Comparison of Transfer-Learning Approaches for Response Selection in Multi-Turn Conversations

Jesse Vig and Kalai Ramea
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

This paper compares three transfer-learning approaches to response selection in dialogs, as part of the Dialog System Technology Challenge 7 (DSTC7) Track 1. In the first approach, Multi-Turn ESIM+ELMo (MT-EE), we incorporate pre-trained contextual embeddings into a sentence-pair model that was originally designed for natural language inference. In the second approach, we fine-tune the Generative Pre-trained Transformer (OpenAI GPT) model. In the third approach, we fine-tune the Bidirectional Encoder Representations from Transformers (BERT) model. Our results show that BERT performed best, followed by the GPT model and then the MT-EE model. We also discuss the relative advantages and disadvantages of each approach. The submitted result for Track 1 (MT-EE) placed second and fifth overall for the Advising and Ubuntu datasets respectively.

Introduction

The Dialog Systems Technology Challenges (DSTC) have been conducted since 2013 to advance the state of the art in dialog systems. This year, three tracks have been introduced for DSTC7: (1) sentence selection, (2) sentence generation, and (3) audio visual scene-aware dialogs (Yoshino et al. 2018). In this paper, we focus on the first track (sentence selection), where the goal is to pick the next utterance in a partial dialog given a set of candidate responses.

We present three approaches to solving this task. In the first approach, we adapt ESIM (Enhanced Sequential Inference Model), which was designed for natural language inference at the sentence level (Chen et al. 2017), to a dialog setting by introducing a multi-turn aggregation scheme. We also incorporate ELMo (Embeddings from Language Models) pre-trained contextualized embeddings into this model (Peters et al. 2018). We refer to this approach as Multi-Turn ESIM+ELMo (*MT-EE*) throughout this paper.

For the second and third approaches, we apply two pre-trained end-to-end models: the Generative Pre-trained Transformer from OpenAI (*OpenAI GPT*) (Radford et al. 2018), and the Bidirectional Encoder Representations from Transformers (*BERT*) model (Devlin et al. 2018). The results from these two models were not submitted to the competition; BERT was released after the challenge ended, and the OpenAI model was released after the June 1 deadline specified in the challenge rules.

In the following sections, we outline the challenge task, discuss related work, describe our methodology in detail, and present a comparative analysis.

Table 1: Subtasks in DSTC7 Track 1

No.	Subtask Description
1	Select the next response from a set of 100 choices that contains 1 correct response (applies to both datasets)
2	Select the next response from a set of 120,000 choices (applies only to Ubuntu data)
3	Select the next response(s) from a set of 100 choices that contains between 1 and 5 correct responses (applies only to Advising data)
4	Select the next response or NONE from a set of 100 choices that contains 0 or 1 correct response (applies to both datasets)
5	Select the next response from a set of 100 choices, given access to an external dataset (applies to both datasets)

Task Description

Track 1 of DSTC7 includes five subtasks, each of which involves selecting the next response in a dialog from a set of possible choices. As shown in Table 1, the subtasks vary in the number of candidate responses, the number of correct responses within the candidate set, and the availability of an external knowledge base.

Each subtask covers one or both of two datasets: the Advising dataset, which includes play-acted dialogs between a student and her advisor; and the Ubuntu dataset, which consists of technical support conversations. Each dataset includes partial conversation between two speakers, along with a set of choices for the next utterance. The response being chosen comes from the advisor in the case of the Advising dataset, and from the technical support provider in the Ubuntu dataset. Figure 1 shows an illustration of partial dialogs from these two datasets. The Advising dataset includes additional student profile information such as the student's previous courses, recommended courses, and grade

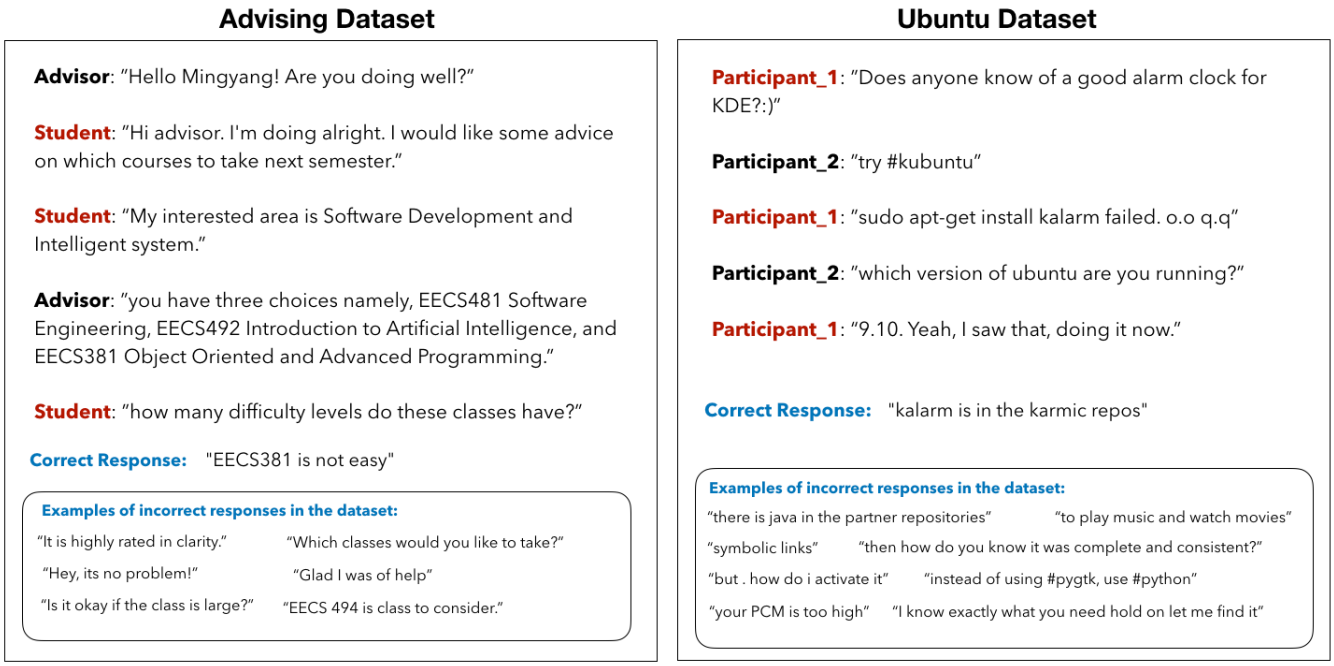

Figure 1: Examples of partial conversations in the DSTC datasets

level. This information may be used in all of the Advising subtasks. For Subtask 5, additional external datasets are also made available: course information for the Advising dataset, and Linux manual pages for the Ubuntu dataset.

Related Work

Various approaches have been developed for response selection in dialogs, ranging from embedding-based approaches (Lowe et al. 2015) to attentional models (Tay, Tuan, and Hui 2018). In this paper we present solutions that draw from recent work in two areas: transfer learning and sentence-pair models. We briefly describe these two areas below and discuss how they relate to the proposed solutions.

Transfer Learning and Pre-trained Models in Natural Language Processing

Using pre-trained models for transfer learning has revolutionized the field of computer vision (Mahajan et al. 2018). Practitioners have been able to take the model weights learned from a large image repository like ImageNet (Deng et al. 2009) and fine-tune them for their own image processing problems (e.g. classification, object detection, etc.). This has not only reduced the computing time and memory usage, but also increased accuracy and precision, especially when using smaller datasets.

Recently, similar approaches have been developed in the field of natural language processing (Ruder 2018). By pre-training on a large corpus on tasks such as language modeling, deep representations are generated that can serve as a starting point to a variety of NLP tasks. Examples of pre-trained models includes ELMo (Peters et al. 2018), ULMFiT (Howard and Ruder 2018), OpenAI GPT (Radford et al. 2018) and most recently BERT (Devlin et al. 2018). In contrast to fixed word embeddings such as Word2Vec (Mikolov et al. 2013) or Glove (Pennington, Socher, and Manning 2014), the newer embeddings incorporate the context in which the word is used. Additionally, many of these embeddings, e.g. ELMo, are computed at the character level and can encode previously unseen words.

We incorporate pre-trained models in all approaches presented in this paper. For the MT-EE model, we use pre-trained ELMo embeddings in the lowest layer, and we learn the remaining model weights from scratch. For the OpenAI GPT and BERT models, we use the full, end-to-end architecture and fine-tune the weights to our task.

Sentence-Pair Models

Many NLP problems may be formulated as a sentence-pair prediction task (Lan and Xu 2018). For example, paraphrase identification predicts whether two sentences have the same meaning; natural language inference determines whether a hypothesis sentence can be inferred from a premise sentence; and question-answering tasks rank candidates by assigning a score to a question-answer pair. Many sentence-pair models have been developed to solve the various tasks discussed above, and recent work shows that these models are often effective across a range of tasks besides the ones they were designed for (Lan and Xu 2018).

One such sentence-pair model is the Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017), which was designed for natural language inference but has also been shown to be effective for other sentence-pair tasks such as paraphrase identification and question-answering (Lan and Xu 2018). ESIM combines multiple BiLSTM layers with

a matrix attention layer to predict the relationship between two sentences. In this paper, we adapt the ESIM model to the task of response selection in dialogs by introducing a multi-turn aggregation method. The closest work to ours is by Dong and Huang (Dong and Huang 2018), which also applies ESIM to response selection but uses a different aggregation method and a custom embedding scheme.

Methodology

In this section we describe our general approach to response selection and present three specific implementations: (1) Multi-Turn ESIM + ELMo (MT-EE), (2) OpenAI GPT, and (3) BERT.

General Approach

We approach the problem of response selection by defining a matching function that measures the affinity between a dialog context (the previous utterances), and each candidate response. Specifically, we implement a matching function that computes the probability that the response is correct for the given context. We then use this matching function to rank the candidate responses for a context. To train the matching function, we first create a dataset of context-response pairs in which half of the pairs contain a response that is correct for the corresponding context (positive example), and half contain a response that is incorrect (negative example). We build this dataset from the existing dialog corpus, including the correct response for each dialog as well as one randomly selected incorrect response. We use this methodology for all three approaches described below. The only difference between them is the matching function.

Multi-Turn ESIM + ELMo (MT-EE)

As discussed earlier, ESIM is a sentence-pair model, i.e. a model that predicts an output given two input sentences. The present task is related to sentence-pair modeling in that we are predicting an output for a context-response pair. In this case, however, the context is not a single sentence but rather a sequence of previous utterances. Moreover, each utterance in the context is associated with a specific speaker, which might impact the significance of that utterance relative to the end task.

We adapt ESIM to the multi-turn setting by applying it to individual utterances within the context and aggregating the results. The aggregation scheme is motivated by the following characteristics of dialog: (1) the relevance of an utterance depends on its position within the context, and (2) the relation of an utterance to the response depends on the speaker of the utterance. We capture the first element by applying a position-based weighted average, and we encode the second element through a speaker-specific residual layer. Many other aggregation schemes are possible such as utterance concatenation or inter-utterance RNNs (Tian et al. 2017). Initial exploration of these approaches yielded inferior results so we did not pursue them further.

We apply transfer learning to our approach through the use of ELMo embeddings. ELMo embeddings are contextual and have been shown to be much more effective than fixed embeddings, e.g., Word2Vec (Peters et al. 2018). Moreover, as ELMo embeddings are character-based, they can capture the domain-specific entities and terminology in both datasets. The combination of ESIM and ELMo had previously achieved state-of-the-art performance in natural language inference and continues to be one of the top algorithms in the Stanford NLI leaderboard (Bowman et al. 2015) at the time of submission.[1]

Model Structure An overview of the approach is shown in Figure 2. Formally, the task of sentence selection is to choose a response r from a set of candidates R, given a partial dialog context $C = \{c_1, ...c_i, ..., c_n\}$, where c_i is the i$^{\text{th}}$ utterance in the partial dialog. We denote the speakers corresponding to these utterances as $S = \{s_1, ..., s_i, ..., s_n\}$, where s_i is the speaker for the i$^{\text{th}}$ utterance in the context. For each dataset, there are two possible speakers, which we refer to as Speaker 1 and Speaker 2. In the Advising dataset, Speaker 1 is the student and Speaker 2 is the advisor. For the Ubuntu dataset, Speaker 1 is the person seeking help, while Speaker 2 is the one giving advice.

Our goal is to define a matching function $F(C, r)$ that can be used to rank each response candidate r for a given context C. The first step in our approach is to compare each $c_i \in C$ with r. Here we apply the sentence-pair model ESIM+ELMo to encode a vector that represents the relationship between c_i and r. Instead of directly using the ESIM outputs, which are class probabilities, we strip away the final layers and use the previous vector output. We denote this truncated model as $ESIM^-$ and define the output vector v_i as follows:

$$v_i = ESIM^-(c_i, r) \tag{1}$$

Each vector v_i is then transformed based on the speaker s_i. We do this by passing v_i through a speaker-specific residual layer with weights W_{s_i} as shown in Figure 2. We chose a residual architecture based on the intuition that most of the information is speaker-independent, and therefore the speaker effect should be an offset to the input vector rather than a complete transformation. We denote the speaker-informed output vector as $v_i^{'}$:

$$v_i^{'} = v_i + relu(W_{s_i} v_i) \tag{2}$$

We then compute a weighted average of these output vectors using learned weights $\{\alpha_i\}$, as shown in equation 3. These weights capture the varying importance of utterances based on their position within the context, as illustrated in Figure 2.

$$v_{out} = \frac{\sum_{i=1}^{n} \alpha_i v_i^{'}}{\sum_{i=1}^{n} \alpha_i} \tag{3}$$

We apply a final layer to the weighted average to obtain the matching function $F(C, r)$:

$$F(C, r) = softmax(W_F v_{out}) \tag{4}$$

We train this model on the binary labeled dataset described above, with cross entropy as the loss function.

[1] https://nlp.stanford.edu/projects/snli/

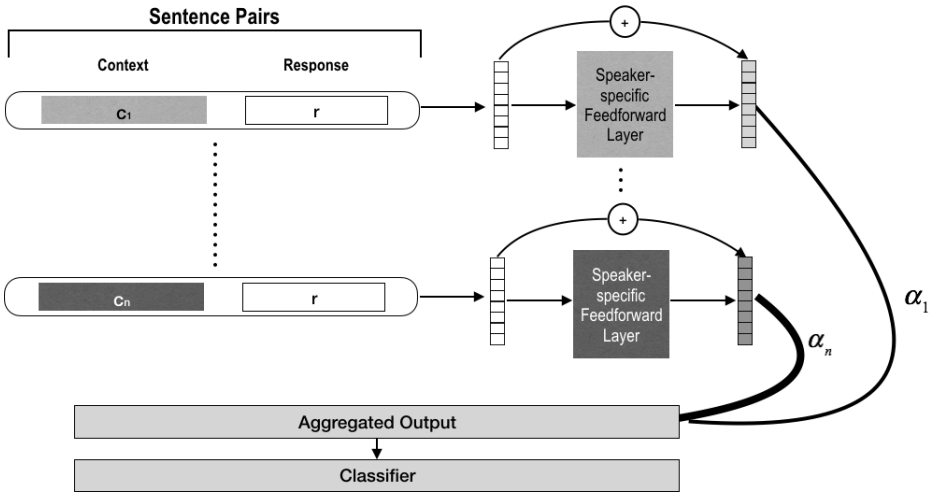

Figure 2: Aggregation scheme in the MT-EE model

Implementation Details We implemented our model using PyTorch (Paszke et al. 2017) and the AllenNLP framework, including its implementation of ESIM (Gardner et al. 2018). We used the spaCy tokenizer (Honnibal and Johnson 2015) and additionally split words on boundaries between alpha and numeric characters. For the Advising dataset we concatenated consecutive utterances from the same speaker into a single utterance. (In the Ubuntu dataset, each turn contains just one utterance.) We also incorporated student profile information from the Advising dataset by extracting relations, e.g., whether the token matched a suggested course offering. These properties were then appended to the ELMo embeddings.

Besides batch size and learning rate, the hyperparameters for the ESIM model itself were unchanged from the original implementation. For the speaker-specific residual layers in our model, we set a dropout rate of 0.5 (Srivastava et al. 2014). We used Adam (Kingma and Ba 2014) for optimization. For the Advising dataset we limited the context size to the 4 previous turns, and for the Ubuntu dataset we used the previous 6 turns, because larger contexts yielded the same or worse performance on the validation set.

Modifications for Specific Subtasks For some of the subtasks (see Table 1) we modified the approach described above. Details are listed below:

1. For Subtask 2, where the number of candidate responses was 120,000 rather than 100, we implemented a computationally efficient pre-filter using the Universal Sentence Encoder[2] (Cer et al. 2018). We used this prefilter to identify semantically similar responses to the last two turns of the dialog and ranked them. We then used the ranked list to reduce the candidate set to 100 and then fed this smaller subset into the main model.
2. For Subtask 4, in which NONE was a valid response if we believed that the correct response was not present, we introduced a parameter ψ representing the fixed matching score for NONE. We added NONE to each pool of candidate responses and ranked it according to ψ. Grid search was used to determine the value of ψ that maximized the mean of Recall@1, Recall@10, Recall@50, and MRR.

[2]https://tfhub.dev/google/universal-sentence-encoder/1

End-to-End Pre-trained Models

We also implemented two end-to-end pre-trained models: the Generative Pre-trained Transformer[3] developed by OpenAI (OpenAI GPT) (Radford et al. 2018), and the Bidirectional Encoder Representations from Transformers (BERT) model[4] developed by Google AI (Devlin et al. 2018).

For both models, the standard approach for sentence-pair tasks is to concatenate the two sentences, along with a separator token, into a single input sequence. We take this same approach for the present task, by concatenating context and response (and separator token). To capture the structure within the context, we also insert speaker-specific delimiter tokens between the utterances in the context: $< EOU1 >$ if the preceding utterance came from Speaker 1, or $< EOU2 >$ if it was from Speaker 2. Thus the structure is encoded through delimiters tokens rather than through a structured aggregation approach as in MT-EE.

Results and Analysis

In this section, we report results for Subtask 1 of Track 1 (see Table 1). We focus on this subtask because it allows us to compare the models for the most standard form of the response selection task. Table 2 shows the performance of the MT-EE, OpenAI GPT, and BERT models on both datasets.

[3]We used a PyTorch implementation of this model: https://github.com/huggingface/pytorch-openai-transformer-lm

[4]We used the BERT-Base, Cased model: https://github.com/google-research/bert

Note that the submitted result for the Advising dataset (MT-EE model) incorporates the provided student profile information. In order to provide a fair comparison with the other two approaches, which did not use the student profile data, we also ran the MT-EE model on the Advising dataset without the profile information.

Table 2: Results on DSTC test datasets for Subtask 1 (Recall@1, Recall@10, Recall@50, Mean Reciprocal Rank). We include two versions of the Advising dataset: one with the student profile information, and one without.

Dataset	Model	R@1	R@10	R@50	MRR
Advising	***MT-EE****	***0.152***	***0.574***	***0.930***	***0.286***
Advising (w/o profile)	MT-EE	0.132	0.512	0.890	0.252
	OpenAI GPT	0.172	0.568	0.932	0.293
	BERT	0.186	0.580	0.942	0.312
Ubuntu	***MT-EE****	***0.478***	***0.765***	***0.952***	***0.578***
	OpenAI GPT	0.489	0.799	0.972	0.595
	BERT	0.530	0.817	0.978	0.632

* *Submitted result.*

For both datasets, BERT performs best, followed by OpenAI GPT and then MT-EE. This is consistent with results reported on the GLUE dataset (Devlin et al. 2018), in which BERT outperformed OpenAI GPT on all tasks, and OpenAI GPT outperformed all other models on several of the tasks. BERT's performance is substantially better than the other models on both datasets. For example, if we look at Recall@1 for the Ubuntu dataset, OpenAI GPT's improvement over MT-EE is around 2.3%, while BERT's improvement over MT-EE is around 10.9%. One possible reason for the superior performance of both over the MT-EE model is that these models are pre-trained in an end-to-end fashion, whereas in the MT-EE model, only the embeddings are pre-trained. Another explanation is that these two models use self-attention mechanism across the entire context while the MT-EE model treats each utterance independently.

However, the MT-EE model does yield reasonable performance despite the fact that it uses components designed for a different task (natural language inference) without making changes to their internal structure or hyperparameters. This gives further evidence for the versatility and generality of sentence-pair models, building on the work of Lan and Xu (Lan and Xu 2018). A computational advantage of MT-EE is that its complexity is linear in the number of utterances, versus both the BERT and OpenAI GPT models, which have self attention layers with complexity that is quadratic in the overall sequence length (and hence number of utterances).

BERT's superior performance over OpenAI may be attributed to differences in both architecture and pre-training methods. BERT uses a bidirectional self-attention mechanism in contrast to OpenAI's left-to-right attention model. Accordingly, BERT is pre-trained using a masked language model, where the goal is to predict an unknown (masked) token given the left and right context, whereas OpenAI is pre-trained using a left-to-right language model. Ablation studies in (Devlin et al. 2018) suggest that the bidirectional nature of BERT plays a key role in its superior performance across a range of tasks.

In contrast to OpenAI, BERT is also pre-trained on the *next sentence prediction* (NSP) task, where the model is given two input sentences and must predict whether the second sentence follows the first. Like response selection, NSP is a sentence-pair task, so one would expect that pre-training on NSP would improve performance on response selection. In fact, ablation studies in (Devlin et al. 2018) show that NSP pre-training is particularly important for the QNLI answer-selection task, which is closely related to response selection.

We also performed more detailed analyses on the role of context in the three models, focusing on the Ubuntu dataset. Table 3 shows the performance of the models when using contexts of varying sizes, i.e. with a varying number of previous dialog utterances included. As Figure 3 illustrates, performance initially improves for all three models as more turns are added to the context, but the effect diminishes and appears to plateau around 6 to 8 turns[5]. Adding turns beyond

Table 3: Results on Ubuntu dataset (Recall@1, Recall@10, Recall@50, Mean Reciprocal Rank) based on number of previous turns included in context.

Model	# Turns	R@1	R@10	R@50	MRR
MT-EE	1	0.223	0.500	0.859	0.322
	2	0.375	0.667	0.899	0.476
	4	0.448	0.730	0.928	0.545
	6*	***0.478***	***0.765***	***0.952***	***0.578***
	8	0.486	0.754	0.941	0.577
OpenAI GPT	1	0.241	0.538	0.864	0.348
	2	0.390	0.704	0.934	0.503
	4	0.477	0.795	0.970	0.588
	6	0.497	0.786	0.974	0.599
	8	0.498	0.794	0.956	0.602
	no limit	0.489	0.799	0.972	0.595
BERT	1	0.302	0.602	0.891	0.402
	2	0.466	0.770	0.942	0.572
	4	0.516	0.820	0.972	0.624
	6	0.557	0.836	0.979	0.650
	8	0.549	0.823	0.983	0.649
	no limit	0.530	0.817	0.978	0.632

* *Submitted result.*

[5]For BERT and OpenAI GPT models, we set overall sequence

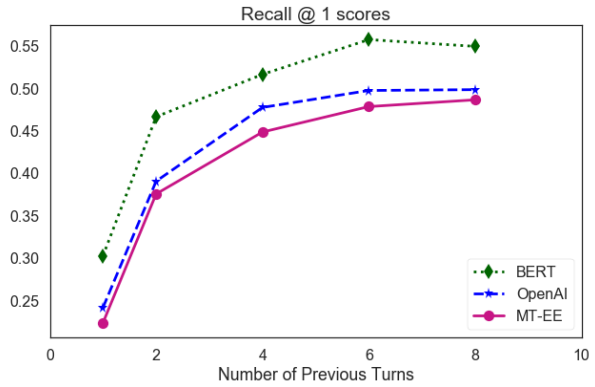

Figure 3: Plot of Recall@1 scores of MT-EE, OpenAI GPT and BERT models on Ubuntu dataset, with varying context size

this point decreases performance in some cases. This is most evident for BERT, which achieved Recall@1 of 0.557 when using just 6 turns compared to a value of 0.530 without the turn limit, an increase of 5.1%. This suggests that more context is not always better and there may be a benefit to treating context size as a hyperparameter to optimize.

Table 4: Learned weights α_i over previous turns for MT-EE model on Ubuntu dataset using context size of 8 turns

Turn Index i	**Speaker 1**	**Speaker 2**
n	0.375	–
n - 1	–	0.061
n - 2	0.251	–
n - 3	–	0.050
n - 4	0.117	–
n - 5	–	0.044
n - 6	0.118	–
n - 7	–	0.046

Table 4 shows the turn weights α_i learned by the MT-EE model on the Ubuntu dataset, broken out by speaker and position within the context. One observation is that higher weights are assigned to utterances from Speaker 1, suggesting the model finds that speaker's content more informative. This is not surprising, since the target utterance (from Speaker 2) is a response to Speaker 1. Although the weights for Speaker 2's utterances are smaller, they are still greater than zero. This makes sense as the target utterance may continue topics mentioned in Speaker 2's previous utterances. A second observation is that the weights decrease as the distance from the target response increases. This makes sense since the most recent utterances are more likely to relate to the response.

length to 512 tokens due to memory limitations (this includes both context and response). A turn consists of 20 tokens on average when using OpenAI GPT tokenization.

Conclusion

This paper compares three approaches for response selection in multi-turn conversations as part of DSTC7, Track 1. In the first approach (MT-EE), we introduced a dialog-specific aggregation scheme that uses a position-based weighted average and speaker-specific layers to select the next response from a given set of candidates. The results generated using MT-EE were submitted for the challenge, and placed second overall for the Advising dataset, and fifth for the Ubuntu dataset. We compared these results to two end-to-end pre-trained models released after the challenge start date: OpenAI GPT and BERT.

We used transfer learning techniques for all three approaches but through different methods and pre-trained models. For MT-EE, we only used pre-trained weights for the ELMo embedding layer, and for OpenAI GPT and BERT, we used fully pre-trained end-to-end transformer models. We observed that BERT performs the best followed by OpenAI GPT; however, both are constrained in sequence length due to the fact that complexity increases quadratically with sequence length. The MT-EE model yields reasonable performance despite the fact that the underlying ESIM algorithm was designed for a different task, reflecting the versatility of sentence-pair models.

The three models presented in this paper reflect the advances in NLP transfer learning made over the course of DSTC7. Our first model, MT-EE, uses ELMo contextual embeddings, which were released a few months prior to the challenge. Within two weeks of the start of the challenge, the OpenAI GPT achieved SOTA results on a variety of tasks. Just after the challenge ended, but before the submission of this paper, BERT was released, establishing a new SOTA on all GLUE tasks. We expect that the rapid evolution of pre-trained models will continue to transform the field of natural language processing in the coming months.

For future work, we would like to perform ablation studies on the MT-EE model to determine the effect of the different components of the model, including the speaker-specific residual layers and the position-based weighted averaging. We would also like to explore hybrid models that combine the structured aggregation approach with transformer models such as BERT, in order to address the challenges of scaling the latter to very long sequences.

References

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strope, B.; and Kurzweil, R. 2018. Universal sentence encoder. *CoRR* abs/1803.11175.

Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for Natural Language Inference. *Association for Computational Linguistics*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Li, F.-F. 2009. Imagenet: A large-scale hierarchical image database.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv Computation and Language*.

Dong, J., and Huang, J. 2018. Enhance word representation for out-of-vocabulary on ubuntu dialogue corpus. In *ArXiv Computation and Language*.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. Allennlp: A deep semantic natural language processing platform.

Honnibal, M., and Johnson, M. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1373–1378. Lisbon, Portugal: Association for Computational Linguistics.

Howard, J., and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. *Association for Computational Linguistics*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Lan, W., and Xu, W. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *ArXiv Computation and Language*.

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. *CoRR* abs/1506.08909.

Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; and Maaten, L. v. d. 2018. Exploring the limits of weakly supervised pretraining. In *ArXiv Computer Vision and Pattern Recognition*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. *The North American Chapter of the Association for Computational Linguistics*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

Ruder, S. 2018. NLP's imagenet moment has arrived.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958.

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2018. Multi-Cast Attention Networks for Retrieval-based Question Answering and Response Prediction. *Knowledge Discovery and Data Mining*.

Tian, Z.; Yan, R.; Mou, L.; Song, Y.; Feng, Y.; and Zhao, D. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

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*.