ISSN 2007-9737

Towards Simple but Efficient Next Utterance Ranking

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

Université de Nantes,
France

firstname.lastname@ls2n.fr

Abstract. Retrieval-based dialogue systems converse with humans by ranking candidate responses according to their relevance to the history of the conversation (context). Recent studies either match the context with the response on only sequence level or use complex architectures to match them on the word and sequence levels. We show that both information levels are important and that a simple architecture can capture them effectively. We propose an end-to-end multi-level response retrieval dialogue system. Our model learns to match the context with the best response by computing their semantic similarity on the word and sequence levels. Empirical evaluation on two dialogue datasets shows that our model outperforms several state-of-the-art systems and performs as good as the best system while being conceptually simpler.

Keywords. Dialogue systems, response retrieval, sequence similarity.

1 Introduction

Recently, many works were interested in building neural dialogue systems that converse with humans in natural language by either generating or retrieving responses. Despite the capacity of generative systems to produce customized responses for each conversation context, they tend to generate short and general responses [14]. Thus, they prefer to generate, for example *”I don’t know”* and *”Good !”*, most of the time. This is due essentially to the lack of diversity in their objective function [9]. On the other hand, response retrieval systems are able to provide more accurate and syntactically correct responses [13, 21] by ranking a set of candidate responses based on their coherence with the context. In this work we focus on this category of dialogue systems.

	Context
A	Hi, I can not longer access the graphical **login screen** on ubuntu 12.04
B	what exactly happen?
A	I can’t remember the error message, would it have auto-logged to a file or should I reboot quick?
B	you mean it won’t **automaticaly start** and what happen then?
A	it just stop at a text **screen**, but I can access the command line **login** via alt F1-6, and **start** x **manually** there. I think it might me **lightdm** that’s break but I’m not sure
	Candidate responses
R1	for me **lightdm** often won’t start **automatically** either. It show me console tty1 instead and I have to **start lightdm manually** ✓
R2	what about sources.list ? ✗

Fig. 1. Example of a conversation between two participants (A and B) extracted from the Ubuntu Dialogue Corpus [12]

Given the technical conversation between two users in Figure 1, a response retrieval system should rank the first response before the second one. It is important that the system captures the common information (carried by words written in bold) between the context turns and between the whole context and the candidate response. According to [21], the challenges of the next response ranking task are (1) how to identify important information (words, phrases, and sentences) in the context and how to match this information with those in the response and (2) how to model the relationships between the context utterances.

Most of the recent works use complex architectures to capture sequence and word level

doi: 10.13053/CyS-23-3-3272

information from the context and the candidate response in addition to multiple response matching and aggregation mechanisms [24, 21]. Other works neglect word level information and simply rank candidate responses based on only sequence level information [12, 6, 2, 20, 22]. Some of them use external modules (ex. topic modelling) or have external knowledge requirements (ex. knowledge bases/graphs), making their training and adaptation to different domains more complex.

In this paper, we argue that these approaches suffer from two fundamental drawbacks: the complexity of their architectures and/or their domain dependency. We propose a simple neural architecture that is domain independent and can be trained end-to-end without any external knowledge. We evaluate our approach on two large dialogue datasets of two different languages: the Ubuntu Dialogue Corpus [12] and the Douban Conversation Corpus [21]. We show that the resulting system achieves state-of-the-art performance while being conceptually simpler and having fewer parameters compared to the previous, substantially more complex, systems.

The remainder of this work is as follows: first, we investigate works around retrieval-based dialogue systems. Second, we describe the problem and the architecture of our system. Third, we present the experimental environment and the evaluation results. Then we discuss the results, perform a model visualization and study the errors produced by our system. Finally, we conclude and discuss future work.

2 Related Work

The recently built retrieval-based dialogue systems either match the candidate response with only one dialogue turn of the context *”single-turn”* or with every dialogue turn *”multi-turn”*. In the first category, some early studies consider only the last context turn for matching the response [19, 20] or concatenate the context turns and match them with the response [12, 22, 24, 23]. Even if the architecture of these systems is quite simple, some of them require external modules in order to provide topic words or knowledge bases.

On the other hand, the most recent multi-turn systems [21, 25] highlight the importance of matching the response with every context turn. While these systems achieve higher performances, they require more modules (LSTMs, GRUs, CNNs ..) in order to learn representations of every turn in addition to complex matching mechanisms. Thus, the estimation of the number of turns to consider, the training and adaptation of such architectures become a hard task.

In this work, we propose a single-turn[1] response ranking system that matches the candidate response with the context on two levels. Our model is conceptually simpler and can be easily adapted to other domains since it does not require domain related information.

3 Multi-Level Retrieval-Based Dialogue System

In this section, we formalize the problem that we address and we describe the architecture of our multi-level retrieval-based dialogue system.

3.1 Problem Formalization

Given a conversation context C as a succession of s words w_{ci} such as $C = \{w_{c1}, w_{c2}, w_{c3}, \dots, w_{cs}\}$ and a set of candidate responses R where each candidate response R is a succession of t words w_{rj} such as $R = \{w_{r1}, w_{r2}, w_{r3}, \dots, w_{rt}\}$. The problem consists of selecting the best response R to C. We define the problem as a ranking task in which we want to order candidate responses by their increasing score of suitability to the conversation context. The utterance with the highest score is then chosen as the next utterance[2].

3.2 System Architecture

We propose an end-to-end multi-level context response matching dialogue system. First, we project the context and the candidate response into a distributed representation (word embeddings).

[1] We concatenate all the context turns as one single context.

[2] Note that throughout this paper we use the terms *next utterance*and *response* indifferently.

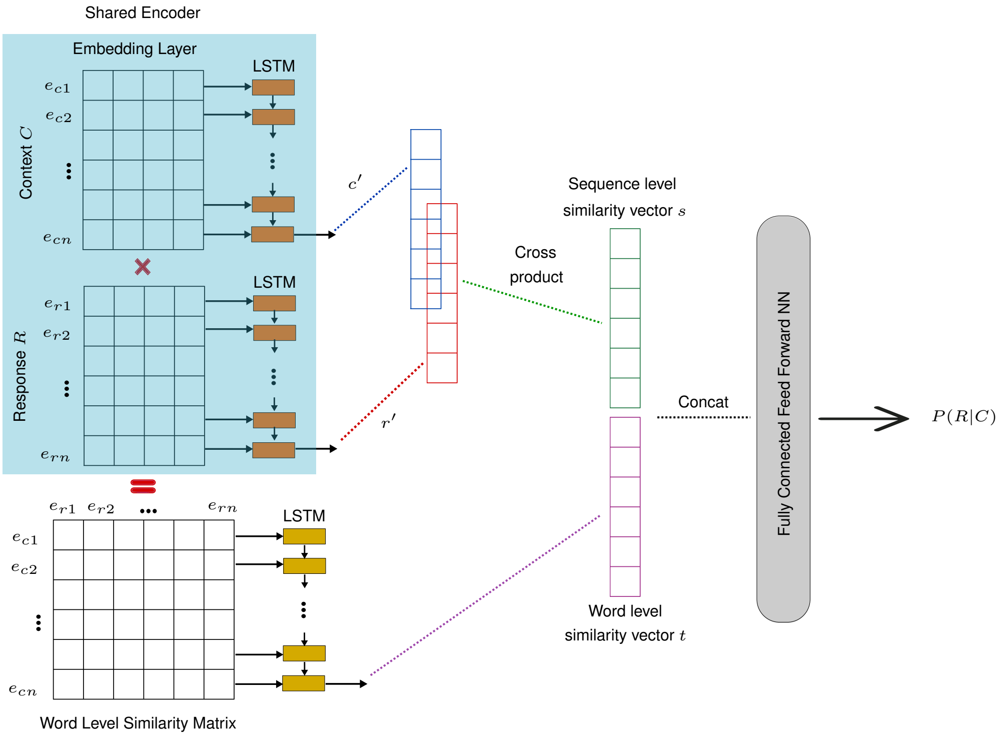

Fig. 2. Architecture of our multi-level context response matching dialogue system

Second, we encode the context and the candidate response into two fixed-size vectors using a shared recurrent neural network (described in Figure 2 with the blue frame). Then, in parallel, we compute two similarities: on word level and sequence level. The sequence level similarity is obtained by multiplying the context and the response vectors. Whereas the word level similarity is obtained by multiplying word embeddings of the context and the candidate response. Both similarities are concatenated and transformed into a probability of the candidate response being the next utterance of the given context. In the following, we elaborate on the functions of our system.

3.2.1 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 (more details are given in 4.4). 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 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 $h_{c,0} = 0$.

3.2.2 Sequence Level Similarity

We hypothesis that positive responses are semantically similar to the context. Thus, the aim of a response retrieval system is to rank the response that shares the most common semantics with 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:

$$s = c' \wedge r' \equiv s = h_{c,n} \wedge h_{r,n}, \tag{2}$$

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

3.2.3 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 learning an improved relationship between C and R. This assumption was consolidated by observing the scores dropping when word level similarity was removed from our system (see section *"Model ablation"*).

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:

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

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 (see Figure 2).

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.

3.2.4 Response Score

At this stage we have two vectors: S representing the similarity between C and R on the sequence level and T representing the word level similarity. 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 as:

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

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

The advantages of our system compared to the state of the art ones are: (1) unlike [22] and [20], in our architecture no external module is required to provide extra information such as topic words or related knowledge; (2) we extract sequence and word level similarity with a simple end-to-end architecture that learns to match the context with the best response by considering all the context utterances.

4 Experimental Setup

In this section we describe our experimental environment. First we provide a description of the datasets on which we evaluated our system. Then we present the baseline systems and the parameter tuning. Finally we provide the evaluation metrics.

	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 1. Statistics on the datasets. *C*, *R* and *cand.* denote context, response and candidate respectively

4.1 Datasets

Ubuntu Dialogue Corpus: [12] collected a large public domain specific corpus of Ubuntu dialogues called the Ubuntu Dialogue Corpus (UDC). The corpus contains conversations with at least three dialogue turns extracted from the chat logs of the channel *#Ubuntu* on the Freenode Internet Relay Chat (IRC)[3]. Conversations from this source are multi users on which heuristics were applied in order to extract two-user discussions. Two versions of this corpus exist. We evaluated our system on the version V1 of the dataset.

Each sample in the training set is a triplet *(context, response, label)*. In the validation and test sets, each sample is made of a context and 10 candidate responses where one is the *ground-truth response* and 9 are negative responses randomly sampled from the corpus. We use the copy shared by [22] in which *numbers*, *urls*, and *paths* were replaced by special placeholders[4]

Douban Conversation Corpus: Douban Conversation Corpus[5] is an open domain corpus extracted from Douban Group by [21]. Douban is a public Chinese social network allowing registered users to record information and create content related to film, books, music, recent events and activities in Chinese cities[6]. The corpus contains more than 1 million conversations between two persons with at least three dialogue turns.

Each dialogue sample in the training and validation sets has one positive and one negative responses randomly sampled from the corpus. In the test set, each dialogue sample may have more than one positive response unlike the test set of the Ubuntu Dialogue Corpus. Labelers were recruited in order to judge whether each candidate response is positive or negative (see section 5.2 of [21] for more details about the corpus). We follow [21] and remove test samples with all positive or all negative responses and thus the test set size is reduced to 6,670 samples. According to the authors, Douban Conversation Corpus is the first human-labeled multi-turn response selection dataset. The task on these datasets consists of ranking the ground-truth response on top of the negative responses. Table 1 summarizes statistics on both corpora.

4.2 Baselines

We report the results of 7 state of the art systems to which we compare our system. We copy the scores produced by the authors in the original papers.

TF-IDF We report results of the Term Frequency-Inverse Document Frequency (TF-IDF) model [13]. 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 [13]. The context and the response were presented using their word embeddings and then they were fed word by word into two an LSTM network to encode them into fixed size vectors. Then a response ranking score is computed using a bilinear model [15].

BiLSTM dual encoder The system of [6] in which the LSTM cells where replaced by bidirectional LSTM cells. We do not report 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.

[3] For the period 2004-2015 available on `https://irclogs.ubuntu.com/`

[4] Available on `https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntu_data.zip`

[5] Available on `https://www.dropbox.com/s/90t0qtji9ow20ca/DoubanConversaionCorpus.zip`

[6] `https://www.douban.com/group`

Deep Learning to Respond (DL2R) Proposed by [23] based on contextually 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 [24] in which a 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 [21]. The candidate response and every dialogue turn of the context are encoded using a GRU network [5]. 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 [25]. This system is an improvement of the SMN [21] in which the Transformer [17] was used in order to produce utterance representations based on self-attention. These representations are matched together to produce self- and cross-attention scores which are stacked as a 3D matching image. Then, a ranking score is produced from this image via convolution and max pooling operations.

4.3 Evaluation Metrics

The evaluation of conversational systems is an open research domain in which there are no standard evaluation metrics [11, 10]. We followed [12, 20, 22, 21] in using *Recall@k*, *Precision@1*, *Mean Average Precision* (MAP) [1] and *Mean Recall Rank* (MRR) [18] as evaluation metrics. These are common metrics in evaluating IR systems such as recommendation systems and research engines, etc. Note that since in UDC each context has one single positive response in among the candidate responses, we only report MRR and R@1 as they are equivalent to MAP and P@1 respectively.

4.4 System Parameters

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. We trained word embeddings of size 300 on UDC and 100 on Douban using FastText [3]. The sizes of the hidden layers of the sequence LSTM and the word LSTM were set to 300 and 200 respectively. The system parameters were updated using Stochastic Gradient Descent with Adam algorithm [7]. All the hyper-parameters were obtained with a grid search on the validation set. We implemented our system with Keras [4] and Theano [16] in backend. We release our source code on `https://github.com/basma-b/multi_level_chatbot`.

5 Results and Analysis

In this section we provide a table summarizing the results of our system and the baseline systems in addition to a visualization of the WLSM matrix, an error analysis and a model ablation study.

5.1 Results

Table 2 summarizes evaluation results on UDC (V1) and Douban Conversation Corpus[7]. Compared to the single-turn systems (the first five 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$_{dynamic}$ [21] 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

[7] We limited the number of baseline systems in our table to the most representative ones of each category. For more systems, we refer to the results Table of [21]

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 [13]	0.659	0.410	0.545	0.708	0.096	0.172	0.405	0.180	0.331	0.359
LSTM [13]	0.901	0.638	0.784	0.949	0.187	0.343	0.720	0.320	0.485	0.527
BiLSTM [6]	0.895	0.630	0.780	0.944	0.184	0.330	0.716	0.313	0.479	0.514
DL2R [23]	0.899	0.626	0.783	0.944	0.193	0.342	0.705	0.330	0.488	0.527
Multi-View [24]	0.908	0.662	0.801	0.951	0.202	0.350	0.729	0.342	0.505	0.543
$\text{SMN}_{dynamic}$ [21]	0.926	0.726	0.847	0.961	0.233	0.396	0.724	0.397	0.529	0.569
DAM [25]	**0.938**	**0.767**	**0.874**	**0.969**	0.254	0.410	0.757	**0.427**	**0.550**	**0.601**
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 2. Evaluation results on the UDC V1 and Douban Corpus using retrieval metrics

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.

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) [25] on both datasets and on all metrics. The DAM as detailed in Section 4.2 is based on multiple layers of the self attention (Transformer) and Convolutional Neural Networks [8]. Even if the advantages of the Transformer are related to the performance improvement and the acceleration of the learning compared to neural networks [17]. However, we proposed an architecture that is fully based on 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 [25], 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[8]) and our system converges after two epochs[9]. Having such architectures (as DAM) makes reproduciblity of results harder due to hardware limitations and 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 Douban corpus in which a context may have more than one ground-truth response and hence every retrieval system must find all the responses.

5.2 Error Analysis

We performed a human evaluation of 200 randomly selected test samples from UDC where the ground-truth response was not retrieved by our system. By observing the test samples that were misclassified, we identified 4 error classes. Table 3 summarizes the distribution of the test samples over these classes. Around 50% of the errors are cases where our system produced a response that is either functionally or semantically equivalent to the ground-truth response.

Error class	Percentage
Functionally equivalent	31%
Semantically equivalent	20%
Out of context	35.5%
Very general responses	13.5%

Table 3. Error classes

In fact, considering these cases as errors may falsify the evaluation. Surprisingly, the other half of errors are due to out of context and very general responses. This drawback was usually noticed in

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

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

generative dialogue systems, however, in this case of study, it is also a major drawback of our retrieval-based dialogue system.

These findings encourage us to perform a deep comparative study between these two categories of dialogue systems.

5.3 Visualization

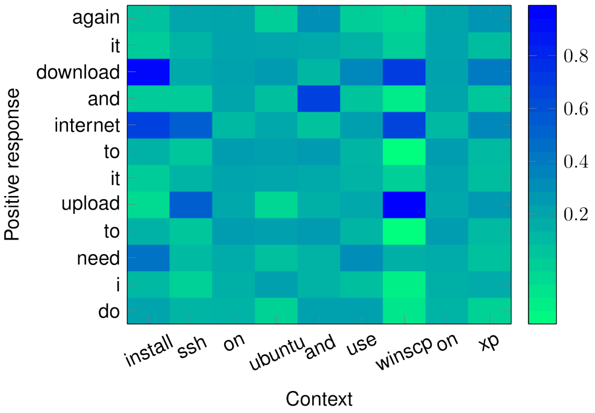

Fig. 3. Visualization of the Word-Level Similarity Matrix (WLSM)

Furthermore, we visualized WLSM for the following test sample. The last turn of the context is:
A: *hey anybody know how i can share file between xp guest and ubuntu 12.04 lts host in vmware ?*
B: *"install ssh on ubuntu and use winscp on xp"*.
The positive response is *"do i need to upload it to internet and download it again"*.

In Figure 3, we plotted the Word Level Similarity Matrix WLSM between the context (x-axis) and the response (y-axis). For a matter of space we visualize only the last dialogue turn (**B**) of the context. As we can see, important (key) words in the context and the response were successfully recognized by our system and were given higher scores.

For instance, *upload*, *internet* and *download* were matched with *install*, *ssh*, *winscp* and *xp*. This observation illustrates the importance of computing word level similarity from word embeddings in order to match the context with the best response.

5.4 Model Ablation

We report in the two last rows of Table 2 the performance of our system while having only one similarity level. We notice that having only one level of similarity causes a drop of the system performance. Results are higher when matching the context with the candidate response on the word level compared to the sequence level. Considering the example of Section 5.3, the whole context and the response are semantically similar. Having in addition to this sequence similarity, the fact that *upload*, *internet* and *download* match with *install*, *ssh* and *winscp* will help the system better recognizing the good responses. Vice versa, we can have responses that share semantically equivalent words with the context while the whole meaning of the response is not related to the whole meaning of the context.

These results highlight the importance of considering both similarity levels in our system in order to achieve higher performances. Note that there is a slight difference in the performance of our system with only one similarity level on both datasets. We believe that this is related to the characteristics of each corpus.

6 Conclusion

We presented a simple and efficient multi-level retrieval-based dialogue system. Our system learns to match the context with the best response based on their similarity that we capture on word and sequence levels with a simple architecture. By learning a word level and sequence level similarities our system was able to capture deep relationships between the context and the candidate responses. The experimental results on two large datasets demonstrate the efficiency of our approach by bringing significant improvements compared to complex state-of-the-art systems.

In essence, a simple model can suffice to achieve good performance, sometimes even better than complex response matching models. As future work, we will extend this study by investigating the possibility of adding more similarity levels while keeping the simplicity of the architecture. Moreover, we plan to enrich text with discursive information such as dialogue acts and rhetorical relations.

7 Acknowledgment

We thank the anonymous reviewers for their valuable comments. This work was partially supported by the project ANR 2016 PASTEL[10].

References

1. **Baeza-Yates, R. A. & Ribeiro-Neto, B. (1999).** *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
2. **Baudiš, P., Pichl, J., Vyskočil, T., & Šedivỳ, J. (2016).** Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.
3. **Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017).** Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics (TACL)*, Vol. 5, pp. 135–146.
4. **Chollet, F. et al. (2015).** Keras. `https://github.com/keras-team/keras`.
5. **Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014).** Empirical evaluation of gated recurrent neural networks on sequence modeling. *Workshop on Deep Learning and Representation Learning at the 28th Annual conference on Advances in Neural Information Processing Systems (NIPS'14)*, Montreal, Canada.
6. **Kadlec, R., Schmid, M., & Kleindienst, J. (2015).** Improved deep learning baselines for ubuntu corpus dialogs. *Workshop on Machine Learning for Spoken Language Understanding and Interaction at the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*, Montreal, Canada.
7. **Kingma, D. & Ba, J. (2015).** Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations (ICLR'15)*, San Diego, CA, USA.
8. **LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998).** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324.
9. **Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016).** A diversity-promoting objective function for neural conversation models. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, San Diego, CA, USA, pp. 110–119.
10. **Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., & Pineau, J. (2016).** How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Austin, Texas, pp. 2122–2132.
11. **Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., & Pineau, J. (2017).** Towards an automatic turing test: Learning to evaluate dialogue responses. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, Vancouver, Canada, pp. 1116–1126.
12. **Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015).** The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'15)*, Prague, Czech Republic, pp. 285–294.
13. **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*, Vol. 8, No. 1, pp. 31–65.
14. **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. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, Copenhagen, Denmark, pp. 2210–2219.
15. **Tenenbaum, J. B. & Freeman, W. T. (2000).** Separating style and content with bilinear models. *Neural computation*, Vol. 12, No. 6, pp. 1247–1283.
16. **Theano Development Team (2016).** Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, Vol. abs/1605.02688.
17. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is all you

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

need. *Advances in Neural Information Processing Systems (NIPS'17)*, Long Beach, CA, USA, pp. 5998–6008.

18. **Voorhees, E. M. (2001).** The trec question answering track. *Natural Language Engineering*, Vol. 7, No. 4, pp. 361–378.

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

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

21. **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. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, Vancouver, Canada, pp. 496–505.

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

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

24. **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. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Austin, Texas, pp. 372–381.

25. **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. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, Melbourne, Australia, pp. 1118–1127.

Article received on 17/01/2019; accepted on 04/03/2019.
Corresponding author is Basma El Amel Boussaha.