Multi-level Context Response Matching in Retrieval-Based Dialog Systems

Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin, Emmanuel Morin
LS2N, UMR CNRS 6004, Université de Nantes, France
firstname.lastname@ls2n.fr

Abstract

We present our work on the Dialog System Technology Challenges 7 (DSTC7). We participated in Track 1 on sentence selection which evaluates response retrieving in dialog systems on more realistic test scenarios compared to the state-of-the-art evaluations. Our proposed dialog system matches the context with the best response by computing their semantic similarity on the word and sequence levels. Evaluation results on the provided datasets show the effectiveness of our system by achieving higher performance compared to the provided baseline system. Our system enjoys the advantages of its simple and end-to-end architecture making its training and adaptation to other domains easier.

1 Introduction

The increasing interest in building goal-oriented dialog systems is a result of the high costs and the difficulty of having enough human assistants to book restaurants, hotels, solve problems, etc. of millions of users. Today, a large amount of human-human conversations are available thanks to social media, emails and community question-answering platforms (Song et al. 2018). Therefore, researchers are now able to build automated dialog systems that learn from human-human conversations in order to produce human-computer conversations with lower costs.

When a user asks a question, the dialog system either searches a correct response in a set of candidate responses (retrieval-based system) or generates a response word by word (generative system). In both cases, the retrieved or generated response should match the question and should be coherent with the conversation's history called *context*. Recent generative systems are based on the `seq2seq` model (Sutskever, Vinyals, and Le 2014). Despite the capacity of these systems (Vinyals and Le 2015; Serban et al. 2016; Sordoni et al. 2015) to generate customized responses for each context, they tend to generate short and general responses (Li et al. 2016; Shao et al. 2017). On the other hand, retrieval systems match the context with every candidate response based on their semantic similarity and pick the response that matches the best (Lowe et al. 2015; Wu et al. 2016; Xu et al. 2017; Baudiš et al. 2016; Wu et al. 2017; Zhou et al. 2018). Thus, they can produce syntactically correct and more specific responses but only if this response is available in the set of candidate responses.

The existing retrieval-based dialog systems are evaluated on non realistic scenarios. Usually, these systems select the correct response from a very small set of candidate responses of size 10 (Lowe et al. 2015; Wu et al. 2016; Baudiš et al. 2016; Wu et al. 2017; Zhou et al. 2018). However, when building goal-oriented dialog systems, the set of possible responses is usually very large. Moreover, the actual systems provide a response even if no correct response is available in the candidate set in addition to the fact that most of them hypothesize that only one response is correct. However, multiple candidate responses could be correct responses. Addressing these limitations was the goal of the 1^{st} track (sentence selection) of the 7^{th} edition of the DSTC challenges (Yoshino et al. 2018). This track aims to push the state-of-the-art goal-oriented dialog systems in more realistic evaluation scenarios.

In this paper, we propose an end-to-end multi-level dialog system. Our system matches the context with the candidate responses on the word and sequence levels. First, by encoding the context and the candidate response using a shared encoder, we obtain their sequence level representations as two separated vectors. Then, we multiply these two vectors in order to obtain their sequence level similarity. In parallel, we compute a world level similarity matrix as the product between the word embeddings of the context and the candidate response. We encode this matrix into a vector. Finally, we concatenate both word and sequence similarity vectors and we produce the final score that we use to rank the candidate responses regarding the given context. Through experiments carried out on the two datasets provided by the challenge organizers, we show that our model achieves 73.2% on Recall@10 and 55.1% on MRR outperforming the baseline system by 37% on the first dataset.

The remainder of this paper is organized as follows. In Section 3, we describe the challenge and the tasks to which we are participating. In Section 4, we describe our proposed system, the experimental setup and the system parameters. We discuss the results in Section 5. Finally, we conclude with some perspectives for future work in Section 6.

2 Related Work

Recently, many studies were interested in building neural retrieval-based dialogue systems. In the following, we provide an overview of these systems.

(Lowe et al. 2015) built an utterance ranking system based on a *dual encoder*. They first encode the context and the response separately into two fixed size vectors. Then, a ranking score is computed as a dot product between a learned parameter matrix and these two vectors. With this approach only similarity on sequence level is captured. An extension of this work was realized by (Kadlec, Schmid, and Kleindienst 2015) in which Bidirectional LSTMs (BiLSTMs) and an ensemble system were deployed. The ensemble system regroups 11 LSTMs, 7 Bi-LSTMs and 10 CNNs (Convolutional Neural Networks(LeCun et al. 1998)). They average predictions of these models in order to obtain a final prediction. The complexity of this ensemble architecture makes it less interesting.

(Baudiš et al. 2016) combined both Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). They used the RNN to firstly encode the input text long-term dependencies and model contextual representations of words. Then they applied CNN and max pooling to compute the response score. In addition to the complexity of their architecture, only sequence level information was used.

Inspired by the human brain, (Xu et al. 2017) incorporated domain knowledge into their system in order to improve context and response modeling. They introduced for the first time a new cell called *r-LSTM* which has an extra gate called *Recall Gate*. This cell helps in memorizing information about domain knowledge words in addition to encoding the context and the response with the same process as (Lowe et al. 2015). The domain knowledge words are obtained from a hand-made knowledge base which makes the approach not easily scalable from one domain to another.

(Wu et al. 2017) designed a system which considers the context utterances separately. From the candidate response and each utterance of the context, they computed word level and sequence level similarities. These similarities are encoded using a succession of convolution and pooling and then accumulated using Gated Recurrent Units (GRU).

Unlike (Lowe et al. 2015; Wu et al. 2017; Baudiš et al. 2016), (Wang et al. 2013) and (Wu et al. 2016) only considered the last dialogue turn of the context. (Wu et al. 2016) used the conversation topic as extra information to improve the quality of the selected response. The conversation topic words were extracted from both the last turn of the context and the candidate response using the state-of-the-art topic Twitter LDA model.The limitations of this approach are related to its dependency to the topic generator and to considering only the last dialogue turn of the context.

3 Task Description

DSTC7[1] is the 7^{th} edition of the Dialog System Technology Challenges. This edition contains three tracks: sentence selection, sentence generation and audio visual scene-aware dialog. The first track aims to retrieve the correct response for a given conversation's history called the context from a set of candidate responses. The goal of the sentence generation track is to generate conversational responses that go beyond chitchat, by injecting informational responses that are grounded in external knowledge. The last track, aims to understand the scenes of an input video in order to have conversations with users about the objects and events around the video. The common point between the three tracks is: the participating systems must be data-driven and end-to-end. In this work we focus on the *sentence selection* track. In the following, we describe the track and its related subtasks.

3.1 Sentence Selection Track

Until today, the recent studies evaluated retrieval-based dialog systems in non-realistic conditions. We can summarize the limitations of the state-of-the-art systems in the following four points.

- Most of the recent systems were challenged to retrieve the ground-truth response among a set of only 10 candidate responses randomly sampled which is far from approaching the reality (Lowe et al. 2015; Xu et al. 2017; Wu et al. 2016; 2017). In real configuration, the dialog system has a large base of responses usually collected from human conversations from which the system has to pick one or more responses.
- Recent works limit the number of correct responses of a given context to only one. Whereas, in most cases, multiple correct responses are possible.
- Even if no correct answer is included in the set of candidate responses, most of the systems are not able to know what is wrong and retrieve a response anyway. However, they should be able to not provide an answer in such situations and ask the help of humans for example.

The main aim of the first track of DSTC7 is to address these limitations and to push goal-oriented dialog systems to more realistic problems that every practical automated agent have to deal with. In this track, two dialogue datasets were provided: the Ubuntu Dialogue Corpus and the Advising Corpus. Five subtasks were proposed where each subtask concerns one or both datasets. In the following we describe the subtasks and the datasets.

Subtask 1 Given a context of a conversation and a set of 100 candidate responses, the task consists of selecting the correct response. On 100 candidate responses, only one is correct. This subtask is available on both datasets.

Subtask 2 This subtask challenges the logical capability of the dialog model by increasing the size of the candidate responses set. Hence, the task consists of selecting the correct response from a pool of 120,000 candidate responses which is 12,000 times the usual size of the candidate set. Only Ubuntu Dialogue Corpus is concerned with this task. The 120,000 candidate responses are shared across training, validation and test sets and also across samples.

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

	Subtask 1							Subtask 3				Subtask 4						
	Ubuntu Corpus			Advising Corpus				Advising Corpus				Ubuntu Corpus			Advising Corpus			
	Train	Dev	Test	Train	Dev	Test1	Test2	Train	Dev	Test1	Test2	Train	Dev	Test	Train	Dev	Test1	Test2
# dialogues	100K	5K	1K	100K	500	500	500	100K	500	500	500	100K	5K	1K	100K	500	500	500
# cand. R per C	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
# + responses	1	1	1	1	1	1	1	1-5	1-5	1-5	1-5	0-1	0-1	0-1	0-1	0-1	0-1	0-1
Min # turns per C	3	3	3	1	1	1	1	1	1	1	1	3	3	3	1	1	1	1
Max # turns per C	75	53	43	41	34	36	26	41	34	36	26	81	51	65	41	34	36	26
Avg. # turns per C	5.49	5.59	3.84	9.22	9.78	9.47	9.44	9.22	9.78	9.47	9.44	5.45	5.43	5.59	9.22	9.78	9.47	9.44
Avg. # tokens per C	74.03	72.47	81.32	79.88	83.86	87.37	82.22	79.88	83.86	87.37	82.22	73.24	72.90	72.73	79.88	83.86	87.37	82.22
Avg. # tokens per R	62.92	62.82	63.06	57.83	66.13	66.60	67.38	57.90	65.94	66.57	67.15	62.91	62.96	62.66	57.82	66.10	66.59	67.39

Table 1: Datasets statistics. *C*, *R* and *cand.* denote context, response and candidate respectively.

Subtask 3 In this subtask, between one and five correct responses are available in the set of candidate responses of size 100. This subtask is only available on Advising corpus. The set of correct responses if available are paraphrases of the original correct response and the number of paraphrases has been chosen randomly. The aim of this subtask is to evaluate the ability of the participating systems to retrieve all the correct responses (the correct response and its paraphrases) by ranking them on top of the candidate responses.

Subtask 4 The candidate set contains 100 responses that may not include the correct response. Here, retrieval systems must be able respond with a `None` response when no correct response is found. This subtask is applicable on both datasets.

Subtask 5 In this last subtask, external knowledge base is provided and the model should incorporate it to retrieve the only correct response in a set of 100 candidate responses. The knowledge bases are Ubuntu manual pages in the case of the Ubuntu Dialogue Corpus and course descriptions in the case of the Advising Corpus.

In this paper, we focus on three subtasks: 1, 3 and 4.

3.2 Datasets

DSTC7 provided two new goal-oriented dialog datasets in order to build and evaluate retrieval-based dialog systems. Each dataset is splitted into training, validation and testing sets. Table 1 summarizes statistics of both datasets for each subtask. Note that Subtask 2 concerns only the Ubuntu Dialogue Corpus, the Subtask 3 concerns only the Advising Corpus.

The Ubuntu Dialogue Corpus This corpus contains two-party dialogues extracted from the Ubuntu channel on the Freenode Internet Relay Chat (IRC) (Kummerfeld et al. 2018). The corpus contains Ubuntu-related conversations. Every sample of this corpus is composed of a context which is a set of successive dialogue turns and a response which is the next turn of the same conversation. Moreover a set of randomly crawled candidate responses is provided. The task consists of ranking the correct response on top of the candidate responses.

Advising Corpus The advising corpus contains conversations between teacher and student in which the teacher tries to answer the student's questions about the courses he/she will take. The teacher aims to provide information related to the duration of the course, its difficulty, whether the student's profile is adapted to the course, etc.

3.3 Evaluation Metrics

For all the subtasks, DSTC7 uses Recall@1, Recall@10, Recall@50, and Mean Recall Rank (MRR) as evaluation metrics. Only for subtask 3, Mean Average Precision (MAP) is used in addition to the previous metrics.

4 Proposed System

Inspired by the previous works of (Lowe et al. 2015) and (Wu et al. 2017), we propose an end-to-end multi-level context response matching dialog system for the task of sentence selection. Our system enjoys the advantages of the efficiency of the dual encoder proposed by (Lowe et al. 2015) in encoding the context and the candidate response. In addition to that, we incorporate word level similarity proposed in the work of (Wu et al. 2017) into the dual encoder in order to help the system in learning learning a rich similarity between the context and the candidate responses.

First, we project the context and the candidate response into a distributed representation (word embeddings). Second, we encode the context and the candidate response into two fixed-size vectors using a shared recurrent neural network. Then, in parallel, we compute two similarity vectors: on the word and sequence levels. The sequence level similarity vector is obtained by multiplying the context and the response vectors. Whereas the word level similarity vector is obtained by multiplying word embeddings of the context and the candidate response. Both vectors are concatenated and transformed into a probability of the candidate response being the next dialogue turn of the given context. In the following, we elaborate on the functions of our system.

4.1 Approach

Sequence Encoding The first layer of our system maps each word of the input into a distributed representation $\mathbb{R}^d$ by looking up a shared embedding matrix $E \in \mathbb{R}^{|V| \times d}$ where V is the vocabulary and d is the dimension of word embeddings. We initialize the embedding matrix E using pretrained vectors and fine-tune them during training. E is a parameter of our model to be learned by propagation. This layer produces matrices $C = [e_{c1}, e_{c2}, ..., e_{cn}]$ and $R = [e_{r1}, e_{r2}, ..., e_{rn}]$ where $e_{ci}, e_{ri} \in \mathbb{R}^d$ are the embeddings of the i-th word of the context and the response respectively and n is a fixed sequence length. Context and

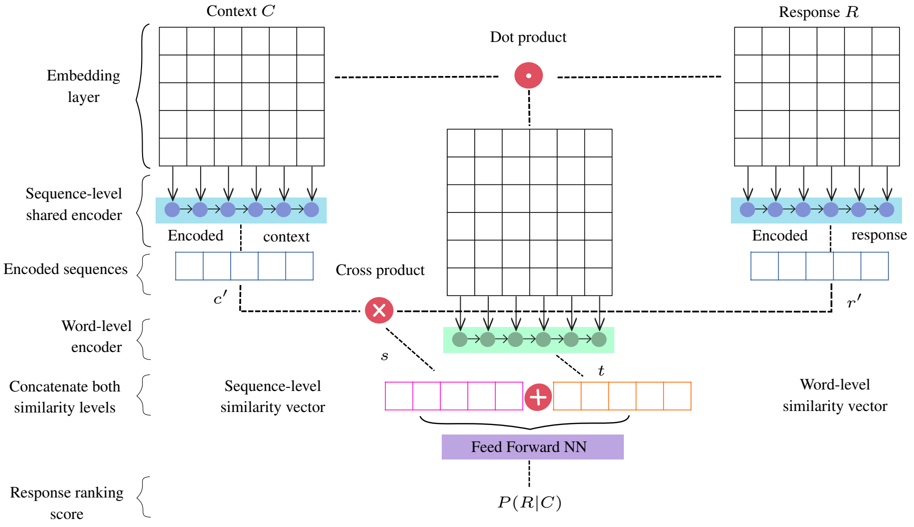

Figure 1: Architecture of our multi-level context response matching dialog system.

response matrices $C, R \in \mathbb{R}^{d \times n}$ are then fed into a shared LSTM (Hochreiter and Schmidhuber 1997) network word by word in order to get encoded.

Let c' and r' be the encoded vectors of C and R. They are the last hidden vectors of the encoder such as $c' = h_{c,n}$ and $r' = h_{r,n}$ where $h_{c,i}, h_{r,i} \in \mathbb{R}^m$ and m is the dimension of the hidden layer of the LSTM recurrent network. $h_{c,i}$ is obtained by Equation 1. $h_{r,i}$ is obtained similarly by replacing e_{ci} by e_{ri}.

$$\begin{aligned} z_i &= \sigma(W_z \cdot [h_{c,i-1}, e_{ci}]) \\ r_i &= \sigma(W_r \cdot [h_{c,i-1}, e_{ci}]) \\ \widetilde{h}_{c,i} &= \tanh(W \cdot [r_i * h_{c,i-1}, e_{ci}]) \\ h_{c,i} &= (1 - z_i) * h_{i-1} + z_i * \widetilde{h}_{c,i} \end{aligned} \tag{1}$$

W_z, W_r and W are parameters, z_i and r_i are an update gate and $h_{c,0} = 0$.

Sequence Level Similarity We hypothesize that positive responses are semantically similar to the context. Thus, the aim of a response retrieval system is to rank the most semantically similar response to the context on top of the candidate responses. Once the input vectors are encoded, we compute a cross product s between c' and r' as follows:

$$s = c' \otimes r' \tag{2}$$

$\otimes$ denotes the cross product. As a result, $s \in \mathbb{R}^m$ models the similarity between C and R on the sequence level.

Word Level Similarity We believe that sequence level similarity is not enough to match the context with the best response. Adding word level similarity could help the system learning an improved relationship between C and R. This assumption was consolidated by observing the scores dropping when word level similarity was removed from the system of (Wu et al. 2017) (see section *"Model ablation"* in their paper).

Therefore, we compute a word level similarity matrix $WLSM \in \mathbb{R}^{n \times n}$ by multiplying every word embedding of the context e_{ci} by every word embedding of the response e_{rj} as:

$$WLSM_{i,j} = e_{ci} \cdot e_{rj} \tag{3}$$

Where $\cdot$ denotes the dot product. In order to transform the word level similarity matrix into a vector, we feed every row $WLSM_i$ into an LSTM recurrent network which learns a representation of the chronological dependency and the semantic similarity between the context and response words. Similarly to Equation 1, we encode the word level similarity matrix into a vector $t = h'_n \in \mathbb{R}^l$ where l is the dimension of the hidden layer of the LSTM network and h'_n is the last hidden vector of the network.

Response Score At this stage, we have two vectors: s representing the similarity between C and R on the sequence level and t representing their similarity on the word level. We concatenate both vectors and transform the resulting vector into a probability using a one-layer fully-connected feed-forward neural network with sigmoid activation (Equation 4). The last layer predicts the probability $P(R|C)$ of the response R being the next utterance of the context C.

$$P(R|C) = sigmoid(W' \cdot (s \oplus t) + b) \tag{4}$$

Where W' and b are parameters and $\oplus$ denotes concatenation. We train our model to minimize the binary cross-entropy loss.

As stated at the beginning of this Section, our system is inspired by the dual encoder (Lowe et al. 2015) and the Sequential Matching Network (SMN) (Wu et al. 2017). We brought some modifications on the dual encoder as follows. First, we used a shared encoder to project the context and the response into the same space instead of using two separated encoders as in the original work. Second, in order to

System	Subtask	Measure	Ubuntu Dialogue Corpus	Advising Corpus case 1	Advising Corpus case 2
Baseline	Subtask 1	Recall@1	0.083	0.008	0.008
		Recall@10	0.359	0.102	0.094
		Recall@50	0.794	0.542	0.498
		MRR	0.175	0.053	0.048
Our system	Subtask 1	Recall@1	**0.446**	**0.114**	**0.1**
		Recall@10	**0.732**	**0.398**	**0.42**
		Recall@50	**0.937**	**0.782**	**0.802**
		MRR	**0.551**	**0.205**	**0.200**
	Subtask 3	Recall@1	-	**0.212**	**0.176**
		Recall@10	-	**0.586**	**0.57**
		Recall@50	-	**0.906**	**0.926**
		MRR	-	**0.338**	**0.297**
		MAP	-	**0.37**	**0.343**
	Subtask 4	Recall@1	**0.388**	**0.088**	**0.066**
		Recall@10	**0.592**	**0.31**	**0.316**
		Recall@50	**0.751**	**0.618**	**0.686**
		MRR	**0.462**	**0.163**	**0.15**

Table 2: Experimental results on test sets of Subtasks 1, 3 and 4.

compute the sequence similarity between the encoded vectors produced by the encoders, we simply compute a cross product instead of using a bilinear model that requires learning an additional matrix of parameters noted as M in (Lowe et al. 2015).

The idea of adding word level similarity in our system was consolidated by seeing the performance of the SMN dropping when the word level similarity matrix was removed in the work of (Wu et al. 2017). Hence, we computed and used this similarity matrix with a slight difference compared to the original one. First, we compute one similarity matrix between the candidate response and all the context instead of computing n similarity matrix between the candidate response and each of the n dialogue turns of the context. Second, we encode this matrix using an LSTM network in order to produce one vector representing the similarity on the word level, whereas in the SMN, a CNN network was used in order to encode each matrix into multiple vectors aggregated later using a GRU network. We made these choices for a sake of simplicity and efficiency.

4.2 System Extension

We used the same system described above with the same parameters in the three subtasks to which we participated with/without extension depending on the subtask. In subtask 1, we used the system described in Figure 1. In subtask 3, we hypothesize that if our system is able to match the context with the correct response, it should be able to match its paraphrases with the same context as well. Thus, we used the same system as subtask 1. In subtask 4, our system should be able to recognize cases where no correct response is available in the set of candidate responses. Therefore, we extended the same system used in subtasks 1 and 3 with an SVM classifier (Ben-Hur et al. 2001) with RBF kernel as described in figure 2. For every candidate response and a context, our response retrieval system (described in Section 4) provides a ranking score. Once we have the ranking scores of all the candidate responses, we feed them to the SVM classifier. We train this classifier to predict the presence of a correct response among the candidate responses using the labeled training data.

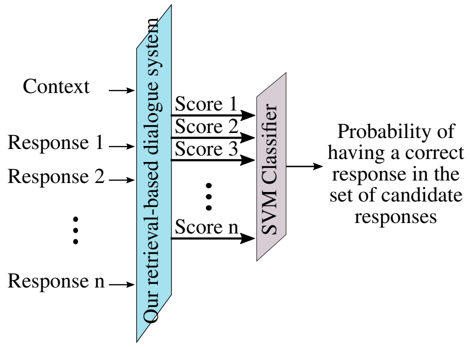
Context
Response 1
Response 2
Response n
Our retrieval-based dialogue system
Score 1
Score 2
Score 3
Score n
SVM Classifier
Probability of having a correct response in the set of candidate responses

Figure 2: Extension of our proposed system for subtask 4.

4.3 System Parameters

The only pre-processing performed on the dataset is tokenization using Keras Tokenizer. The system parameters were updated using Stochastic Gradient Descent with Adam algorithm (Kingma and Ba 2015). The initial learning rate was set to 0.001 and Adam's parameters β_1 and β_2 were set to 0.9 and 0.999 respectively. As a regularization strategy we used *early-stopping* and to train the model we used mini batch of size 256. The size of word embeddings[2] and the size of the hidden layer of the encoder LSTM were set to 300. Whereas the size of the hidden layer of the second LSTM that learns the WLSM matrix was set to 200. All the hyperparameters were obtained with a grid search on the validation set. We implemented our system with Keras (Chollet and others 2015) with Theano (Theano Development Team 2016) in backend that we trained on a single Titan X GPU. We used the SVM implementation provided by Scikit-learn (Pedregosa et al. 2011) with the default parameters. We made publicly available the source code that reproduces our results on `https://github.com/basma-b/multi_level_chatbot`.

[2] We trained word embeddings on the training sets using fastText (Bojanowski et al. 2017) `-ws 5 -minCount 1 -dim 100` (Advising) `-dim 300` (Ubuntu)

5 Results and Discussion

The baseline system is an extension of the `dual encoder` of (Lowe et al. 2015). The differences between our system and the baseline system are the following. (1) our system learns to match the context and the candidate response on the word and sequence levels whereas the baseline system is based on only the sequence similarity. (2) We use a shared encoder to encode the context and the candidate response while the baseline system uses different encoders. This allows the encoded context and the encoded response to be presented in the same vector space. (3) Unlike the baseline system, at each time step of the training, our system matches the context with one candidate response and thus the encoder is alternating the context and the response which is coherent to the chronological order of dialogue turns in the context and the response.

We used the scripts[3] provided by the organizers to evaluate the baseline system on the test set[4]. We also report the results of our system produced by the task organizers. Table 2 summarizes these results on the three subtasks. Note that two test sets were provided for the Advising Corpus noted as `case 1` and `case 2`. As we can see, our system outperforms the provided baseline system on all the metrics with a good margin. These results confirm the effectiveness of matching the context and the response on the word and the sequence levels and using a shared encoder instead of different encoders for the context and the response. Also, we observe that the performance of our system in addition to the baseline system on the Advising Corpus are lower than the performance on the Ubuntu Dialogue Corpus.

	Train	Dev	Test	
			Case 1	Case 2
Ubuntu	20%	20%	20.20%	-
Advising	20.05%	18.80%	23.40%	18.40%

Table 3: Percentage of cases where no correct response is provided in the candidate set (Subtask 4).

The performance of our system on Subtask 3 are higher than Subtask 1 on all the metrics. In theory, retrieving multiple correct responses (Subtask 3) is more difficult than retrieving only one correct response (Subtask 1). However, the results in Table 2 do not follow the same logic even if the two compared systems were trained with the same hyper-parameters. We hypothesize that this is due to the fact that the set of correct responses contains the correct response and its paraphrases which are semantically similar to the correct response and are easily retrieved.

The results of subtask 4 are quite lower than expected. We analyzed the subtask datasets and found that the SVM classifier is hard to train because of the unbalanced data. As mentioned in Table 3, the percentages of training samples where no correct response is available in the candidate set are 20% and 20.05% in the case of Ubuntu and Advising datasets respectively. At the training step, the system will see 80% of dialogues with a correct response and thus will tend to generalize and predict a correct response most of the time. Applying some data balancing techniques may solve this problem (we will investigate this part in future work).

System Ablation As mentioned in previous sections, we incorporated a modified word level similarity (Wu et al. 2017) to a slightly improved dual encoder (Lowe et al. 2015). In order to evaluate the impact of these modifications, we performed an ablation study in which we kept only sequence level similarity. Table 4 summarizes the results of this study on the `validation` sets of Subtask 1. Based on these results, we can deduce two points. (1) The modification of the dual encoder with only sequence similarity results in better performance compared to the baseline (the original dual encoder). (2) Having word similarity in addition to sequence similarity can help the system to perform a better matching between the context and the correct responses.

			Ubuntu	**Advising**
	Baseline	R@1	0.083	0.062
		R@10	0.359	0.296
		R@50	0.800	0.728
		MRR	-	-
Our system	**Only sequence similarity**	R@1	0.290	0.080
		R@10	0.575	0.364
		R@50	0.910	0.800
		MRR	0.389	0.176
	Word + sequence similarities	R@1	**0.399**	**0.116**
		R@10	**0.693**	**0.444**
		R@50	**0.944**	**0.848**
		MRR	**0.501**	**0.219**

Table 4: Ablation results on the validation data of Subtask 1.

6 Conclusion

In this paper we proposed an end-to-end retrieval-based dialog system that learns to match the context with the correct response. We evaluated our system on the track of sentence selection of the DSTC7 challenge. Experimental results have shown the effectiveness of combining sequence and word level similarities by bringing significant improvements compared to the baseline system. The DSTC7 challenge provided an excellent evaluation environment for retrieval-based dialog systems and has successfully pushed them towards dealing with more realistic constraints. As future work, we plan to qualitatively evaluate the errors by doing a error analysis in order to improve the system performance. We plan to incorporate attention mechanism (Graves 2013; Bahdanau, Cho, and Bengio 2014) in order to help the system accentuate the more important words.

7 Acknowledgment

We thank the anonymous reviewers for their valuable comments. This work was partially supported by the project ANR 2016 PASTEL[5].

[3] `https://github.com/IBM/dstc7-noesis/tree/master/noesis-tf`

[4] We used the hyper-parameters defined by the organizers.

[5] `http://www.agence-nationale-recherche.fr/?Projet=ANR-16-CE33-0007`

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baudiš, P.; Pichl, J.; Vyskočil, T.; and Šedivỳ, J. 2016. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.

Ben-Hur, A.; Horn, D.; Siegelmann, H. T.; and Vapnik, V. 2001. Support vector clustering. *Journal of machine learning research* 2(Dec):125–137.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics (TACL)* 5:135–146.

Chollet, F., et al. 2015. Keras. `https://github.com/keras-team/keras`.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

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. In *Workshop on Machine Learning for Spoken Language Understanding and Interaction at the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*.

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*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

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 (NAACL'16)*, 110–119.

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. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'15)*, 285–294.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 3776–3783.

Shao, Y.; Gouws, S.; Britz, D.; Goldie, A.; Strope, B.; and Kurzweil, R. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, 2210–2219.

Song, Y.; Li, C.-T.; Nie, J.-Y.; Zhang, M.; Zhao, D.; and Yan, R. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, (IJCAI'18)*, 4382–4388.

Sordoni, A.; Bengio, Y.; Vahabi, H.; Lioma, C.; Grue Simonsen, J.; and Nie, J.-Y. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*, 553–562.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 2014 conference on Advances in Neural Information Processing Systems (NIPS'14)*. 3104–3112.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688.

Vinyals, O., and Le, Q. 2015. A neural conversational model. In *Workshop on Deep Learning at the 31 st International Conference on Machine Learning (ICML'15)*.

Wang, H.; Lu, Z.; Li, H.; and Chen, E. 2013. A dataset for research on short-text conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, 935–945.

Wu, Y.; Wu, W.; Li, Z.; and Zhou, M. 2016. Response selection with topic clues for retrieval-based chatbots. *arXiv preprint arXiv:1605.00090*.

Wu, Y.; Wu, W.; Xing, C.; Zhou, M.; and Li, Z. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, 496–505.

Xu, Z.; Liu, B.; Wang, B.; Sun, C.; and Wang, X. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'17)*, 3506–3513.

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*.

Zhou, X.; Li, L.; Dong, D.; Liu, Y.; Chen, Y.; Zhao, W. X.; 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 (ACL'18))*, 1118–1127.