Knowledge-incorporating ESIM models for Response Selection in Retrieval-based Dialog Systems

Jatin Ganhotra
IBM Research
jatinganhotra@us.ibm.com

Siva Sankalp Patel
IBM Research
siva.sankalp.patel@ibm.com

Kshitij Fadnis
IBM Research
kpfadnis@us.ibm.com

Abstract

Goal-oriented dialog systems, which can be trained end-to-end without manually encoding domain-specific features, show tremendous promise in the customer support use-case e.g. flight booking, hotel reservation, technical support, student advising etc. These dialog systems must learn to interact with external domain knowledge to achieve the desired goal e.g. recommending courses to a student, booking a table at a restaurant etc. This paper presents extended Enhanced Sequential Inference Model (ESIM) models: a) K-ESIM (Knowledge-ESIM), which incorporates the external domain knowledge and b) T-ESIM (Targeted-ESIM), which leverages information from similar conversations to improve the prediction accuracy. Our proposed models and the baseline ESIM model are evaluated on the Ubuntu and Advising datasets in the *Sentence Selection* track of the latest Dialog System Technology Challenge (DSTC7), where the goal is to find the correct next utterance, given a partial conversation, from a set of candidates. Our preliminary results suggest that incorporating external knowledge sources and leveraging information from similar dialogs leads to performance improvements for predicting the next utterance.

Introduction

The *Dialog State Tracking Challenge* (DSTC) was initially started to serve as a testbed for dialog state tracking task and was rebranded as *Dialog System Technology Challenges* in 2016 to provide a benchmark for evaluating dialog systems. The latest DSTC challenge, DSTC7[1], is divided into three different tracks. Our work addresses the *Sentence Selection* track, where the objective is to push the utterance classification towards real world problems.

Retrieval-based dialog systems, in comparison to generation-based dialog systems, have the benefit of informative and fluent responses, as the proper response for the current conversation is selected from a set of candidates responses. The main goal for a dialog system in the Sentence Selection track is that it should:

- select the correct next utterance(s) from a set of candidates, given a partial conversation.
- learn to select none of the candidates if none of them align with the given partial conversation.
- learn to incorporate external knowledge sources.

Goal-oriented dialog systems usually rely on external knowledge sources, such as restaurant names and details for restaurant table booking task, course information at a university for student advising task and flight details for flight reservation task. The external knowledge could be provided in a structured format (a knowledge base (KB)) or unstructured format (man pages for Linux commands). It is essential for a dialog system to incorporate the external knowledge source for task completion and achieving the desired goal. Bordes, Boureau, and Weston (2016) introduced bAbI dialog tasks, as a simulation for restaurant table booking task but interaction with the domain KB was circumvented as restaurant details relevant to a given dialog were provided as part of the dialog history. Eric and Manning (2017) proposed Key-Value retrieval networks, which performs attention over the entries of a KB to extract relevant information. Lowe et al. (2015a) explored incorporating unstructured external textual information (man pages) for the next utterance classification task on Ubuntu dialog corpus. In this work, we extend the Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017) and propose two end-to-end models for next utterance selection:

- K-ESIM (Knowledge-based ESIM) incorporates the additional unstructured external textual information.
- T-ESIM (Targeted ESIM) leverages information from similar dialogs seen during training.

The paper is structured as follows. In the next section, we briefly describe the problem and the two datasets used for evaluating our proposed models. Then, we present related work and introduce the baseline ESIM model and our proposed models: K-ESIM and T-ESIM. In the Experiments and Results section, we present our evaluation results across all models on the two datasets. Finally, we draw conclusion from our work and indicate directions of future work.

Problem Statement and Datasets

The DSTC7 challenge for *Sentence Selection* track consists of two datasets: a) Ubuntu dataset and b) Advising dataset. Both datasets share the common goal to predict the correct

[1] http://workshop.colips.org/dstc7/index.html

next utterance from a set of potential next utterance candidates, given a partial conversation. In addition to selecting the correct next utterance, the dialog system is also evaluated on its ability to identify that none of provided candidates is a good next utterance for the given partial conversation. The 5 subtasks and the 2 datasets are described below:

- **Subtask 1**: The Subtask 1 serves as the baseline task for both datasets. The goal is to select the next utterance for the partial conversation from the given candidate set, which contains 1 correct and 99 incorrect next utterances.
- **Subtask 2**: Subtask 2 increases the task complexity by testing the dialog system to select the next utterance for the partial conversation from a large global pool of next utterance candidates. Subtask 2 is evaluated only on the Ubuntu dataset, where the total number of candidates in the global pool is 120000.
- **Subtask 3**: The goal of Subtask 3 is to evaluate the dialog system to select all correct next utterances from the given set of candidates. Subtask 3 is evaluated only on the Advising dataset, where the given candidate set can contain 1 to 5 correct next utterances (original correct utterance and paraphrases) and 95 to 99 incorrect next utterances.
- **Subtask 4**: Subtask 4 extends Subtask 1 and serves as a benchmark to check if the dialog system can learn to identify when correct next utterance for the partial conversation is not available in the given candidate set. For such cases, the dialog system must respond with '*None*' as next utterance. Subtask 4 is evaluated on both datasets.
- **Subtask 5**: In Subtask 5, the dialog system is evaluated on its ability to incorporate additional external knowledge provided and is evaluated on both datasets. The expectation for Subtask 5 is that the dialog system must perform better after incorporating external knowledge.

Ubuntu dialog corpus

The Ubuntu dialog corpus provided as part of the challenge is a new version of disentangled two-party conversations from Ubuntu IRC logs (Kummerfeld et al. 2018). The purpose is to solve an Ubuntu user's posted problem, i.e. select the correct next utterance based on the conversation so far between the two users. The training data contains over 100k complete conversations, and the test data contains 1000 partial conversations, where each dialog has a minimum of 3 turns. In addition to the dialog corpus, additional knowledge is also provided in the form of linux manual pages.

Advising dataset

The Advising data contains conversations between a student and advisor, where the goal of the advisor is to guide the student to pick courses that a) align with the student's curriculum and b) match the student's personal preferences about time (when classes are held e.g. morning, afternoon etc.), course difficulty (easy, hard etc.), career path etc. The dataset collected is play-acted where two students act as the two roles using provided personas. The data also includes paraphrases of the sentences and of the target responses by the advisor. In addition to the dataset, an additional knowledge base (KB) is provided which contains information about various courses and possible personal preferences for the students. The training data contains 100,000 partial dialogs from the original 500 dialogs. The test data consists of 500 partial dialogs, where a set of 100 candidates are provided which includes 1-5 correct next utterances.

Additional details for the Ubuntu and Advising datasets, along with the external knowledge sources for both datasets are provided in the *Sentence Selection* track description document[2].

Related Work

End-to-end dialog systems, based on neural networks, have shown the promise of learning directly from human-to-human dialog interactions. They have performed well in non goal-oriented chit-chat settings (Vinyals and Le (2015); Sordoni et al. (2015); Serban et al. (2016); as well as goal-oriented settings (Le, Dymetman, and Renders (2016), Ghazvininejad et al. (2017), Bordes, Boureau, and Weston (2016), Seo et al. (2016)). There are two active research directions for building goal-oriented dialog systems: *generative*, where the system generates the next utterance in the conversation word-by-word, and *retrieval-based*, where the system has to pick the next utterance from a list of potential responses. Response selection has been actively explored by the research community in the last few years (Dong and Huang (2018), Wu et al. (2016), Bartl and Spanakis (2017)).

Ubuntu dialog corpus introduced by Lowe et al. (2015b) consists of conversations from the Ubuntu IRC channel where users ask and answer technical questions about Ubuntu. The diversity of conversations and large quantity of dialogs available in the corpus presents a unique challenge for goal-oriented dialog systems. Lowe et al. (2015b) proposed Dual-encoder architecture which uses Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) to embed both context and response into vectors and response selection is based on the similarity of embedded vectors. Kadlec, Schmid, and Kleindienst (2015) built an ensemble of convolution neural network (CNN) Krizhevsky, Sutskever, and Hinton (2012) and Bi-directional LSTM. Dong and Huang (2018) integrated character embeddings (dos Santos et al. 2015) into Enhanced LSTM method (ESIM) (Chen et al. 2017) and achieve significant performance improvement.

There have been several studies on incorporating unstructured external information into dialog models. Ghazvininejad et al. (2017) proposed a knowledge-grounded neural conversation model by improving the seq2seq approach to produce more contentful responses. The model was trained using Twitter Dialog Dataset (Li et al. 2016) as dialog context and foursquare data as the external information. Young et al. (2017) incorporated structured knowledge from knowledge graphs and achieved an improved dialog system over the Twitter Dialog Dataset. Lowe et al. (2015a) used the Linux manual pages as the external knowledge information for improving the next utterance prediction task and showed reasonable accuracy gains.

[2] https://ibm.github.io/dstc7-noesis/public/data_description.html

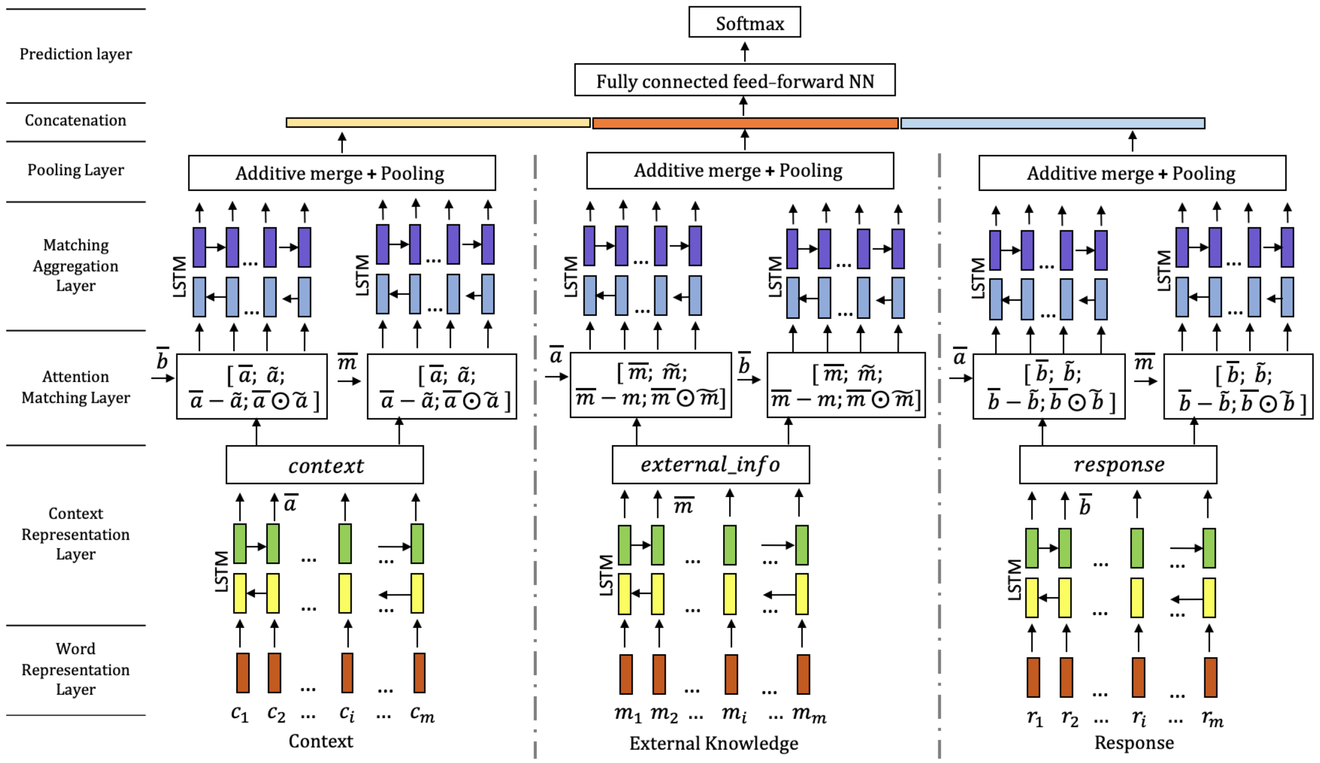
Prediction layer
Concatenation
Pooling Layer
Matching Aggregation Layer
Attention Matching Layer
Context Representation Layer
Word Representation Layer
Softmax
Fully connected feed-forward NN
Additive merge + Pooling
Additive merge + Pooling
Additive merge + Pooling
LSTM
$[\bar{a};\ \tilde{a};\ \bar{a}-\tilde{a};\bar{a}\odot\tilde{a}]$
$[\bar{m};\ \tilde{m};\ \bar{m}-m;\bar{m}\odot\tilde{m}]$
$[\bar{b};\ \tilde{b};\ \bar{b}-\tilde{b};\bar{b}\odot\tilde{b}]$
$\bar{b}$
$\bar{m}$
$\bar{a}$
context
external_info
response
c_1 c_2 ... c_i ... c_m
Context
m_1 m_2 ... m_i ... m_m
External Knowledge
r_1 r_2 ... r_i ... r_m
Response

Figure 1: **K-ESIM**: A high-level overview of K-ESIM model, which incorporates external knowledge.

Baseline model: ESIM

We use the ESIM model proposed by Chen et al. (2017) as the baseline model. The implementation details for the baseline model are provided in Appendix. As mentioned in the 'Problem Statement' section, the task is to select the next response given the dialog history (context). The multi-turn dialog history is concatenated together to form the context of length m, represented as $C = (c_1, c_2, ..., c_i, ..., c_m)$, where c_i is the ith word in context. Given a response R with length n as $R = (r_1, r_2, ..., r_j, ..., r_n)$ where r_j is the jth word in response, the next response is selected using the conditional probability $P(y = 1|C, R)$, which shows the confidence of selecting the response R given context C.

Word Representation Layer:

We represent each word in the context and the response with a d-dimension vector. Following Dong and Huang (2018), the word representation is generated by concatenating the word embedding and the character-composed embedding for the word. The word embedding for each word is generated by concatenating the Glove word embeddding (Pennington, Socher, and Manning 2014) and the word2vec embedding (Mikolov et al. 2013) for the word. The word2vec vectors are generated by training on the training data, where each sentence is treated as a document. The character-composed embedding, introduced by (Dong and Huang 2018) is generated by concatenating the final state vector of the forward and backward direction of bi-directional LSTM (BiLSTM).

Context representation layer:

The context representation layer utilizes BiLSTM to capture word representation and it's local sequence context. The hidden states at each time step for both directions are concatenated to form a new local context-aware word representation, denoted by $\bar{a}$ and $\bar{b}$ for context and response below.

$$\bar{a}_i = BiLSTM(\bar{a}_{i-1}, w_i), 1 \le i \le m \qquad (1)$$

$$\bar{b}_j = BiLSTM(\bar{b}_{j-1}, w_j), 1 \le j \le n \qquad (2)$$

Attention matching layer: As in ESIM model, the co-attention matrix where $E \in \mathbb{R}^{m*n}$ where $E_{ij} = \bar{a}_i^T \bar{b}_j$ computes the similarity between context and response. The attended response vector computed via equation 3 represents the most relevant response word for each word in context. Similarly, the attented context vector is computed via equation 4 and represents the most relevant context word for each word in response.

$$\tilde{a}_i = \sum_{j=1}^{n} \frac{exp(E_{ij})}{\sum_{k=1}^{n} exp(E_{ik})} \bar{b}_j, 1 \le i \le m \qquad (3)$$

$$\tilde{b_j} = \sum_{i=1}^{m} \frac{exp(E_{ij})}{\sum_{k=1}^{m} exp(E_{kj})} \bar{a}_i, 1 \le j \le n \qquad (4)$$

Using the attended context and response vectors from equations 3 and 4, vector difference and element-wise product is computed to further elicit interaction information between context and response. The difference and element-wise product vectors are concatenated with the original vectors to generate m_a^i and m_b^i as shown below.

$$m_a^i = [\bar{a}_i, \tilde{a}_i; \bar{a}_i - \tilde{a}_i; \bar{a}_i \odot \tilde{a}_i], 1 \leq i \leq m \tag{5}$$

$$m_b^i = [\bar{b_i}, \tilde{b_i}; \bar{b_i} - \tilde{b_i}; \bar{b_i} \odot \tilde{b_i}], 1 \leq j \leq n \tag{6}$$

Matching aggregation layer:

In this layer, another BiLSTM is used to aggregate response-aware context representations and context-aware response representations (Chen et al. 2017). The matching aggregation layer learns to compose local inference information sequentially using the BiLSTM, as shown below.

$$v_i^a = BiLSTM(v_{i-1}^a, m_i^a), 1 \leq i \leq m, \tag{7}$$

$$v_j^b = BiLSTM(v_{j-1}^b, m_j^b), 1 \leq j \leq n. \tag{8}$$

Pooling layer:

We use max pooling via combining max pooling and final state vectors (concatenation of both forward and backward one) $(v_{last}^a; v_{last}^b)$ to form the final fixed vector (Dong and Huang 2018), which is calculated as follows:

$$v_{max}^a = \max_{i=1}^{m} v_i^a \tag{9}$$

$$v_{max}^b = \max_{j=1}^{n} v_j^b \tag{10}$$

$$v = [v_{max}^a; v_{max}^b; v_{last}^a; v_{last}^b] \tag{11}$$

v from equation 11, is fed into the final prediction layer (a fully-connected feed-forward neural network).

Proposed model: K-ESIM

We use the baseline ESIM model described above and update the model architecture for incorporating external knowledge e.g. command description from man pages for Ubuntu dataset. We refer to the new model as K-ESIM, (Knowledge incorporating ESIM model). A high-level overview of K-ESIM is shown in Figure 1. We introduce the following changes to the baseline ESIM model:

Word Representation and Context Representation layers:

The external command information is passed through the same *Word Representation* and *Context Representation* layers i.e. these layers share weights for encoding a) the dialog context, b) the candidate response and c) the external knowledge. We also experimented with using separate weights for external knowledge (weight untying) i.e. a different BiLSTM was used to encode external information but did not observe increase in performance.

Attention Matching layer:

The Attention Matching layer is updated to incorporate the additional external information, such as man pages for Ubuntu dataset and course information for Advising dataset. In addition to computing co-attention between the dialog history (context) and the candidate response, we also compute attention between a) the dialog context and the external information and b) candidate response and the external information. Using the attention scores for similarity, we compute the following:

- attended context vectors (based on candidate response and the external information)
- attended candidate response vectors (based on context and the external information)
- attended external information vectors (based on context and candidate response)

These attended vectors are then used to compute vector difference and element-wise product, to enrich the interaction information between each pair from the set {dialog context, candidate response, external information}, similar to equations 5 and 6 as shown in Figure 1.

Matching Aggregation and Pooling layers:

The Matching Aggregation layer aggregates the encoded information from *Attention Matching* layer. We use the same BiLSTM for all pairs mentioned above (weight-tying). Since we have 2 representations for each variable {dialog context, candidate response and external information}, we perform an additive merge in the *Pooling layer* to get a final representation for each variable, which gets concatenated and used as input to the final *Prediction layer*.

Extracting relevant external information:

Ubuntu dataset: The external data is provided as man pages information for 1200 Linux commands. We map the command description for all commands into TF-IDF vectors. We use two hashtables: entity hashtable and relation hashtable for knowledge extraction (Lowe et al. 2015a). The command names are used as the keys to the entity hashtable, and their descriptions as the corresponding values. The words following the command name in the *Name* section[3] are used as keys for the relation hashtable. We also filter the keys of the relation hashtable by removing common English words. The values to the relation hashtable contains the commands from which the keys of the relation hashtable were extracted. Thus, the relation hashtable contains pointers to the keys in the entity hashtable.

We compare all the tokens in the context with keys in entity hashtable to find any direct matches for command names. We also check for partial matches if the token length is greater than 8 as sometimes the full command name may not be used in the context. If we don't get any matches in the entity hashtable, we look for token matches in the context with keys in the relation hashtable. These matches eventually lead us to the list of command names relevant to the given dialog. The shortlisted commands are ranked based on the TF-IDF match score for their corresponding description with the given dialog context. We further split the top-k[4] command descriptions into individual sentences. These sentences are

[3] The Name section has a one-sentence summary of the command

[4] We used k=5 in our experiments.

Model	Validation				Test			
	R@1	R@10	R@50	MRR	R@1	R@10	R@50	MRR
subtask-1								
ESIM	43.76	71.70	95.84	53.24	50.1	78.3	95.4	59.34
T-ESIM	52.62	76.46	96.08	60.57	61.9	82.2	96.6	69.09
T-ESIM-CR	54.46	79.26	97.92	62.73	63.4	84.2	98.5	70.69
T-ESIM-Sampled	53.16	78.5	96.46	61.54	62.8	83.4	96.6	69.7
T-ESIM-Sampled-CR	55.46	81.98	98.2	64.12	64.3	84.7	97.3	71.25
subtask-2								
ESIM	11.12	31.5	58.6	18.59	12.8	28.5	36.5	18.43
T-ESIM	18.72	35.9	61.26	25.13	21.6	36.0	44.1	26.68
subtask-4								
ESIM	40.16	76.18	96.36	53.43	43.5	82.1	96.2	57.96
T-ESIM	47.76	77.22	96.4	58.43	52.5	82.3	97.1	63.6
subtask-5								
K-ESIM	44.82	72.74	96.4	54.52	50.1	78.3	96.3	60.2
TK-ESIM	53.10	75.88	96.26	60.88	60.9	80.2	96.6	67.93
TK-ESIM-CR	54.84	79.26	97.96	62.98	62.3	83.4	97.8	69.56

Table 1: Performance of models on the Ubuntu validation and test datasets. R@k refers to Recall at position k in 100 candidates, denoted as R@1, R@10 and R@50. MRR refers to the Mean Reciprocal Rank.

Model	Validation				Test			
	R@1	R@10	R@50	MRR	R@1	R@10	R@50	MRR
ESIM (subtask-1)	17.2	47.6	88.8	27.5	14.8	46.2	86.6	25.43
ESIM (subtask-3)	10.2	47.6	87.6	22.08	18.6	60.2	92.6	31.62
ESIM (subtask-4)	22.2	57.2	91.8	33.89	17.0	72.8	91.2	30.14
K-ESIM (subtask-5)	16.4	50.4	85.6	27.45	11.6	49.2	88.2	23.02

Table 2: Performance of models on the Advising validation and test datasets. R@k refers to Recall at position k in 100 candidates, denoted as R@1, R@10 and R@50. MRR refers to the Mean Reciprocal Rank.

again ranked in descending order of their cosine similarity with respect to the context. Finally, first 200 words were selected from these ranked sentences[5].

Advising dataset: The external data is provided in a KB which contains information such as 'Course Name', 'Area', 'Credits', 'Workload', 'Class Size', etc. We construct a natural language sentence representation for the course information provided in the KB using a set of predefined rules. Each suggested course is mapped to its corresponding sentence representation. An example natural language sentence representation for a suggested course is: "`EECS281` is `Data Structures and Algorithms`, has `moderate` workload, `large` class size, `4` credits, has `a discussion`, the classes are `on Thursday, Tuesday afternoon`". This representation is used as external knowledge input to our proposed model K-ESIM for the given dialog.

Proposed model: T-ESIM

In a dialog corpus, similar conversations can appear many times. For example, in a customer care scenario, a common problem might arise for multiple users and many similar dialogs would be present in the corresponding dialog corpus. Pandey et al. (2018) proposed Exemplar Encoder-Decoder (EED) architecture that makes use of similar conversations for the generation-based dialog system and achieved better results than models such as HRED (Serban et al. 2016) and VHRED (Serban et al. 2017). We adpot a similar approach and propose a new training strategy to incorporate the information from similar dialogs present in the available training data for the next utterance selection task. We refer to the new training strategy as T-ESIM (Targeted ESIM), where additional information in terms of probable target responses is added to the contextual information.

For our T-ESIM implementation, we use text-based similarly technique to identify relevant dialogs, similar to K-ESIM. Each dialog in the training data is split at multiple points to create a larger pool of dialogs, which are called sub-dialogs. The sub-dialogs are then converted to TF-IDF vector representations. The current dialog is matched against these sub-dialogs (excluding its children) to identify similar sub-dialogs to select top-k similar sub-dialogs[6]. The corresponding response(s) for the top-k similar sub-dialogs are concatenated to the current dialog context as a new turn in

[5] We select the first 200 words as we use num_units=200 for the BiLSTM cells in our model.

[6] We use k=3 during training and k=1 during evaluation

the partial conversation[7]. The core motivation here is that the model can learn to use responses for similar dialogs present in the training data, to get improved performance on the next utterance selection task. We also explore additional training strategies: T-ESIM-Sampled and T-ESIM-CR, which are described below. We evaluate these strategies on the Ubuntu dialog corpus, as shown in Table 1.

T-ESIM-Sampled: When the candidate set for each dialog contains 100 utterances, 1 candidate is the correct utterance and the remaining 99 candidates are incorrect responses. To speed up the training, we randomly sample 9 utterances from the 99 incorrect utterances. We refer to this training strategy as *T-ESIM-Sampled*.

T-ESIM-CR: The Ubuntu and Advising corpus are constructed from real-world human-to-human conversations. This makes them unique, as different people can answer the same question in different ways. The different answers could theoretically have the same information but would differ in terms of natural language. Therefore, during evaluation, we employ the Candidate Reduction (CR) trick to use the presence of unique responses in the dataset. For evaluation on the validation set, we reduce the total number of candidates in the candidate set by removing the candidates which are present as correct responses in the training data and similarly, for test data, we remove the candidates which are present in the training and validation data.

Experiments and Results

Our results for the baseline model ESIM and our proposed models: K-ESIM and T-ESIM for the Ubuntu dataset are given in Table 1 and for the Advising dataset are given in Table 2. The models are evaluated on two metrics - Recall@k, which refers to recall at position k in the set of the 100 candidates and MRR (mean reciprocal rank).

We observe that the baseline ESIM model achieves 50.1 R@1 on the Ubuntu test set and 14.8 R@1 on the Advising test set for subtask 1. For Advising dataset subtask 5, we observe that the K-ESIM model performance is slightly below the baseline ESIM model. We believe that our external knowledge representation for the Advising dataset is not suited for the task. For the Ubuntu dataset, we also observe that our proposed models: K-ESIM and T-ESIM perform better than the baseline ESIM model. K-ESIM achieves 44.82 R@1 and 0.5452 MRR on Ubuntu subtask 5 validation set, compared to 43.76 R@1 and 0.5324 MRR for ESIM. K-ESIM also performs slightly better than ESIM on MRR on the test set. T-ESIM performs significantly better than the baseline ESIM model on all Ubuntu subtasks and achieves 61.9 R@1 on subtask 1. Our proposed techniques T-ESIM-Sampled and T-ESIM-CR perform well and achieve 64.3 R@1 score on the Ubuntu subtask 1. These results show that our proposed models and training strategies perform well.

For Ubuntu Subtask 2, the size of global pool of candidates is 120000. For training purposes, we reduce the candidate set by randomly sampling 99 incorrect responses from the global pool. These 99 responses, in addition to the correct response, construct our candidate set of 100 responses per dialog, similar to Subtask 1. During evaluation on the validation and test sets, we first employ the CR technique mentioned above. Then, we shortlist the number of candidates to 100, by selecting the top-100 candidates from the reduced candidate global pool using IR-based methods similar to knowledge extraction for K-ESIM and T-ESIM.

Conclusion and Future Work

In this paper, we introduced two knowledge incorporating end-to-end dialog systems for retrieval-based goal-oriented dialog, by extending the ESIM model. Evaluation based on the Ubuntu dataset show that our methods are effective to improve performance by incorporating additional external knowledge sources and leveraging information from similar dialogs. Although our proposed model K-ESIM shows improvement on the Ubuntu subtask 5, we observe a slight decrease in performance on the Advising subtask 5 as explained in the previous section.

In our future work, we plan to explore the following areas to improve our proposed K-ESIM and T-ESIM models: a) improve the knowledge representation for course information, b) investigate attention mechanisms over a KB (Eric and Manning 2017) and c) explore neural approaches, instead of TF-IDF, for extracting relevant external information (man pages) and identifying similar dialogs for T-ESIM.

Appendix: Model Training and Hyperparameter Details

In Word Representation Layer, we used 300-dimensional Glove pre-trained vectors[8] ((Pennington, Socher, and Manning 2014)), 100-dimensional word2vec vectors (Mikolov et al. 2013) and 80-dimensional character-composed embedding vectors for generating the representation of a word. For training word2vec vectors, we use the `gensim.models.Word2Vec` API with the following hyper-parameters: size=100, window=10, min_count=1 and epochs=20. The final prediction layer is a 2-layer fully-connected feed-forward neural network with ReLu activation. We use sigmoid function and minimize binary cross-entropy loss for training and updating the model.

The baseline model was implemented in Tensorflow (Abadi et al. 2016) and we used the source code released by Dong and Huang (2018)[9] for the baseline model. We generated word2vec word embeddings from scratch on the DSTC7 datasets as mentioned in Algorithm-1 from Dong and Huang (2018). We used Adam (Kingma and Ba 2014) with a learning rate of 0.001 and exponential decay with a decay rate of 0.96 decayed every 5000 steps. Batch size used was 128. The number of hidden units for BiLSTM in both the context representation layer and the matching aggregation layer was 200. For the prediction layers, we used 256 hidden units with ReLU activation.

[7] The training data is increased by a factor of k

[8] glove.42B.300d.zip : `https://nlp.stanford.edu/projects/glove/`

[9] source code released by Dong and Huang (2018): `https://github.com/jdongca2003/next_utterance_selection`

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Bartl, A., and Spanakis, G. 2017. A retrieval-based dialogue system utilizing utterance and context embeddings. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, 1120–1125. IEEE.

Bordes, A.; Boureau, Y.-L.; and Weston, J. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1657–1668.

Dong, J., and Huang, J. 2018. Enhance word representation for out-of-vocabulary on ubuntu dialogue corpus. *arXiv preprint arXiv:1802.02614*.

dos Santos, C.; Guimaraes, V.; Niterói, R.; and de Janeiro, R. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, 25.

Eric, M., and Manning, C. D. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Ghazvininejad, M.; Brockett, C.; Chang, M.-W.; Dolan, B.; Gao, J.; Yih, W.-t.; and Galley, M. 2017. A knowledge-grounded neural conversation model. *arXiv preprint arXiv:1702.01932*.

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

Kadlec, R.; Schmid, M.; and Kleindienst, J. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

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*.

Le, P.; Dymetman, M.; and Renders, J.-M. 2016. Lstm-based mixture-of-experts for knowledge-aware dialogues. *arXiv preprint arXiv:1605.01652*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 110–119.

Lowe, R.; Pow, N.; Serban, I.; Charlin, L.; and Pineau, J. 2015a. Incorporating unstructured textual knowledge sources into neural dialogue systems. In *Neural Information Processing Systems Workshop on Machine Learning for Spoken Language Understanding*.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015b. 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*.

Pandey, G.; Contractor, D.; Kumar, V.; and Joshi, S. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1329–1338.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Seo, M.; Min, S.; Farhadi, A.; and Hajishirzi, H. 2016. Query-reduction networks for question answering. *arXiv preprint arXiv:1606.04582*.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, 3776–3784.

Serban, I. V.; Sordoni, A.; Lowe, R.; Charlin, L.; Pineau, J.; Courville, A. C.; and Bengio, Y. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, 3295–3301.

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

Vinyals, O., and Le, Q. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

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*.

Young, T.; Cambria, E.; Chaturvedi, I.; Huang, M.; Zhou, H.; and Biswas, S. 2017. Augmenting end-to-end dialog systems with commonsense knowledge. *arXiv preprint arXiv:1709.05453*.