Convolutional Neural Encoder for the 7th Dialogue System Technology Challenge

Mandy Korpusik, James Glass
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
korpusik@mit.edu, glass@mit.edu

Abstract

In this paper, we demonstrate that our convolutional neural network (CNN) approach to selecting the next best system response in a spoken dialogue system is competitive on the recently released 7th Dialogue System Technology Challenge (DSTC7). Our CNN model outperforms the strong dual LSTM encoder baseline, placing us in 11th place out of 20 participants on the first subtask of the Advising dataset in track one, where the goal is to select the next best response given the previous dialogue history between a student and the advisor, who are discussing the best courses for the student to enroll in. We show that our learned CNN filters identify semantically meaningful categories of tokens, such as greetings and course names, illustrating the interpretability of convolutional models for dialogue response generation.

Introduction

With the rise of conversational agents such as Siri[1] and Cortana,[2] dialogue systems that interact with people through spoken language are becoming more and more popular. Typically, however, these commercial products are still heavily rule-based. With the abundance of data and more powerful computation available to us today, we can apply increasingly powerful models to machine learning problems. In particular, neural networks have been shown to outperform prior state-of-the-art statistical models in computer vision and speech recognition, and can learn to handle raw input without requiring any manual feature engineering. Thus, we propose applying neural methods to spoken dialogue systems, allowing the models to handle raw natural language internally, with minimal pre-processing required.

The standard pipeline of steps in a spoken dialogue system is shown in Fig. 1, where the user query (either written or spoken) is fed through an automatic speech recognizer if it is spoken, and the text of the query is sent to the language understanding component. In this component, the semantic tagging step involves isolating *slots* and corresponding slot *values* (e.g., in a flight booking system, the `departure city` slot may have the value `Boston`). The database retrieval then looks up the relevant information (e.g., flights) from a knowledge base. This information is passed to the dialogue manager, which updates the state of the dialogue (i.e., the user's goals, which are usually represented as a set of slots and matching slot values) and determines the next action the system should take, such as asking a followup question about the user's preferred departure time.

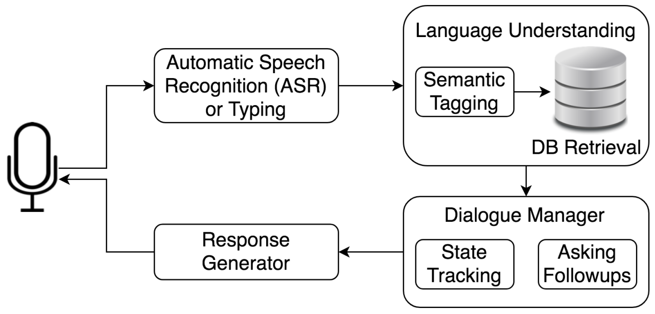


Figure 1: A standard pipeline for a spoken dialogue system, where the input spoken user query is passed to an automatic speech recognizer to generate its text, and the generated text or the input textual user query is sent to the language understanding component for pre-processing and semantic tagging. Then, the information is passed to the dialogue manager to update the state of the dialogue and predict the next action the system should take to generate a desired response.

In this paper, we focus on the final response generation step, taking the user's query and dialogue history as the only input, completely bypassing the language understanding and dialogue management components. Therefore, the model is trained fully end-to-end, with only two inputs: the previous two utterances in a dialogue between a student and the advisor, and a candidate utterance for the next system response. Given a set of 100 possible response candidates, the goal is to select the next best response based on the dialogue history (see Table 1 for an example partial dialogue). We evaluate our convolutional neural network approach, which does not require any manual feature engineering or semantic dictionaries, on the recently released 7th Dialogue System Technology Challenge (DSTC7), focusing in particular on subtask one of the Advising dataset in track one.[3]

[1]https://www.apple.com/siri

[2]https://www.microsoft.com/en-us/cortana

[3]https://ibm.github.io/dstc7-noesis/public/index.html

The challenge provides five subtasks: 1) selecting the next response from among 100 candidates, 2) selecting the next response from 120,000 responses, 3) selecting the next response and its paraphrases from among 100 candidates, 4) selecting the next response from among 100 candidates that may not contain the correct response, and 5) selecting the next response from among 100 candidates and incorporating the provided external knowledge. We demonstrate that our convolutional neural encoder places 11th out of 20 participants on the first subtask, with a *Recall@50* score of 82.4%.

Advisor: Hello Mingyang! Are you doing well?
Student: Hi advisor. I'm doing alright. I would like some advice on which courses to take next semester.
Student: My interested area is Software Development and Intelligent system.
Advisor: you have three choices namely, EECS481 Software Engineering, EECS492 Introduction to Artificial Intelligence, and EECS 381 Object Oriented and Advanced Programming.
Student: how many difficulty levels do these classes have?
Advisor: EECS381 is not easy
Advisor: any thoughts about that?
Student: What time does the course occur? I like afternoon classes and will find something else if it's scheduled too early.
Advisor: EECS351 is after lunch. The others are before. EECS481 is from nine to ten thirty and EECS492 is from ten thirty to twelve.

Table 1: Example DSTC7 dialogue snippet, between a student and the advisor. 100 candidate responses are provided, and the system must choose the correct response, given the conversation history: "*481 is the early morning and is quite similar to EECS381, so you might want to skip it.*"

Related Work

CNNs for Sentence Matching

In our work, we use CNNs to learn a shared embedding space for input student and advisor utterances in a conversation about selecting the best courses for the student. Recent work (Adi et al. 2016) has analyzed the relative strengths of various other sentence embeddings, including averaging word vectors learned with the continuous-bag-of-words method (Mikolov et al. 2013), LSTM auto-encoders (Li, Luong, and Jurafsky 2015), and skip-thought vectors based on gated recurrent units (GRU) (Kiros et al. 2015). Our approach differs from these in that we use a CNN, as in related work on learning semantic vector embeddings of natural language meal logs and food database items (Korpusik, Collins, and Glass 2017b; 2017a; Korpusik and Glass 2018a), rather than recurrent networks, and we learn the vectors through a domain-specific task for predicting whether a candidate system response is a good match for previous utterances.

Similar work in learning joint embeddings for two different modalities or languages have explored a margin-based contrastive loss, which would be interesting to compare against our binary verification cross-entropy loss. For ranking annotations given an image, prior work directly incorporated the rank into the model's loss function, along with a hinge loss between true and false annotation samples (Weston, Bengio, and Usunier 2011); similarly, a margin-based loss was used to learn a joint multimodal space between images and captions for caption generation (Karpathy and Fei-Fei 2015; Harwath, Torralba, and Glass 2016), and sentence/document embeddings were learned through a multilingual parallel corpus with a noise-contrastive hinge loss ensuring non-aligned sentences were a certain margin apart (Hermann and Blunsom 2014). Other related work predicted the most relevant document given a query through the cosine similarity of jointly learned embeddings based on bag-of-words term frequencies (Huang et al. 2013).

Many researchers are now exploring CNNs for natural language processing (NLP). For example, in question answering, recent work has shown improvements using deep CNN models for text classification (Conneau et al. 2017; Zhang, Zhao, and LeCun 2015; Xiao and Cho 2017) following the success of deep CNNs for computer vision. Whereas these architectures take in a simple input text example and predict a classification label, our task takes in two input sentences and predicts whether they match. In work more similar to ours, parallel CNNs predict the similarity of two input sentences. While we process each input separately, others first compute a word similarity matrix between the two sentences (like an image matrix of pixels) and use the matrix as input to one CNN (Pang et al. 2016; Wang, Mi, and Ittycheriah 2016; Hu et al. 2014).

Attention-based CNN (ABCNN) models have also been proposed for sentence matching. The ABCNN (Yin et al. 2016) combines two approaches: applying attention weights to the input representations before convolution, as well as after convolution but before pooling. Our method is similar, but we compute dot products (our version of the attention scheme) with the max-pooled high-level vector representation of the dialogue history. Hierarchical ABCNN applies cosine similarity attention between CNN representations of a query and each sentence in a document for machine comprehension (Yin, Ebert, and Schütze 2016). Thus, the attention comes after pooling across the input, whereas we compute the dot products between each token in the candidate system response, and the learned vector representing the previous utterances in the dialogue.

Dialogue State Tracking

We evaluate our system on the 7th Dialogue System Technology Challenge, which has been a premier research competition for dialogue systems since its inception in 2013. The seventh challenge focuses on end-to-end dialogue tasks, in order to explore the issue of applying end-to-end technologies to dialogue systems in a pragmatic way, while previous challenges focused on Dialogue State Tracking (DST), or tracking the user's goal over time. Traditionally, spoken dialogue systems relied on separately trained components for spoken language understanding (SLU) and dialogue state tracking. The SLU component would identify

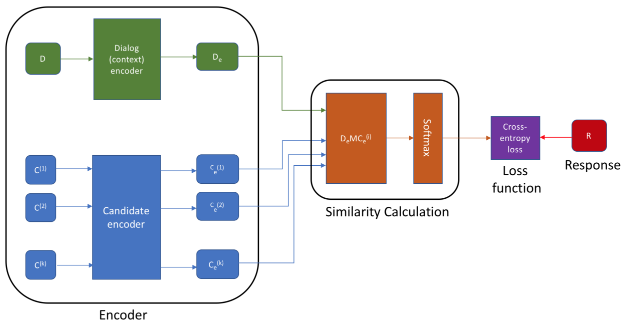


Figure 2: The strong baseline dual encoder LSTM network for predicting the next best system response.

slot-value pairs from the speech recognition output, which would be passed to the state tracking module to update the belief state (Thomson and Young 2010; Wang and Lemon 2013). However, this pipeline of steps would accumulate errors, as the SLU component often would not have the necessary context to accurately predict the slot values. Thus, belief tracking research shifted to jointly predicting slot-value pairs and updating the dialogue state (Henderson, Thomson, and Young 2014b; Sun et al. 2014).

Typically, these jointly trained SLU and dialogue state updating models rely on a delexicalization-based strategy, which translates various instantiations of slot and value mentions in the user utterance into generic labels. Prior work by *Henderson et al.* fed delexicalized user utterances into a recurrent neural network, which output a distribution over slot values (Henderson, Thomson, and Young 2014a). However, delexicalizing the input often requires a manually defined semantic dictionary that maps from slot-value pairs to all possible text forms, or synonyms.

To avoid this reliance on hand-crafted semantic dictionaries, *Mrksic et al.* recently demonstrated the ability of their Neural Belief Trackers (NBT) (Mrkšić et al. 2016) to match the performance of delexicalization-based models, without requiring any hand-crafted semantic dictionaries, as well as the ability to significantly outperform such models when the semantic resources are not available. In addition, *Zhong et al.'s* state-of-the-art work has explored deep learning methods for dialogue state tracking, with recurrent self-attentive encoders (Zhong, Xiong, and Socher 2018). Their self-attentive RNN model encodes user utterances, system actions, and each slot-value pair under consideration. *Rastogi et al.* also fed delexicalized utterances into their multi-domain deep learning model for state tracking (Rastogi, Hakkani-Tür, and Heck 2017), and *Korpusik et al.* took a similar approach, but without using pre-trained word vectors or delexicalization (Korpusik and Glass 2018b).

Models

Baseline Dual LSTM Encoder

We compare our model to a strong baseline—the dual long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) encoder (Lowe et al. 2015). The inspiration for this model is the sequence-to-sequence (seq2seq) (Sutskever, Vinyals, and Le 2014) approach often applied to machine translation, where an encoder (usually a recurrent model, such as an LSTM) encodes the input sentence in the source language, and an LSTM decoder is fed the encoded representation to generate the sentence in the target language.

The LSTM encoder works as follows (see Fig. 2).[4] Each token in the dialogue history is fed through an embedding layer (initialized with Glove (Pennington, Socher, and Manning 2014)), followed by a recurrent layer, to yield an encoded vector representation of the context, D_e. Likewise, each token in a candidate response is fed through an embedding layer and a recurrent layer, generating the encoded representation C_e. This is done for each of the 100 candidate responses, and the similarity of each candidate response with the dialogue context is computed using a learned similarity matrix M, via the matrix multiplication $D_e M C_e^i$, where i refers to the index of the candidate response. Each of these similarity scores is fed through a final softmax layer to generate probabilities of each candidate response. The whole model is trained with cross-entropy loss.

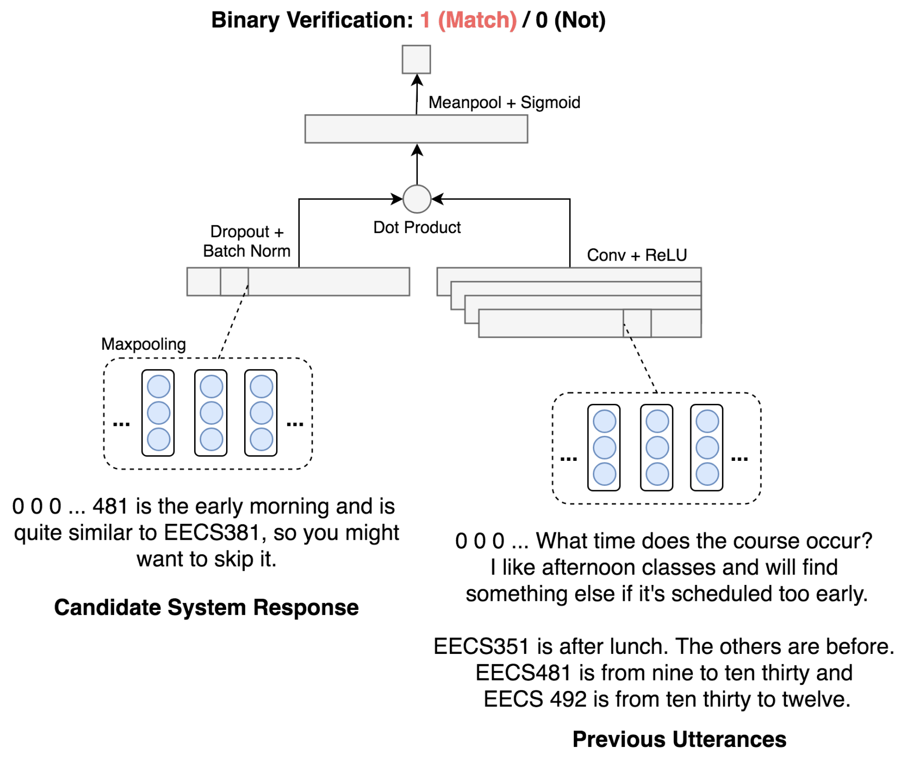
Convolutional Encoder

Our approach is similar to that of the dual LSTM encoder, but with the differences that: (i) we use a convolutional neural network (CNN) instead of the LSTM, (ii) we only feed the last two utterances into the context encoder rather than the full dialogue history, which would likely require attention over all the previous utterances in order to ensure the most important information from the most recent utterances is not lost among the full dialogue history, as shown in related work on multi-turn dialogue (Wu et al. 2016; Zhou et al. 2018), and (iii) we compare against each candidate response one at a time with a sigmoid layer instead of a softmax over all candidates. Our motivation for using convolutional rather than recurrent models is they train faster, and are more easily interpretable by inspecting which tokens for each learned filter have the highest activation.

As shown in Fig. 3, the model is composed of two inputs: one input layer for the previous two utterances in the conversation, concatenated together, and another input layer for a candidate advisor response. Each of the two inputs is first tokenized using spaCy,[5] lowercased, and padded with zeros to a fixed length of 50 tokens. Each candidate system response is fed through a shared word embedding layer (note that we do not use pre-trained word vectors in the system we submitted to the challenge, although we have since compared to pre-training with Glove (Pennington, Socher, and Manning 2014) and word2vec (Mikolov, Chen, and Dean) on the validation set), and is max-pooled to generate a single 256-dimension vector representation of the dialogue context. At the same time, the dialogue history is fed through the 64-dimension shared word embedding layer and a 1-dimension convolutional layer of 256 filters spanning a window of three tokens with a rectified linear unit (ReLU) activation. Dur-

[4]http://www.wildml.com/2016/07/deep-learning-for-chatbots-2-retrieval-based-model-tensorflow/

[5]https://spacy.io



Binary Verification: 1 (Match) / 0 (Not)
Meanpool + Sigmoid
Dropout + Batch Norm
Dot Product
Conv + ReLU
Maxpooling
0 0 0 ... 481 is the early morning and is quite similar to EECS381, so you might want to skip it.
Candidate System Response
0 0 0 ... What time does the course occur? I like afternoon classes and will find something else if it's scheduled too early.
EECS351 is after lunch. The others are before. EECS481 is from nine to ten thirty and EECS 492 is from ten thirty to twelve.
Previous Utterances

Figure 3: The CNN architecture for predicting whether a given candidate system response is the correct match for the previous two utterances in the dialogue between a student and their advisor.

ing training, this is followed by a dropout of probability 0.1, and batch normalization (Ioffe and Szegedy 2015). Following the input encoding step, a dot product is performed with the candidate response vector and each 256-dimension CNN output of the dialogue history. Mean-pooling is then performed across these dot products to produce a single scalar value, which we force to be between zero and one with a final sigmoid layer. We train the model with binary cross-entropy and the Adam optimizer (Kingma and Ba 2014).

Convolutional Ensemble

In the convolutional encoder described above, the final ranking of the candidate responses is generated by computing the sigmoid output probability for each response. We experimented with ensembling several randomly initialized convolutional models, and found that an ensemble of seven models (six with 256-dimension embeddings and learned CNN filters, and one with 128 dimensions instead), performed best, while the best individual model was the CNN with 256 dimensions. We also found that ensembling by summing the ranked indices predicted by individual models performed better on the development set than averaging the predicted probability scores for each candidate response.

Experiments

In the 7th Dialogue System Technology Challenge (DSTC7) (Yoshino et al. 2018), the goal is to predict the next utterance in the dialogue, given the previous utterances in the dialogue history. There are two corpora—the Ubuntu corpus (Kummerfeld et al. 2018), which is based on chat logs from the Ubuntu channels, and Advising data that simulates a discussion between a student and an academic advisor. We focus on the Advising data, where the purpose of the dialogue is to guide the student to pick the courses that best fit their curriculum, as well as personal preferences about time, difficulty, and area of interest. These conversations were collected by asking students at the University of Michigan to play the role of both the student and the advisor, using provided personas. The statistics of the data for the first subtask are shown in Table 2.

	min	max	mean	median
History length	1	41	9.2	8
Utterance length	1	384	10.3	9
Candidate answer length	1	384	12.4	10

Table 2: The Advising dataset's statistics for subtask 1.

For our experiments, we focus on the first subtask, in which there are 100 candidate responses for each dialogue snippet, where only one is the correct match, and the others are distractors. We evaluate performance with the *Recall@n* metric, where *R@n* indicates how often the model ranked the correct response among the top-n. The second metric we use for evaluation is the mean reciprocal rank (MRR), which is the average of the reciprocal ranks (i.e., the multiplicative inverse of the rank of the first correct answer) of results for a sample of candidate responses Q:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad (1)$$

To select the best model, we evaluated individual CNNs, as well as ensembles, on the validation set for Advising subtask 1 (see Table 3), and compared against the dual LSTM encoder baseline. In Table 4, we show the Recall and MRR scores of our system on the two held-out test sets for the Advising subtask 1. Our ensemble of seven convolutional encoders outperforms the dual LSTM encoder baseline, placing us in 11th place for this subtask, and ranked 13 out of 20 total participants in the first track of the DSTC7 challenge.

Model	R@1	R@10	R@50
Dual LSTM Encoder	6.20	36.0	80.0
Single CNN (meanpool)	3.43	27.2	97.9
Single CNN (maxpool)	10.9	46.3	97.2
2-CNN Ensemble	11.1	48.0	97.2
3-CNN Ensemble	11.8	47.5	97.4
4-CNN Ensemble	12.0	46.7	97.0
5-CNN Ensemble	12.2	46.3	97.0
6-CNN Ensemble	12.4	46.9	97.0
7-CNN Ensemble	12.6	46.9	97.0
8-CNN Ensemble	12.4	46.9	97.0
Single CNN (Glove)	12.2	**50.3**	**98.3**
Single CNN (word2vec)	**14.8**	47.5	97.0

Table 3: We report the recall for several methods on the validation dataset for Advising subtask 1. Optimizing for R@1, we select the 7-CNN ensemble for the final evaluation (since at the time of submission, we were not using pre-trained word embeddings). With more than 7 CNNs, performance starts dropping. Note that with mean-pooling instead of max-pooling over the candidate response, recall is lower.

Model	**Data**	R@1	R@10	R@50	MRR
CNN Ensemble	Test 1	20.6	54.8	82.4	32.3
CNN Ensemble	Test 2	8.8	32.0	72.8	16.9

Table 4: We report recall and mean reciprocal rank (MRR) for our CNN on the two test sets for Advising subtask 1.

Analysis

Since one common critique of neural network models is that they are mysterious "black boxes," we analyze the learned CNN filters in order to make the model's behavior more interpretable. Specifically, we identify which tokens in the development set yield the highest activation for each of the learned CNN filters, as in related work (Korpusik and Glass 2017; Korpusik et al. 2016; 2014), and manually inspect these top-10 highest activation tokens to determine whether there is an intuitive pattern. In Table 5, we can see that filter 1 seems to identify greetings (e.g., "hello" and "hi"), filter 11 picks out tokens where the student thanks the advisor and ends the dialogue (e.g., "thanks" and "goodbye"), filter 144 isolates tokens related to course names (e.g., "operating system" and "eecs"), and filters 185 and 209 seem to identify tokens related to personal preferences such as time and workload (e.g., "difficult" and "morning").

CNN Filter	**Top-10 Tokens**
1	'hello', 'for', 'hi', 'today', '?', 'afternoon', 'full', 'doing', 'one', 'in'
11	'thankful', 'for', 'goodbye', ',', 'thanks', '!', 'thank', '.', 'bye', 'will'
144	'system', 'operating', '482', 'heard', 'really', 'any', 'calc', 'eecs', 'last', 'have'
185	'programming', 'difficult', 'light', 'course', 'in', 'workload', 'take', 'load', 'junior', 'computing'
209	'morning', 'light', 'class', 'in', 'relatively', 'semester', 'prefer', 'a', 'like', 'which'

Table 5: Top-10 activated tokens for learned CNN filters.

In addition, we inspect the errors made by our best system (i.e., the ensemble of CNNs) on the validation set to determine 1) whether the mistakes seem reasonable, and 2) to come up with ideas for improving performance. In Table 6, we see that the system is confused by out-of-vocabulary words (i.e., <`UNK`>) that were unseen during training. One approach for handling this better in future work is to use letter trigrams or character-based embeddings, rather than full word embeddings. We also note that in the second example, the student thanks the advisor, and the predicted responses all seem reasonable (e.g., *"you're welcome"*). In the third example mistake, where the system is unable to compare two courses, the limitation is that our system does not use the full conversation history, but only the previous two utterances. In this scenario, it would help the system to know that "they" refers to courses EECS370 and EECS280. The final mistake requires a deeper semantic understanding than our system is capable of currently—the correct response requires commonsense reasoning that by registering for the class, the student can observe firsthand the professor's style, and the predicted responses make sense, but illustrate that the system does not realize "it" refers to courses rather than the professor (e.g., *"it has a high degree of easiness"*).

Conclusion

In this paper, we have demonstrated that a CNN encoder outperforms a strong dual LSTM encoder baseline for selecting the correct response from a list of possible candidates. We report a *Recall@50* score of 82.4% on the first held-out test set for the Advising subtask 1 of the first track of the 7th Dialogue System Technology Challenge (DSTC7). In future work, we will explore methods of incorporating more contextual information and leveraging the full dialogue history and provided knowledge base of course information. In order to handle unknown words at test time, we will investigate character embeddings and delexicalization. We will apply our models to the other subtasks and to the Ubuntu dataset.

Student: does eecs370 have <UNK> ? **Correct**: nope , no labs in eecs 370 . **Top-4 Predicted**: 1) no <UNK> . have a good one ! 2) it starts at 3:00 and ends at 4:30pm. 3) i would suggest taking eecs 280 4) how much challenge do you want ?
Student: many thanks , and i hope to never see you again . **Correct**: my wish for you is that you graduate quickly . **Top-4 Predicted**: 1) you 're welcome . 2) i believe that is a good decision . i wish you luck . 3) what time of day do you need ? 4) do you need anything else ?
Student: do they pair well together ? **Correct**: it depends on your schedule . eecs 370 is a easy class whereas , according to statistics <UNK> from past year students , eecs 281 is not an easy class , so i recommend <UNK> it with an easier class . **Top-4 Predicted**: 1) it depends 2) the students do not like eecs376 . 3) i suggest strongly you take eecs 280 4) you have met all the requirements , so it is possible .
Student: as a professor how is he ? **Correct**: if you want , you can see for yourself by registering for the class . **Top-4 Predicted**: 1) well , that depends 2) 370 has a score of <UNK> for being relatively <UNK> . 3) it has a high degree of easiness , helpfulness , and clarity 4) it 's not a very heavy load

Table 6: Examples of incorrect top-4 response predictions.

Acknowledgements

This research was sponsored by a grant from Quanta Computing, Inc., and by the Department of Defense (DoD) through the National Defense Science Engineering Graduate Fellowship (NDSEG) Program. Special thanks to Mitra Mohtarami for her helpful feedback and advice.

References

Adi, Y.; Kermany, E.; Belinkov, Y.; Lavi, O.; and Goldberg, Y. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Conneau, A.; Schwenk, H.; Lecun, Y.; and Barrault, L. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 11071116.

Harwath, D.; Torralba, A.; and Glass, J. 2016. Unsupervised learning of spoken language with visual context. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*, 1858–1866.

Henderson, M.; Thomson, B.; and Young, S. 2014a. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 360–365. IEEE.

Henderson, M.; Thomson, B.; and Young, S. 2014b. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 292–299.

Hermann, K., and Blunsom, P. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of Advances in neural information processing systems (NIPS)*, 2042–2050.

Huang, P.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2333–2338. ACM.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3128–3137.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.

Korpusik, M., and Glass, J. 2017. Spoken language understanding for a nutrition dialogue system. *IEEE Transactions on Audio, Speech, and Language Processing*.

Korpusik, M., and Glass, J. 2018a. Convolutional neural networks and multitask strategies for semantic mapping of natural language input to a structured database. In *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6174–6178. IEEE.

Korpusik, M., and Glass, J. 2018b. Convolutional neural networks for dialogue state tracking without pre-trained word vectors or semantic dictionaries. In *Proceedings of 2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Korpusik, M.; Schmidt, N.; Drexler, J.; Cyphers, S.; and Glass, J. 2014. Data collection and language understanding of food descriptions. *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*.

Korpusik, M.; Huang, C.; Price, M.; and Glass, J. 2016. Distributional semantics for understanding spoken meal descriptions. *Proceedings of 2016 IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Korpusik, M.; Collins, Z.; and Glass, J. 2017a. Character-based embedding models and reranking strategies for understanding natural language meal descriptions. *Proceedings of Interspeech.*

Korpusik, M.; Collins, Z.; and Glass, J. 2017b. Semantic mapping of natural language input to database entries via convolutional neural networks. *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Kummerfeld, J. K.; Gouravajhala, S. R.; Peper, J.; Athreya, V.; Gunasekara, C.; Ganhotra, J.; Patel, S. S.; Polymenakos, L.; and Lasecki, W. S. 2018. Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *arXiv preprint arXiv:1810.11118.*

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057.*

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909.*

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mikolov, T.; Chen, K.; and Dean, J. word2vec (2013).

Mrkšić, N.; Séaghdha, D.; Wen, T.; Thomson, B.; and Young, S. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777.*

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; and Cheng, X. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2793–2799.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global vectors for word representation. *Proceedings of 2014 Conference on Empirical Methods on Natural Language (EMNLP)* 12.

Rastogi, A.; Hakkani-Tür, D.; and Heck, L. 2017. Scalable multi-domain dialogue state tracking. In *Proceedings of 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 561–568. IEEE.

Sun, K.; Chen, L.; Zhu, S.; and Yu, K. 2014. The SJTU system for dialog state tracking challenge 2. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 318–326.

Sutskever, I.; Vinyals, O.; and Le, Q. 2014. Sequence to sequence learning with neural networks. In *Proceedings of Advances in neural information processing systems (NIPS)*, 3104–3112.

Thomson, B., and Young, S. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language* 24(4):562–588.

Wang, Z., and Lemon, O. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, 423–432.

Wang, Z.; Mi, H.; and Ittycheriah, A. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1340–1349.

Weston, J.; Bengio, S.; and Usunier, N. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI)*, volume 11, 2764–2770.

Wu, Y.; Wu, W.; Xing, C.; Zhou, M.; and Li, Z. 2016. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:1612.01627.*

Xiao, Y., and Cho, K. 2017. Efficient character-level document classification by combining convolution and recurrent layers. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference*, 353–358.

Yin, W.; Schütze, H.; Xiang, B.; and Zhou, B. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. In *Transactions of the Association for Computational Linguistics*, volume 4, 259–272.

Yin, W.; Ebert, S.; and Schütze, H. 2016. Attention-based convolutional neural network for machine comprehension. In *Proceedings of 2016 NAACL Human-Computer Question Answering Workshop*, 15–21.

Yoshino, K.; Hori, C.; Perez, J.; D'Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B.; Gao, S.; Marks, T. K.; Parikh, D.; and Batra, D. 2018. The 7th dialog system technology challenge. *arXiv preprint.*

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Proceedings of Advances in neural information processing systems (NIPS)*, 649–657.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655.*

Zhou, X.; Li, L.; Dong, D.; Liu, Y.; Chen, Y.; Zhao, W.; Yu, D.; and Wu, H. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1118–1127.