End-to-End Response Selection Based on Multi-LevelContext Response Matching

Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin, Emmanuel Morin

End-to-End Response Selection Based on Multi-Level Context Response Matching

Basma El Amel Boussaha*, Nicolas Hernandez, Christine Jacquin, Emmanuel Morin

LS2N, UMR CNRS 6004, Université de Nantes, France

Abstract

This paper presents our work on the Dialog System Technology Challenges 7 (DSTC7). We took part 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 word and sequence levels. Evaluation results on the datasets provided 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.

Keywords: Retrieval systems, chatbots, neural networks, goal-oriented dialogue systems, DSTC

1. Introduction

The increasing interest in building goal-oriented dialog systems is a result of the high costs and the difficulty of employing enough human assistants to book restaurants, hotels, solve technical problems, etc. for 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

*Corresponding author
Email address: basma.boussahaa@gmail.com (Basma El Amel Boussaha)

order to produce human-computer conversations with lower costs.

When a user asks a question, the dialog system either looks for 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 et al., 2014). Despite the capacity of these systems (Vinyals & 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 such as *"Ok"* and *"Thank you"* (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 they are limited by the list of candidate responses which may not contain correct responses or may contain multiple correct responses.

Retrieval-based dialogue systems have gained interest as they have proved their efficiency in both academia and industry such as the Alibaba's chatbot *AliMe* (Qiu et al., 2017) and the Microsoft's social-bot *XiaoIce*[1] (Shum et al., 2018). However, in academia, the existing retrieval-based dialog systems are evaluated on scenarios that do not reflect the reality. Usually, these systems select the correct response from a very small set of around 10 candidate responses (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, most of these systems 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 towards more realistic evaluation scenarios. It consists of

[1] `https://www.msxiaoice.com/`

5 subtasks, each of which challenges the participating systems in a different way.

In this paper, we describe our end-to-end single-turn multi-level dialog system which we proposed for the 1^{st} track of DSTC7. Our system matches the context with the candidate responses on word and sequence levels (multi-level) and does not consider the context turns separately but matches the candidate response with the whole context (single-turn). 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 given the context of the conversation. The experiments carried out on two datasets provided by the challenge organizers 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 (Boussaha et al., 2019a). Also, we evaluate our system on benchmark datasets such as the Ubuntu Dialogue Corpus (Lowe et al., 2015) and the Douban Conversation Corpus (Wu et al., 2017) that are widely used in evaluating retrieval based dialogue systems and show that our simple and efficient system can outperform several complex systems while being conceptually simpler (Boussaha et al., 2019b).

The remainder of this paper is organized as follows: we review the most recent works on retrieval-based dialogue systems in Section 2. 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. The results are discussed in Section 5. We compare our system to other state-of-the-art systems on two widely used datasets in Section 8. Finally, in Section 9, we conclude with perspectives for future work.

2. Related Work

Recently, many studies have concentrated on building neural retrieval-based dialogue systems. This category of dialogue systems has proved its efficiency in both academia and industry (Qiu et al., 2017; Shum et al., 2018). Depending on the way retrieval-based dialogue systems match the context with the candidate responses, we distinguish two main categories that we describe below:

2.1. Single-Turn Matching Models

The first fully end-to-end single-turn dialogue system was the dual encoder (Lowe et al., 2015). First, the context and the candidate response are represented using word embeddings and are fed into an LSTM (Hochreiter & Schmidhuber, 1997) network. As its name indicates, the model is composed of two encoders which fulfill the same function of the encoder in the encoder-decoder model. They encode the context and the response into two fixed-size vectors that are multiplied with a matrix of learned parameters to produce the matching score of the response. Some variants of the dual encoder based on Convolutional Neural Networks (CNNs) (LeCun et al., 1998) and Bidirectional LSTMs were also explored by Kadlec et al. (2015).

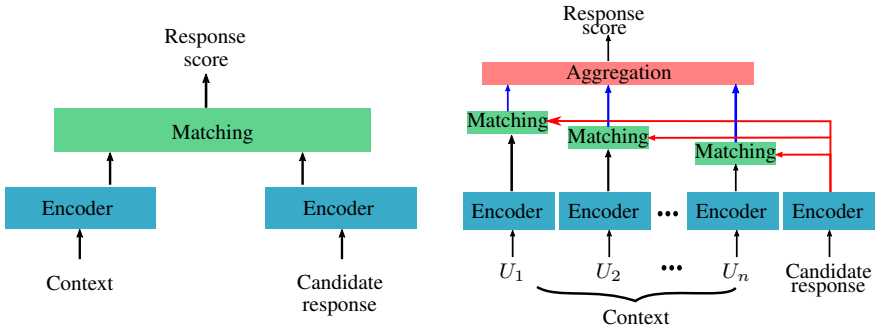
Tan et al. (2015) built a similar framework called Attentive-LSTM for question answering. They used a Bidirectional LSTM (BiLSTM) network to encode the question and the answer combined with attention mechanism (Bahdanau et al., 2014). The attention mechanism allows the dual encoder to alleviate the problem of the LSTM when encoding long sequences which is the case of the context. The attention weights allow the model to give more weights to certain words and thus a better matching between the question and the correct answer can be achieved. (Wan et al., 2016) proposed another semantic matching framework based on a quite similar approach called MV-LSTM. It allows matching two texts based on the positional sentence representations. A BiLSTM is used as an encoder which contains positional information of the inputs at each of its hidden states. By matching the positional information of the inputs by cosine, bilinear or tensor layer, an interaction score is obtained. Finally, and by aggregating all the interaction scores through k-Max pooling and a multi-layer perceptron, the final matching score of the context and the response is obtained.

In the same line of reasoning, Wang & Jiang (2016) proposed Match-LSTM for the natural language inference task in which the problem consists of determining whether we can derive a hypothesis H from a premise sentence P. Here, the premise and the hypothesis could be the context and the response. The difference with the dual encoder (Lowe et al., 2015) is that, while sequentially processing the hypothesis word by word, each word is matched with an attention-weighted representation of the premise. This results in a cross-attention matching of the context and the response. More recently, Chen & Wang (2019a,b) proposed an Enhanced Sequential Inference Model (ESIM) which was originally developed for natural language inference (Chen et al., 2018). They start by encoding the context and the response with a BiLSTM encoder following the same process as Lowe et al. (2015). Then, cross attention mechanism is applied to model the semantic relation between the context and the response. Afterwards, max and mean pooling are applied and the output is transformed into a probability that the response is the next utterance of the given context using a multi-layer perceptron classifier.

On the contrary of the previous systems which consider only the last utterance of the context and can match only short sequences, these single-turn matching systems can match longer sequences. By concatenating the context utterances as a single long utterance, they match the context with the response only one time which makes them quick and robust. Also, they enjoy the advantages of their simple architectures which are based on the dual encoder most of the time.

2.2. Multi-Turn Matching Models

As illustrated in Figure 1b, multi-turn matching systems match the candidate response with every utterance of the context. Then, an aggregation mechanism is applied to combine the different matching scores and produce a response score. Yan et al. (2016) proposed a Deep Learning to Respond (DL2R) framework for open-domain dialogue systems. Their system is based on contextually query reformulation in which the last utterance of the context (called query) is concatenated with the previous utterances to formulate multiple reformulated queries. These reformulated queries, the original query, the response, and its antecedent post are encoded via a BiLSTM en-



(a) Single-turn matching models

(b) Multi-turn matching models

Figure 1: General architectures of single- and multi-turn matching models

coder followed by a convolution and a max-pooling layers. Then, the encoded features are matched with each other and fed into a Feed-Forward Neural Network (FFNN) to compute the final matching score of the response and the context. Zhou et al. (2016) exploited for the first time the word level similarity between the context and the response in their Multi-view response retrieval system. The particularity of this model is that two similarity levels between the candidate response and the context are computed and the model is trained to minimize two losses. The word and sequence level similarities are computed by matching the word embeddings and the sequence vectors. The disagreement loss and the likelihood loss are computed between the prediction of the system and what the system was supposed to predict.

Later on, Wu et al. (2017) further improved the leveraging of utterances relationship and contextual information through the Sequential Matching Network (SMN). They not only match the context utterances with the response one by one but also they are matched on multiple levels of similarity. They start by encoding separately the last 10 utterances of the context in addition to the response with a shared Gated Recurrent Unit (GRU) (Chung et al., 2014) encoder. The hidden states of the GRU form a matrix that represents the sequential information of each input. Moreover, a word similarity matrix is computed as a dot product between the matrices of each utterance and the response. These two matrices are used as input channels of a Convolutional Neural Network (CNN) followed by a max-pooling that computes a two-level matching vec-

tors between the response and each context turn. A second GRU network aggregates the obtained vectors and produces a response ranking score. This sequential matching network constitutes a solid base for later works.

More recently, Zhou et al. (2018) extended the SMN (Wu et al., 2017) through the the Deep Attention Matching Network (DAM). The DAM addresses the limitations of recurrent neural networks in capturing long-term and multi-grained semantic representations. This model is based entirely on the attention mechanism (Bahdanau et al., 2014). It is inspired by the Transformer (Vaswani et al., 2017) to rank the response using self- and cross-attention. The first GRU encoder of the SMN model is replaced by five hierarchically stacked layers of self-attention. Five matrices of multi-grained representations of the context turns and the response are obtained instead of one matrix in the case of SMN. Following the same process as the SMN, the response matrices are matched with the context turns matrices and stacked together in the form of a 3D image (matrix). This image contains self- and cross-attention information of the inputs. Later, a succession of convolution and max-pooling are applied to the image to produce the response score.

Afterward, Yang et al. (2018) proposed the Deep Matching Network (DMN) to extend the SMN[2] furthermore. The extension consists of the inclusion of external knowledge in two different ways. The first approach is based on the Pseudo-Relevance Feedback (Cao et al., 2008) named DMN-PRF and consists of extending the candidate response with relevant words extracted from the external knowledge (Question Answering (QA) data). The second approach incorporates external knowledge with QA correspondence Knowledge Distillation named DMN-KD. It adds a third input channel to the CNN of the SMN as a matrix of the Positive Pointwise Mutual Information (PPMI) between words of the response and the most relevant responses retrieved from the external knowledge. The Deep Utterance Aggregation (DUA) system (Zhang et al., 2018) also extends the SMN with an explicit weighting of the context utterances. The authors hypothesize that the last utterance of the context is the most relevant and thus, they concatenate its encoded representation with all the previous utterances in addi-

[2]Sequential Matching Network (SMN) is called Deep Matching Network (DMN) in their paper.

tion to the candidate response. After that, a gated self-matching attention (Wang et al., 2017) is applied to remove redundant information from the obtained representation before feeding them into the CNN as in the SMN (Wu et al., 2017).

These multi-turn matching dialogue systems assume that the response replies to each of the utterances of the context. Compared to single-turn matching systems, they deploy complex mechanisms of matching and aggregation which slower the training process as more parameters need to be optimized and may constrain the adaptability of the model to multiple domains. We believe that every researcher has to ask himself/herself a question before extending an existing approach or building a new system: "*What is the cost of this architecture in terms of training resources and duration? Are we able to achieve this performance while using fewer layers and fewer parameters?*". Because today, we believe that researchers tend to combine different neural networks with attention and other enhancement tools without caring about the generated costs. Strubell et al. (2019) in her recent work, quantified the computational and environmental cost of training deep neural network models for NLP, and showed that the authors should report training time and computational resources required in addition to the performance of their systems. This will allow a fair comparison of different approaches as well as a preference for systems with fewer parameters and that require fewer resources for ecological reasons. Mainly, for all these reasons, we opted for single-turn matching systems as the architecture of our proposed systems.

3. Task Description

DSTC7[3] is the 7^{th} edition of the Dialog System Technology Challenges. This edition consists of three tracks: sentence selection, sentence generation and audiovisual 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.

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

The last track, aims to understand the scenes of an input video in order to have conversations with users about the objects and events related to the video. The common point between the three tracks is that the participating systems must be data-driven and end-to-end. In this work we focus on the *sentence selection* track. In the following section, we describe the track and its related subtasks.

3.1. Sentence Selection Track

Until today, 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 three following points:

- Most of the recent systems were challenged to retrieve the ground-truth response in a set of only 10 randomly sampled candidate responses 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 Lowe et al. (2015); Xu et al. (2017); Wu et al. (2017); Zhou et al. (2018); Yang et al. (2018). 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 Lowe et al. (2015); Xu et al. (2017); Wu et al. (2017); Zhou et al. (2018); Yang et al. (2018). 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 towards more realistic problems that every practical automated agent has to deal with. In this track, two dialogue datasets were provided: the Ubuntu Dialogue Corpus and the Advising Corpus. Five subtasks were proposed

Subtask	Description	Ubuntu	Advising
1	Select one response from a pool of 100 candidate responses	✓	✓
2	Select one response from a pool of 120000 candidate responses	✓	✗
3	Select a response and its paraphrases from a pool of 100 candidate responses	✗	✓
4	Select one response from a pool of 100 candidate responses that may not contain the response	✓	✓
5	Same as 1 but the usage of a provided external data is mandatory	✓	✓

Table 1: Subtasks of the sentence selection task of DSTC7

where each subtask concerns one or both datasets. In Table 1, we summarize the subtasks of the sentence selection task and the datasets on which they are applied and they are described as follows:

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.

Subtask 3 In this subtask, between one and five correct responses are available in the set of 100 candidate responses. This subtask is only available on the 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 to respond with a `None` response when no correct response is found. This subtask is applicable on both datasets.

Subtask 5 In this last subtask, an 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 four subtasks: 1, 3, 4 and 5.

3.2. Datasets

DSTC7 provided two new goal-oriented dialog datasets in order to build and evaluate retrieval-based dialog systems. Each dataset is split into training, validation and testing sets. Table 2 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, 2019). 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 teacher-student conversations collected at the University of Michigan with students playing teacher and student roles with simulated personas (Yoshino et al., 2018; Kummerfeld et al., 2019). The conversations in this corpus are about the courses that each student wants to take in the next semester based on a provided list of the courses that the student has already taken and a list of suggested courses. The dataset includes additional information about student preferences and course information. A total of 815 conversations were collected and used to generate 100,000 conversations for the training and 500 for each of the validation and test sets. Similar to the Ubuntu

Dialogue Corpus, the negative responses were randomly sampled from a set of 82,094 paraphrases of messages generated from the corpus.

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

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. The ranking of participating systems was performed by considering the average of the Recall@10 and MRR metrics. The challenge allows the submission of a maximum of 3 systems per team.

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 system for the task of sentence selection. Our system enjoys the advantages of the efficiency and the simplicity 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 a more complete similarity between the context and the candidate responses. Compared to other state-of-the-art systems, our system has less parameters, requires less resources (GPU) to be trained, and converges quickly after few epochs. Figure 2 illustrates the architecture of our proposed system.

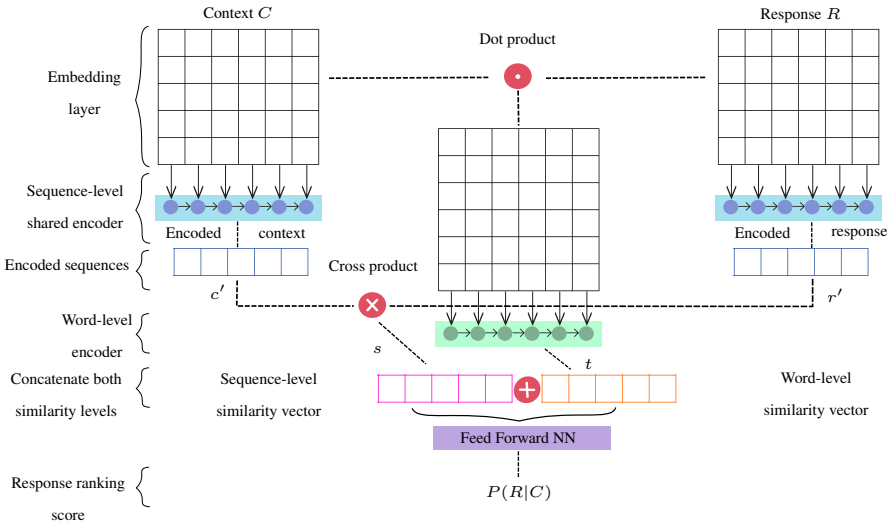


Figure 2: Architecture of our multi-level context response matching dialog system.

Firstly, we project the context and each of the candidate responses into a distributed representation (word embeddings). Secondly, 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 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 section, we elaborate on the functions of our system.

4.1. Approach

4.1.1. 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 response matrices $C, R \in \mathbb{R}^{d \times n}$ are then fed into a shared LSTM (Hochreiter & 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 $h_{c,0} = 0$.

4.1.2. 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:

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

where $\otimes$ denotes the cross product.

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

4.1.3. 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 to learn 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:

$$\text{WLSM}_{i,j} = e_{ci} \cdot e_{rj} \tag{3}$$

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.

4.1.4. 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 of all, 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. Secondly, in order to 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 the whole context instead of computing n similarity matrix between the candidate response and each of the n dialogue turns of the context. Then, 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 the sake of simplicity and efficiency.

4.2. System Extension

We used the same system described above with the same parameters in the four subtasks in which we took part with/without extension depending on the subtask. In subtask 1, we used the system described in Figure 2. 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 and it will try to rank the correct response and its paraphrases on top of other candidate responses. Only the metrics change for this subtask by introducing the Mean Average Precision (MAP) metric.

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 3. 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 clas-

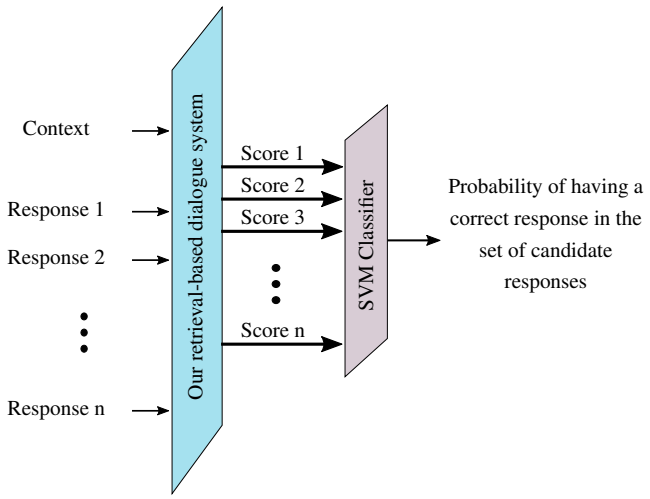


Figure 3: Extension of our proposed system for subtask 4.

sifier. We train this classifier to predict the presence of a correct response among the candidate responses using the labeled training data.

Subtask 5 requires participants to include the external knowledge into their system, while maintaining the end-to-end property of their architectures. Man pages and course descriptions were provided as external data for the Ubuntu Dialogue Corpus and the Advising Corpus respectively. We extracted plain text from these external data and we trained word embeddings on them using fastText. These word embeddings were used later to initialize the embeddings layer in our system.

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 & 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[4] and the size of the hidden layer of the

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

encoder LSTM were set to 300. Whereas the size of the hidden layer of the second LSTM that encodes the WLSM matrix was set to 200. All the hyper-parameters were obtained with a grid search on the validation set. Our system was implemented with Keras (Chollet et al., 2015) and 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 the source code that reproduces our results publicly available on `https://github.com/basma-b/multi_level_chatbot`.

5. Results and Discussion

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	**0.469**	**0.326**	**0.338**
		Recall@10	**0.756**	**0.668**	**0.646**
		Recall@50	**0.947**	**0.922**	**0.932**
		MRR	**0.573**	**0.449**	**0.440**
	Subtask 3	Recall@1		**0.212**	**0.176**
		Recall@10		**0.586**	**0.57**
		Recall@50	NA	**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**
	Subtask 5	Recall@1	**0.451**	**0.282**	**0.301**
		Recall@10	**0.742**	**0.558**	**0.593**
		Recall@50	**0.926**	**0.876**	**0.902**
		MRR	**0.550**	**0.379**	**0.393**

Table 3: Experimental results on test sets of Subtasks 1, 3, 4 and 5.

The baseline system provided by the challenge organizers is an implementation of the dual encoder of Lowe et al. (2015). We recall the differences between our system

and the baseline system in the following. (1) Our system learns to match the context and the candidate response on the word-level and sequence-level 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[5] provided by the organizers to evaluate the baseline system on the test set[6]. We also report the results of our system produced by the task organizers. Table 3 summarizes these results on the four 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. 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 (V3).

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

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

The performance of our system on Subtask 3 are lower than Subtask 1 on all the metrics. We believe that this result is logic as the system is challenged to retrieve all the correct responses which is harder than retrieving only one correct response (as in Subtask 1). 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

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

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

unbalanced data. As mentioned in Table 4, 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.

After incorporation of the external knowledge as required by Subtask 5, the performance of our system did not improve. The results of Subtask 1 and 5 are comparable as they use the same datasets. As we can see in Table 3, the results of Subtask 5 are lower than Subtask 1 on both datasets. We believe that this is mainly due to the new word embeddings that we computed on the external data. When we used the word embeddings produced from the training data as in Subtask 1, we were able to find 89,284 and 4,534 word embedding vectors for the training data of the Ubuntu Dialogue Corpus (V3) and the Advising Corpus respectively. However, when we use the word embeddings produced from the external data, only 23,910 and 2,350 word vectors were found. This explains the drop in the system performance as more words (whose word vectors were not found) will have randomly initialized embedding vectors.

A list of scores of the 20 participating teams to the *sentence selection* track is given in Table 5. As we can see in Figure 4, the systems of teams 2, 3 and 4 ranked at the first positions are based on the ESIM framework (Chen et al., 2018) basically proposed for Natural Language Inference. Almost all the first systems are based on self and cross-attention mechanisms and use data augmentation during training to increase the number of positive samples as we discuss in Section 6. Systems like those of teams 17, 18 and 13 stack many neural network systems or use ensemble systems which results in more complex architectures. Compared to these systems, our system is simpler and at the time of the submission, no data augmentation technique was used. We show later, that when we augment the training set with more positive samples to balance the ratio of positive and negative samples, our system would be ranked 5^{th} on the Ubuntu corpus.

	Ubuntu, Subtask				Advising, Subtask			
Team	1	2	4	5	1	3	4	5
3	**0.819**	0.145	**0.842**	**0.822**	**0.485**	**0.592**	**0.537**	**0.485**
4	0.772	-	-	-	0.451	-	-	-
17	0.705	-	-	0.722	0.434	-	-	0.461
13	0.729	-	0.736	0.635	0.458	0.461	0.474	0.390
2	0.672	0.033	0.713	0.672	0.430	0.540	0.479	0.430
10	0.651	**0.307**	0.696	0.693	0.361	0.434	0.262	0.361
18	0.690	0.000	0.721	0.710	0.287	0.380	0.398	0.326
8	0.641	-	0.527	0.646	0.310	0.433	0.233	0.301
16	0.629	0.000	0.683	-	0.280	-	0.370	-
15	0.473	-	-	0.478	0.300	-	-	0.236
7	0.525	-	0.411	-	-	-	-	-
11	-	-	-	-	0.075	0.232	?	-
12	0.077	-	0.000	0.077	0.075	0.232	0.000	0.075
1	0.580	-	-	-	0.239	-	-	-
6	-	-	-	-	0.245	-	-	-
9	0.482	-	-	-	-	-	-	-
14	0.008	-	0.072	-	-	-	-	-
19	0.265	-	-	-	0.180	-	-	-
5	0.076	-	-	-	-	-	-	-
20	0.002	-	-	-	0.004	-	-	-

Table 5: Track 1 results, ordered by the average rank of each team across the subtasks they participated in. The top result in each column is in bold. For these results the metric is the average of MRR and Recall@10 (Gunasekara et al., 2019). We participated as team number 8.

5.1. System ablation

As mentioned in previous sections, we incorporated word-level similarity to a slightly improved dual encoder (Lowe et al., 2015). To evaluate the impact of these modifications, we performed an ablation study in which we kept only sequence-level similarity. Table 6 summarizes the results of this study on the validation sets of Subtask 1. As we can see, the best results were achieved by having both similarity levels which validates our hypothesis that the correct responses are those that match with the context on the sequence-level and word-level. Moreover, when considering both similarities separately, we notice that matching the context and the candidate response

on the word-level is better than matching them on only the sequence-level. These results mean that explicitly considering the words separately are more meaningful than considering them implicitly when encoding the sequence and provide a fine-grained representation of the context and the response. 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. These results correlate perfectly with our previous experiments on the UDC (V1) and the Douban Conversation Corpus.

			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	0.206	0.084
		R@10	0.567	0.404
		R@50	0.885	0.791
		MRR	0.350	0.186
	Only word similarity	R@1	0.41	0.104
		R@10	0.697	0.418
		R@50	0.936	0.804
		MRR	0.512	0.209
	Word + sequence similarities	R@1	**0.463**	**0.116**
		R@10	**0.753**	**0.444**
		R@50	**0.945**	**0.848**
		MRR	**0.57**	**0.219**

Table 6: Ablation results on the validation data of Subtask 1.

6. Data Augmentation

The training set of Subtask 1 as illustrated in Table 2 is unbalanced. For each training sample, we have one positive response and 99 negative responses. Thus 99% of the training samples are negative while only 1% are positive. As we define the problem of response retrieval in dialogue systems as a classification problem, this unbalanced data will alter the training process. More specifically, our system will "see" more negative samples than positive ones and thus, it will tend to predict label 0 for most of the input samples. In the literature, different approaches and tricks of data balancing exist (He & Ma, 2013). We adopt a data augmentation approach to solve this problem and also to increase the number of training samples.

Each of the training samples is composed of a context, a response, and a label. The context is composed of multiple turns $t_1, t_2, \ldots, t_n$. Hence, we construct new positive samples starting from the second turn by concatenating the previous turns t_i and considering them as a new context and the turn t_j as the response with a label 1. By applying this data augmentation approach to the datasets of Subtask 1, we were able to obtain 10,349,002 training samples for the Ubuntu Dialogue Corpus (V3) which represents an increase of 3.49% of the total number of samples. Even if the training data remains unbalanced, we show in Table 7 that with this small increase in the number of positive samples, the performance of our system increased considerably.

	Ubuntu Dialogue Corpus (V3)			
	R@1	R@10	R@50	MRR
Our system	0.469	0.756	0.947	0.573
Our system + data augmentation	0.526	0.786	0.959	0.619

Table 7: Results of our system after application of data-augmentation on the training set of Subtask 1.

With this simple approach of data augmentation and while keeping the same parameters of our system, we were able to improve Recall@(1, 10, 50) and MRR by 8%, 3%, 1%, and 4% respectively on the Ubuntu Dialogue Corpus (V3). This result sheds the light on the importance of having a balanced training set which helps the system to perform a better learning.

Data augmentation has been used as a solution to the data insufficiency problem in multiple domains such as computer vision (Krizhevsky et al., 2012), speech recognition (Hannun et al., 2014), question answering (Fader et al., 2013), and text classification (Zhang et al., 2015). Few researchers applied data augmentation techniques on dialogue systems. For instance, Kurata et al. (2016) introduced an LSTM encoder-decoder with random noise to generate more training data for the slot filling task. Hou et al. (2018) combined sequence-to-sequence and diversity rank to generate more diverse utterances in the training data for the task of Dialogue Language Understanding (DLU). A more recent work combines Conditional Variational Autoencoder (CVAE) and Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to generate more diverse query-response pairs Li et al. (2019). These techniques are more complex than the data augmentation technique that we used which does not require training deep neural networks which requires itself large amount of training data. However, given the promising results that we obtained after augmenting the training data with a simple method, we can think of more elaborated techniques to achieve better performance.

7. Error Analysis

In order to understand the reasons for the failure of our system to retrieve the correct response, we need to analyze the errors. Therefore, we performed an error analysis on the predictions of our system on the test set of Subtask 1 of the Ubuntu Dialogue Corpus. On 1,000 test samples, our system failed to retrieve the ground-truth response in 531 cases. This represents 53.1% of wrong predictions based on R@1. In order to analyze these results, we performed a human evaluation of 100 (about 19%) randomly sampled wrong predictions out of 531. By observing the test samples that were misclassified, we identified 4 error classes (table 8 contains an example of each class).

(a) **Functionally equivalent**: this class regroups 30% of the samples where our system predicted a response that we believe it could replace (substitute) the ground-truth response without having the same meaning. For example both *"install build essential"* and *"gcc"* are possible responses to the context given in Table 8a without being semantically equivalent.

Context	Candidate responses
hello does anyone know what the package with the c dependancies is called *eot* do you mean a compiler *eot* yeah the gcc c compiler	- install build essential ✓ - **gcc✗**

(a) Functionally equivalent

Context	Candidate responses
hi whats a good data modelling tool for ubuntu *eot* blender type sudo apt get install blender *eot* i thinks it's a 3d modelling tool *eot* yes you mean a structured systems analysis tool or a simulator *eot* i was looking a data modeling tool databases	- ah ok ✓ - **ah ✗**

(b) Semantically equivalent

Context	Candidate responses
when i have a deb file in my case i got the skype deb package what is the command to install it *eot* dpkg i file *eot* thank you	- you need sudo if you are not root however use the graphical package manager don't download crappy packages god knows where ✓ - **normally you would file a bug report for the package that was broken and then upload a suggested solution there which package did your solution involve ✗**

(c) Out of context

Context	Candidate responses
do u have v4 enabled *eot* it's enabled by default no *eot* should be but if you are getting an unable to connect that may be my first thought	- browsing and pinging is working fine ✓ - **sorry though i can't help you with your problem ✗**

(d) Very general response

Table 8: Examples of errors raised by our system are grouped by error classes. The first response in the candidate response column is the ground-truth response whereas the second response (in bold) is the response predicted by our system. "*eot*" denotes the end of turn.

(b) **Semantically equivalent**: in this class we find 4% of the samples where the predicted response has a similar meaning as the ground-truth response. *"ah ok"* and *"ok"* are semantically similar.

(c) **Out of context**: This is the largest class which regroups 56% of the samples where our system predicted a response that is not general (i.e it is a technical response) which is neither functionally nor semantically related to the ground-truth response. The example in Table 8 illustrates an out of context response.

(d) **Very general responses**: in this class we found 10% of the samples where the ground-truth response was very specific to the context whereas our system predicted a thanking, greeting, apologizing, feedback informing, etc. responses. Examples include *"Thank you"*, *"Great"*, *"Ok"* and *"Yes"* or even the example given in Table 8.

We can group the findings of our error analysis into two important points:

1. Through this in-depth analysis we observed more than 60% of errors are due to general and completely out of context (classes c and d) responses which are highly ranked by our retrieval system. This means that our system has some difficulties in finding the correct responses in some cases. A more elaborated study has to be conducted to analyze these cases one by one by considering the context of the conversation and the set of candidate responses in addition to the ground-truth response. Moreover, these limitations were originally observed in generative systems since they were not able to produce coherent, syntactically correct and specific responses to the context of the conversation (Li et al., 2016). This finding encourages us to perform in-depth comparative studies between retrieval-based and generative dialogue systems on the automatic assistance task.

2. We highlight the importance of performing human evaluation on the candidate responses in the validation and the test sets. They should be carefully selected instead of randomly sampled from the corpus. 34% of the errors were due to the presence of responses which were functionally and semantically equivalent

to the ground-truth response that were considered as negative responses in the corpus (classes a and b).

8. Evaluation on Other Benchmark Datasets

To measure the efficiency of our approach and to compare it to other state-of-the-art systems, we evaluated our system on the widely used datasets in building and evaluating retrieval-based dialogue systems such as the Ubuntu Dialogue Corpus (Lowe et al., 2015) and the Douban Conversations Corpus (Wu et al., 2017). In Table 9, statistics of both datasets are given. The choice of these baseline systems is motivated by their novelty, the availability of their performance on one or both datasets, the availability of their source code to reproduce their results and the fact that these systems were considered as baselines in most of the recent works which helped in comparing our system to previous systems. In Table 10, we summarize the evaluation results of our system and the following baseline systems:

	UDC (V1)			Douban		
	Train	Valid	Test	Train	Valid	Test
# dialogues	1M	500,000	500,000	1M	50,000	10,000
# cand. R per C	2	10	10	2	2	10
Min # turns per C	1	2	1	3	3	3
Max # turns per C	19	19	19	98	91	45
Avg. # turns per C	10.13	10.11	10.11	6.69	6.75	6.47
Avg. # tokens per C	115.0	114.6	115.0	109.8	110.6	117.0
Avg. # tokens per R	21.86	21.89	21.94	13.37	13.35	16.29

Table 9: Statistics on the datasets. *C*, *R* and *cand.* denote context, response and candidate respectively (Boussaha et al., 2019b).

TF-IDF We report results of the Term Frequency-Inverse Document Frequency (TF-IDF) model (Lowe et al., 2017). The context and each of the candidate responses are represented as vectors of TF-IDF of their words. Then, a cosine similarity is computed between the context and the response vectors and used as a ranking score of the response.

LSTM dual encoder The model was introduced in the work of (Lowe et al., 2017). The context and the response were presented using their word embeddings and then they were fed separately word by word into two LSTM networks which encode them into fixed size vectors. Then a response ranking score was computed using a bi-linear model (Tenenbaum & Freeman, 2000).

BiLSTM dual encoder The system of Kadlec et al. (2015) in which the LSTM cells were replaced by bidirectional LSTM cells. We do not report the results of their ensemble system which regroups 11 LSTMs, 7 Bi-LSTMs and 10 CNNs because we believe that it is important to build simple systems.

Attentive LSTM Tan et al. (2015) built a similar framework as the dual encoder for question answering. They used the Bidirectional LSTM (BiLSTM) network to encode the question and the answer combined with attention mechanism (Bahdanau et al., 2014). As the input sequences may be long, the hidden vectors of the BiLSTM become bottleneck. The attention mechanism is deployed to alleviate this issue by computing a dynamic weighting of the word vectors at the output of the BiLSTM.

MV-LSTM (Wan et al., 2016) is a semantic matching framework that allows matching two texts based on the positional sentence representations. First, two BiLSTM networks are used to encode the two input texts eg. the context and the response. At each time step, the hidden vector of the BiLSTM contains the positional information of the input. By matching the positional information of the inputs by cosine, bi-linear or tensor layer, an interaction score is obtained. By aggregating all the interaction scores through k-Max pooling and a multi-layer perceptron, the final matching score of the inputs is obtained.

Match-LSTM (Wang & Jiang, 2016) was proposed for a natural language inference task in which the problem consists of determining whether we can derive a hypothesis H from a premise sentence P. Their system is similar to the dual encoder with the premise and the hypothesis as inputs. The difference is that, while sequentially processing the hypothesis word by word, each word is matched with

an attention-weighted representation of the premise.

Deep Learning to Respond (DL2R) Proposed by Yan et al. (2016) based on contextual query reformulation and an aggregation of three similarity scores computed on the sequence level. The reformulated query is matched with the response, the original query and the previous post.

Multi-View This system was designed by Zhou et al. (2016) in which two similarity levels between the candidate response and the context are computed and the model is trained to minimize two losses: the disagreement loss and the likelihood loss between the prediction of the system and what the system was supposed to predict.

Sequential Matching Network (SMN) Proposed by Wu et al. (2017). The candidate response and every dialogue turn of the context are encoded using a GRU network (Chung et al., 2014). Then, the response is matched with every turn using a succession of convolutions and max-pooling.

Deep Attention Matching Network (DAM) Introduced in the work of (Zhou et al., 2018). This system is an improvement of the SMN (Wu et al., 2017) in which the Transformer (Vaswani et al., 2017) was used in order to produce utterance representations based on self-attention. These representations are matched to produce self- and cross-attention scores which are stacked as a 3D matching image. A ranking score is then produced from this image via convolution and max pooling operations.

Deep Utterance Attention (DUA) This system (Zhang et al., 2018) also extends the SMN with an explicit weighting of the context utterances. The authors hypothesize that the last utterance of the context is the most relevant and thus concatenate its encoded representation with all the previous utterances in addition to the candidate response. After that, a gated self-matching attention (Wang et al., 2017) is applied to remove redundant information from the obtained representation before feeding them into the CNN as in the SMN.

System	Ubuntu Dialogue Corpus V1				Douban Conversation Corpus					
	R_2@1	R_{10}@1	R_{10}@2	R_{10}@5	R_{10}@1	R_{10}@2	R_{10}@5	P@1	MAP	MRR
TF-IDF (Lowe et al., 2017)	0.659	0.410	0.545	0.708	0.096	0.172	0.405	0.180	0.331	0.359
LSTM (Lowe et al., 2017)	0.901	0.638	0.784	0.949	0.187	0.343	0.720	0.320	0.485	0.527
BiLSTM (Kadlec et al., 2015)	0.895	0.630	0.780	0.944	0.184	0.330	0.716	0.313	0.479	0.514
Attentive-LSTM (Tan et al., 2015)	0.903	0.633	0.789	0.942	0.192	0.328	0.718	0.331	0.495	0.523
MV-LSTM (Wan et al., 2016)	0.906	0.653	0.804	0.946	0.202	0.351	0.710	0.348	0.498	0.538
Match-LSTM (Wang & Jiang, 2016)	0.904	0.653	0.799	0.944	0.202	0.348	0.720	0.345	0.500	0.537
DL2R (Yan et al., 2016)	0.899	0.626	0.783	0.944	0.193	0.342	0.705	0.330	0.488	0.527
Multi-View (Zhou et al., 2016)	0.908	0.662	0.801	0.951	0.202	0.350	0.729	0.342	0.505	0.543
$SMN_{dynamic}$ (Wu et al., 2017)	0.926	0.726	0.847	0.961	0.233	0.396	0.724	0.397	0.529	0.569
DAM (Zhou et al., 2018)	0.938	0.767	0.874	0.969	0.254	0.410	0.757	0.427	0.550	0.601
DUA (Zhang et al., 2018)	-	0.752	0.868	0.962	0.243	0.421	0.780	0.421	0.551	0.599
Our system	0.935	0.763	0.870	0.968	0.255	0.414	0.758	0.418	0.548	0.594
Only sequence similarity	0.917	0.685	0.825	0.957	0.209	0.357	0.702	0.358	0.500	0.543
Only word similarity	0.926	0.744	0.853	0.956	0.223	0.370	0.719	0.373	0.513	0.556

Table 10: Evaluation results on the UDC V1 and Douban Corpus using retrieval metrics.

As we can see in Table 10, compared to the single-turn systems (the first six rows), our system achieves the best results on all metrics and on both datasets. The first four systems are based on only sequence level similarity between the context and the candidate response whereas our system incorporates word level similarity in addition to the sequence similarity. Moreover, our system outperforms the $SMN_{dynamic}$ (Wu et al., 2017) with a good margin (around 4% and 3% on Recall@1 and 2 respectively on UDC). Even if the SMN matches the response with every context turn and uses multiple convolutions and max pooling to rank the response, its performance is lower than our system's performance. We believe that using our architecture, we were able to efficiently capture both similarity levels without any need to make the system more complicated by matching the response with every utterance of the context.

Our system neither matches each context turn with the candidate response nor uses complex cross and self attention in addition to matching and accumulation mechanisms but achieves almost the same performance as the Deep Attention Matching (DAM) (Zhou et al., 2018) and the Deep Utterance Aggregation system (Zhang et al., 2018) on both datasets and on all metrics. The DAM system is based on multiple layers of the self attention (Transformer) and Convolutional Neural Networks (LeCun et al., 1998). Even if the advantages of the Transformer are related to the performance improvement

and the acceleration of the learning compared to recurrent neural networks (Vaswani et al., 2017) but the cost of its training is very high (Strubell et al., 2019). We proposed an architecture that is fully based on recurrent neural networks but that achieves almost the same results as the DAM and sometimes better. The advantages of our system compared to the DAM is in contrast to what was said before, our system converges quickly. According to the authors (Zhou et al., 2018), their system was trained on one Nvidia Tesla P40 GPU, on which one epoch lasts for 8 hours on UDC and their system converges after 3 epochs. However, training our system for one epoch lasts for 50 minutes on one Nvidia Titan X pascal GPU (Both GPUs have almost the same characteristics[7]) and our system converges after only two epochs[8]. Having such architectures (as DAM) makes reproduciblity of results more difficult due to the hardware limitations in addition to the time necessary to perform training and cross-validation.

Note that on Douban, the overall performance of all the systems are lower than on UDC. This is due to the nature of the Douban corpus in which a context may have more than one ground-truth response and hence every retrieval system must find all the responses.

9. Conclusion

This paper describes our 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 and two widely used datasets in building and evaluating retrieval-based dialogue systems. Experimental results have shown the effectiveness of combining sequence and word level similarities by bringing significant improvements compared to the baseline systems. Throughout our work, we show the importance of having simple architectures that can work as well as complex architecture and sometimes better. Having a simple architecture, facilitates its adaptation and reuse in other domains and applications easy and costs less energy and pollutes less the environment. The DSTC7 challenge provided an excellent evaluation environment

[7] https://technical.city/en/video/Titan-X-Pascal-vs-Tesla-P40

[8] The number of trainable parameters of our system and DAM is almost the same

for retrieval-based dialog systems and has successfully pushed them towards dealing with more realistic constraints. Unbalanced data was one of the challenges of the task and we show that using a simple approach of data balancing through data augmentation can help the system achieve a better performance. The advising dataset offers metadata in addition to the conversations between the teacher and the student including course description and student preference. It would be interesting to efficiently include them in our system as external knowledge to enrich the information that we extract from the conversation.

References

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

Baudiš, P., Pichl, J., Vyskočil, T., & 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., & Vapnik, V. (2001). Support vector clustering. *Journal of machine learning research*, *2*, 125–137.

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

Boussaha, B. E. A., Hernandez, N., Jacquin, C., & Morin, E. (2019a). Multi-level context response matching in retrieval-based dialog systems. In *Proceedings of the 7th edition of the Dialog System Technology Challenges Workshop at AAAI (DSTC7'19)*. Honolulu, HI, USA.

Boussaha, B. E. A., Hernandez, N., Jacquin, C., & Morin, E. (2019b). Towards simple but efficient next utterance ranking. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'19)*. La Rochelle, France.

Cao, G., Nie, J.-Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)* (pp. 243–250). New York, NY, USA.

Chen, Q., & Wang, W. (2019a). Sequential attention-based network for noetic end-to-end response selection. In *Proceedings of the 7 th Dialog System Technology Challenge (DSTC7)*. Honolulu, HI, USA.

Chen, Q., & Wang, W. (2019b). Sequential matching model for end-to-end multi-turn response selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)* (pp. 7350–7354). Brighton, UK.

Chen, Q., Zhu, X., Ling, Z.-H., Inkpen, D., & Wei, S. (2018). Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)* (pp. 2406–2417). Melbourne, Australia.

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

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Workshop on Deep Learning and Representation Learning at the 28th Annual conference on Advances in Neural Information Processing Systems (NIPS'14)*. Montreal, Canada.

Fader, A., Zettlemoyer, L., & Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)* (pp. 1608–1618). Sofia, Bulgaria.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. N. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems 27* (pp. 2672–2680). Montreal, Canada.

Gunasekara, C., Kummerfeld, J., Polymenakos, L., & Lasecki, W. S. (2019). Dstc7 task 1: Noetic end-to-end response selection. In *Proceedings of the 7th Dialog System*

Technology Challenges DSTC7 at the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19). Honolulu, HI, USA.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, .

He, H., & Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. (1st ed.). Wiley-IEEE Press.

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

Hou, Y., Liu, Y., Che, W., & Liu, T. (2018). Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)* (pp. 1234–1245). Santa Fe, NM, USA.

Kadlec, R., Schmid, M., & 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)*. Montreal, Canada.

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

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS'12)* (pp. 1097–1105). Lake Tahoe, USA.

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

Kummerfeld, J. K., Gouravajhala, S. R., Peper, J., Athreya, V., Gunasekara, C., Ganhotra, J., Patel, S. S., Polymenakos, L., & Lasecki, W. S. (2019). A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Kurata, G., Xiang, B., & Zhou, B. (2016). Labeled data generation with encoder-decoder lstm for semantic slot filling. In *Proceedings of The 17th Annual Conference of the International Speech Communication Association (Interspeech'16)* (pp. 725–729). San Francisco, CA, USA.

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

Li, J., Galley, M., Brockett, C., Gao, J., & 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)* (pp. 110–119). San Diego, CA, USA.

Li, J., Qiu, L., Tang, B., Chen, D., Zhao, D., & Yan, R. (2019). Insufficient data can also rock! learning to converse using smaller data with augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)* (pp. 6698–6705). Honolulu, HI, USA.

Lowe, R., Pow, N., Serban, I., & 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)* (pp. 285–294). Prague, Czech Republic.

Lowe, R. T., Pow, N., Serban, I. V., Charlin, L., Liu, C.-W., & Pineau, J. (2017). Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, *8*, 31–65.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Courna-

peau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Qiu, M., Li, F.-L., Wang, S., Gao, X., Chen, Y., Zhao, W., Chen, H., Huang, J., & Chu, W. (2017). Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)* (pp. 498–503). Vancouver, Canada.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & 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)* (pp. 3776–3783). Phoenix, AZ, USA.

Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., & 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)* (pp. 2210–2219). Copenhagen, Denmark.

Shum, H.-y., He, X.-d., & Li, D. (2018). From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, *19*, 10–26.

Song, Y., Li, C.-T., Nie, J.-Y., Zhang, M., Zhao, D., & 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)* (pp. 4382–4388).

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., & 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)* (pp. 553–562). Melbourne, Australia.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)* (pp. 3645–3650). Florence, Italy.

Sutskever, I., Vinyals, O., & 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)* (pp. 3104–3112). Montreal, Canada.

Tan, M., Santos, C. d., Xiang, B., & Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. In *Proceedings of the International Conference on Learning Representation (ICLR'15)*. San Juan, Puerto Rico.

Tenenbaum, J. B., & Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, *12*, 1247–1283.

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'17)* (pp. 5998–6008). Long Beach, CA, USA.

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

Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., & Cheng, X. (2016). A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)* (pp. 2835–2841). Phoenix, AZ, USA.

Wang, S., & Jiang, J. (2016). Learning natural language inference with lstm. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'16)* (pp. 1442–1451). San Diego, CA, USA.

Wang, W., Yang, N., Wei, F., Chang, B., & Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)* (pp. 189–198). Vancouver, Canada.

Wu, Y., Wu, W., Li, Z., & 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., & 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)* (pp. 496–505). Vancouver, Canada.

Xu, Z., Liu, B., Wang, B., Sun, C., & 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)* (pp. 3506–3513). Anchorage, AK, USA.

Yan, R., Song, Y., & Wu, H. (2016). Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)* (pp. 55–64).

Yang, L., Qiu, M., Qu, C., Guo, J., Zhang, Y., Croft, W. B., Huang, J., & Chen, H. (2018). Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)* (pp. 245–254). New York, NY, USA.

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., & Batra, D. (2018). The 7th dialog system technology challenge. *arXiv preprint*, .

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15)* (pp. 649–657). Montreal, Canada.

Zhang, Z., Li, J., Zhu, P., Zhao, H., & Liu, G. (2018). Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Confer-*

ence on Computational Linguistics (COLING'18) (pp. 3740–3752). Santa Fe, NM, USA.

Zhou, X., Dong, D., Wu, H., Zhao, S., Yu, D., Tian, H., Liu, X., & Yan, R. (2016). Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)* (pp. 372–381).

Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W. X., Yu, D., & 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))* (pp. 1118–1127).

Team	Model Type	External Data Use	Used Raw Advising	Train Data	Model Details
1	CNN	-	No	Train+Val	Combination of CNN for utterance representation and GRU for modeling the dialogue.
2	LSTM	-	Yes	Train	ESIM with an aggregation scheme that captures the dialog-specific aspects of the data + ELMo.
3	LSTM	Embeddings	Yes	Train	ESIM plus a filtering stage for subtask 2.
4	LSTM	-	No	Train	ESIM with (1) enhanced word embeddings to address OOV issues, (2) an attentive hierarchical recurrent encoder, and (3) an additional layer before the softmax.
6	Ensemble	-	No	Train	An ensemble of CNNs.
7	LSTM	-	No	Train+Val	LSTM representation of utterances followed by a convolutional layer.
8	Other	-	Yes	Train	A multi-level retrieval-based approach that aggregates similarity measures between the context and the candidate response on the sequence and word levels.
10	LSTM	TF-IDF Extraction	No	Train	ESIM with matching against similar dialogues in training, and an extra filtering step for subtask 2.
12	RNN	TF-IDF Extraction	No	Train	BoW over ELMo with context as an RNN.
13	Ensemble	Embeddings	No	Train	Ensemble approach, combining a Dynamic-Pooling LSTM, a Recurrent Transformer and a Hierarchical LSTM.
14	Ensemble	-	No	Train	An ensemble using voting, combining the baseline LSTM, a GRU variant, Doc2Vec, TF-IDF, and LSI.
15	Memory	Memory	No	Train	Memory network with an LSTM cell.
16	LSTM	-	No	Train	ESIM with utterance-level attention, plus additional features.
17	Memory	Memory & Embeddings	Yes	Train	Self-attentive memory network, with external advising data in memory and external ubuntu data for embedding training.
18	GRU	-	No	Train	Stacked Bi-GRU network with attention, aggreagting attention across the temporal dimension followed by a CNN and softmax.
19	LSTM	-	No	Train+Val	Bidirectional LSTM memory network.
20	CNN	-	No	Train+Val	CNN with attention and a pointer network, plus a novel top-k attention mechanism.

Figure 4: Summary of approaches used by participants. All teams applied neural approaches, with ESIM (Chen et al., 2018) being a particularly popular basis for system development. External data refers to the man pages for Ubuntu, and course information for Advising. Raw advising refers to the variant of the training data in which the complete dialogues and paraphrase sets are provided. Three teams (5, 9 and 11) did not provide descriptions of their approaches (Gunasekara et al., 2019).