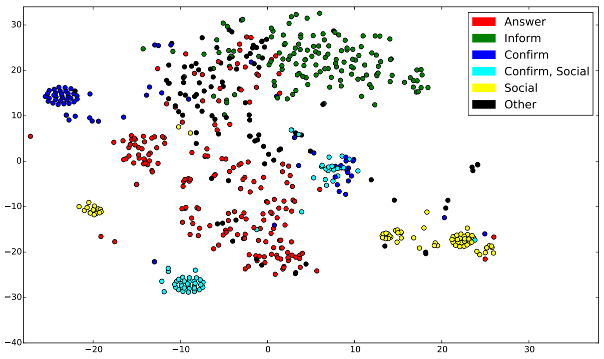

Figure 4: *t-SNE plot of turn embeddings for the ASR hypotheses of the test set. Colors represent DA classes.*

Table 2: *Performance (BLEU score) of the dialogue model and dialogue system baselines with local history on the test set.*

Model	**Reference**	**ASR**
BL: TF-IDF	35.1	36.0
BL: Word emb.	29.4	32.2
DiaM with ST[+1]	42.2 (41.4)	39.2 (39.5)
DiaM with SLU	43.0 (38.8)	41.1 (38.6)
DiaM with SLU + ST[+1]	41.4 (41.1)	40.7 (40.0)
DiaM with SLU + ST[-1,1]	**45.2 (41.8)**	**46.0 (42.8)**
DiaM with SLU + ST[-2,2]	41.9 (40.3)	44.8 (41.7)

We observe that the best performing dialogue model is the one which is trained with the turn embeddings obtained from the "SLU + ST[-1,1]" model. Although, the embeddings from the "SLU" model has a good best performance, the average score is low, i.e., it is hard to optimize the dialogue model consistently with these turn embeddings. Including the previous system turns in the turn embeddings makes the dialogue model robust to ASR noise. We obtain no additional gain by increasing the window size of the system turns, on the contrary, the performance drops. Finally, all of the dialogue models outperform the baselines, since the dialogue models use the whole dialogue history and the baselines only use a local history.

5. Conclusion

In this paper we propose turn embeddings as robust representations of user turns for building spoken dialogue systems. Turn embeddings can be trained by predicting the preceding and succeeding system turns from the user turn by using sequence-to-sequence models. Training an SLU model jointly with the sequence-to-sequence models improves the performance of the SLU model which is a crucial component of spoken dialogue systems. Also, turn embeddings learn representations that cluster turns with respect to DAs implicitly. A dialogue model that considers the whole dialogue history is built by using the pre-trained turn embeddings. We observe that the pre-trained turn embeddings trained with the previous system turns makes the dialogue model more robust to ASR noise.

6. References

[1] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *ACL*. The Association for Computer Linguistics, 2015, pp. 1577–1586.

[2] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Hierarchical neural network generative models for movie dialogues," *CoRR*, vol. abs/1507.04808, 2015. [Online]. Available: http://arxiv.org/abs/1507.04808

[3] T. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P. Su, S. Ultes, D. Vandyke, and S. J. Young, "A network-based end-to-end trainable task-oriented dialogue system," *CoRR*, vol. abs/1604.04562, 2016. [Online]. Available: http://arxiv.org/abs/1604.04562

[4] X. Yang, Y. Chen, D. Z. Hakkani-Tür, P. Crook, X. Li, J. Gao, and L. Deng, "End-to-end joint learning of natural language understanding and dialogue manager," in *Proceedings of ICASSP*. IEEE, 2017.

[5] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1724–1734.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

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

[10] D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. V. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Interspeech 2016*, June 2016.

[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

[12] M. Dinarelli, S. Quarteroni, S. Tonelli, A. Moschitti, and G. Riccardi, "Annotating spoken dialogs: from speech segments to dialog acts and frame semantics," in *Proc. of EACL Workshop on the Semantic Representation of Spoken Language*, Athens, Greece, 2009, pp. 34–41.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*. IEEE, 2011.

[15] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*. Morgan Kaufmann, 2001, pp. 282–289.

[16] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," *CoRR*, vol. abs/1605.07683v3, 2017.

[17] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–June 2015, pp. 196–205. [Online]. Available: http://www.aclweb.org/anthology/N15-1020

[18] "Italian word embeddings," http://tanl.di.unipi.it/embeddings/.

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[20] H. Bunt, J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. R. Traum, "ISO 24617-2: A semantically-based standard for dialogue annotation." in *LREC*, 2012, pp. 430–437.

[21] S. A. Chowdhury, E. A. Stepanov, and G. Riccardi, "Transfer of corpus-specific dialogue act annotation to ISO standard: Is it worth it?" in *Language Resources and Evaluation Conference (LREC)*. Portorož, Slovenia: ELRA, May 2016, pp. 132–135.

[22] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.