Building Chatbots from Forum Data: Model Selection Using Question Answering Metrics

Martin Boyanov, Ivan Koychev
Faculty of Mathematics and Informatics
Sofia University "St. Kliment Ohridski"
Sofia, Bulgaria
mboyanov@gmail.com
koychev@fmi.uni-sofia.bg

Preslav Nakov, Alessandro Moschitti, Giovanni Da San Martino
Qatar Computing Research Institute
HBKU, Doha, Qatar
{pnakov,amoschitti}@hbku.edu.qa
gmartino@hbku.edu.qa

Abstract

We propose to use question answering (QA) data from Web forums to train chatbots from scratch, i.e., without dialog training data. First, we extract pairs of question and answer sentences from the typically much longer texts of questions and answers in a forum. We then use these shorter texts to train seq2seq models in a more efficient way. We further improve the parameter optimization using a new model selection strategy based on QA measures. Finally, we propose to use extrinsic evaluation with respect to a QA task as an automatic evaluation method for chatbots. The evaluation shows that the model achieves a MAP of 63.5% on the extrinsic task. Moreover, it can answer correctly 49.5% of the questions when they are similar to questions asked in the forum, and 47.3% of the questions when they are more conversational in style.

1 Introduction

Recently, companies active in diversified business ecosystems have become more and more interested in intelligent methods for interacting with their customers, and even with their employees. Thus, we have seen the development of several general-purpose personal assistants such as Amazon's Alexa, Apple's Siri, Google's Assistant, and Microsoft's Cortana. However, being general-purpose, they are not a good fit for every specific need, e.g., an insurance company that wants to interact with its customers would need a new system trained on specific data; thus, there is a need for specialized assistants.

This aspect is a critical bottleneck as such systems must be engineered from scratch. Very recently, solutions based on neural networks have been developed, e.g., using seq2seq models (Vinyals and Le, 2015). Such systems provide shallow solutions, but at the same time are easy to train, provided that a large amount of dialog data is available. Unfortunately, the latter is a critical bottleneck as (*i*) the specificity of the domain requires the creation of new data; and (*ii*) this process is rather costly in terms of human effort and time.

Many real-world businesses aiming at acquiring chatbot technology are associated with customer services, e.g., helpdesk or forums, where question answering (QA) sections are often provided, sometimes with user evaluation. Although this data does not follow a dialog format, it is still useful to extract pairs of questions and answers, which are essential to train seq2seq models. Typically, forum or customer care sections contain a lot of content, and thus the requirement about having large datasets is not an issue. The major problem comes from the quality of the text in the pairs that we can extract automatically. One solution is to select data using crowdsourcing, but the task will still be very costly given the required size (hundreds of thousands of pairs) and its complexity.

In this paper, we propose to use data extracted from a standard question answering forum for training chatbots from scratch. The main problem in using such data is that the questions and their associated forum answers are noisy, i.e., not all answers are good. Moreover, many questions and answers are very long, e.g., can span several paragraphs. This prevents training effective seq2seq models, which can only manage (i.e., achieve effective decoding for) short pieces of text.

We tackle these problems by selecting a pair of sentences from each questions–answer pair, using dot product over averaged word embedding representations. The similarity works both (*i*) as a filter of noisy text as the probability that random noise occurs in the same manner in both the question and the answer is very low, and (*ii*) as a selector of the most salient part of the user communication through the QA interaction.

We further design several approaches to model selection and to the evaluation of the output of the seq2seq models. The main idea is, given a question, (*i*) to build a classical vector representation of the utterance answered by the model, and (*ii*) to evaluate it by ranking the answers to the question provided by the forum users. We rank them using several metrics, e.g., the dot product between the utterance and a target answer. This way, we can use the small training, development and test data from a SemEval task (Nakov et al., 2016) to indirectly evaluate the quality of the utterance in terms of Mean Averaged Precision (MAP). Moreover, we use this evaluation in order to select the best model on the development set, while training seq2seq models.

We evaluate our approach using (*i*) our new MAP-based extrinsic automatic evaluation on the SemEval test data, and (*ii*) manual evaluation carried out by four different annotators on two sets of questions: from the forum and completely new ones, which are more conversational but still related to the topics discussed in the forum (life in Qatar). The results of our experiments demonstrate that our models can learn well from forum data, achieving MAP of 63.45% on the SemEval task, and accuracy of 49.50% on manual evaluation. Moreover, the accuracy on new, conversational questions drops very little, to 47.25%, according to our manual evaluation.

2 Related Work

Nowadays, there are two main types of dialog systems: sequence-to-sequence and retrieval-based. Here we focus on the former. Seq2seq is a particular kind of neural network architecture, initially proposed for machine translation (Sutskever et al., 2014). Since then, it has been applied to other tasks such as text summarization (Abigail See, 2017), image captioning (Vinyals et al., 2014), and, of course, dialog modeling (Shang et al., 2015; Li et al., 2016; Gu et al., 2016). The initial seq2seq model assumed that the semantics of the input sequence can be encoded in a single vector, which is hard, especially for longer inputs. Thus, attention mechanisms have been introduced (Bahdanau et al., 2014). This is what we use here as well.

Training seq2seq models for dialog requires large conversational corpora such as Ubuntu (Lowe et al., 2015). Unstructured conversations, e.g., from Twitter, have been used as well (Sordoni et al., 2015). See (Serban et al., 2015) for a survey of corpora for dialog. Unlike typical dialog data, here we extract, filter, and use question-answer pairs from a Web forum.

An important issue with the general seq2seq model is that it tends to generate general answers like *I don't know*, which can be given to many questions. This has triggered researchers to explore diversity promotion objectives (Li et al., 2016). Here, we propose a different idea: select training data based on performance with respect to question answering, and also optimize with respect to a question answering task, where giving general answers would be penalized.

It is not clear how dialog systems should be evaluated automatically, but it is common practice to use BLEU (Papineni et al., 2002), and sometimes Meteor (Lavie and Agarwal, 2007): after all, seq2seq models have been proposed for machine translation (MT), so it is natural to try MT evaluation metrics for seq2seq-based dialog systems as well. However, it has been shown that BLEU, as well as some other popular ways to evaluate a dialog system, do not correlate well with human judgments (Liu et al., 2016). Therefore, here we propose to do model selection as well as evaluation extrinsically, with respect to a related task: Community Question Answering.

3 Data Creation

In order to train our chatbot system, we converted an entire Community Question Answering forum into a set of question–answer pairs, containing only one selected sentence for each question and for each answer.[1] We then used these selected pairs in order to train our seq2seq models. Below, we describe in detail our data selection method along with our approach to question-answer sentence pair selection.

[1] We released the data here: `http://goo.gl/e6UWV6`

3.1 Forum Data Description

We used data from a SemEval task on Community Question Answering (**?**Nakov et al., 2016; **?**). The data consists of questions from the Qatar Living forum[2] and a (potentially truncated) thread of answers for each question. Each answer is annotated as *Good*, *Potentially Useful* or *Bad*, depending on whether it answers the question well, does not answer well but gives some potentially useful information, or does not address the question at all (e.g., talks about something unrelated, asks a new question, is part of a conversation between the forum users, etc.). The goal of the task is to rank the answers so that *Good* answers are ranked higher than *Potentially Useful* and *Bad* ones. The participating systems are evaluated using Mean Average Precision (MAP) as the official evaluation metric.

The data for SemEval-2016 Task 3, subtask A comes split into training, development and test parts with 2,669/17,900, 500/2,440 and 700/3,270 questions/answers, respectively. In addition, the task organizers provided raw unannotated data, which contains 200K questions and 2M answers. Thus, our QA data consists of roughly 2M answers extracted from the forum. We paired each of these answers with the corresponding question in order to make training question–answer pairs for our seq2seq system. We made sure that the development and the testing datasets for SemEval-2016 Task 3 were excluded from this set of training question–answer pairs.

We used the annotated development and test sets both to carry out our new model selection, as explained in Section 4.2, and our new evaluation, as described in Section 5. Both model selection and our proposed evaluation are based on extrinsic evaluation with respect to the SemEval task.

3.2 Sentence Pair Selection

As the questions and the comments[3] in Qatar Living can be quite long, we reduced the question-answer pairs to single-sentence pairs. In particular, given a question-answer pair, we first split the question and the answer from the pair into individual sentences, and then we computed the similarity between each sentence from the question and each sentence from the answer. Ultimately, we kept the most similar pair.

We measured the similarity between two sentences based on the cosine between their corresponding embeddings. We computed the latter as the average of the embeddings of the words in a sentence. We used pre-trained word2vec embeddings that have been fine-tuned[4] for Qatar Living (Mihaylov and Nakov, 2016).

More specifically, we generated the vector representation for each sentence by averaging 300-dimensional word2vec vector representations after stopword removal. We assigned a weight to the word2vec vectors with TF×IDF, where IDF is derived from the entire dataset. Note that averaging has the consequence of ignoring the word order in the sentence. We leave for future work the exploration of more sophisticated sentence representation models, e.g., based on long short-term memory (Hochreiter and Schmidhuber, 1997) and convolutional neural networks (Kim, 2014).

4 Model Selection and Evaluation

In this section, we describe our approach to automatic evaluation as well as model selection for seq2seq models.

4.1 Evaluation

Intrinsic evaluation. We evaluated our model *intrinsically* using BLEU as is traditionally done in dialog systems.

Extrinsic evaluation. We further performed *extrinsic* evaluation in terms of how much the answers we generate can help solve the SemEval CQA task. In particular, we input each of the test questions from SemEval to the trained seq2seq model, and we obtained the generated answer. Then, we calculated the similarity, e.g., TF×IDF-based cosine (see below for more detail), between that seq2seq-generated answer and each of the answers in the thread, and we ranked the answers in the thread based on this similarity. Finally, we calculated MAP for the resulting ranking, which evaluates how well we do at raning the *Good* answers higher than the not-Good ones (i.e., *Bad* or *Potentially Useful*). As a baseline, we used the MAP ranking produced by comparing the answers to the question (instead of the generated answer).

4.2 Model Selection

The training step produces a model that evolves over the training iterations. We evaluated that

[2]Qatar Living: http://www.qatarliving.com

[3]In our forum, the comments are considered as answers.

[4]https://github.com/tbmihailov/semeval2016-task3-CQA#resources

model after each 2,000 minibatch iterations. Then, among these evaluated models, we selected the best one, which we used for the test set. We used three model selection approaches, optimizing for MAP and for BLEU calculated on the development set of the SemEval-2016 Task 3, and for the seq2seq loss on the training dataset.

Seq2seq loss. We consider the loss that the seq2seq model optimizes during the training phase. Notice that in this case no development set is required for model selection.

Machine translation evaluation measure (BLEU). A standard model selection technique for seq2seq models is to optimize BLEU. Here, we calculated multi-reference BLEU between the generated response and the *Good* answers in the thread (on average, there are four *Good* answers out of ten in a thread). We then take the average score over all threads in the development set.

Extrinsic evaluation based on MAP. The main idea for this model selection method is the following: given a question, the seq2seq model produces an answer, which we compare to each of the answers in the thread, e.g., using cosine similarity (see below for detail), and we use the score as an estimation of the likelihood that an answer in the thread would be good. In this way, the list of candidate comments for each question can be ranked and evaluated using MAP. We used the gold relevancy labels available in the development dataset to compute MAP.

More formally, given an utterance u_q returned by the seq2seq model in response to a forum question, we rank the comments $c_1, \ldots, c_n$ from the thread according to the values $r(u_q, c_i)$. We considered the following options for $r(u_q, c_i)$:

- **cos**: this is the cosine between the embedding vectors of u_q and c_i, where the embeddings are calculated as the average of the embedding vectors of the words, using the fine-tuned embedding vectors from (Mihaylov and Nakov, 2016);
- **BLEU**: this is the sentence-level BLEU+1 score between u_q and c_i;
- **bm25**: this is the BM25 score (Robertson and Zaragoza, 2009) between u_q and c_i;
- **TF×IDF**: we build a TF×IDF vector, where the TF is based on the frequency of the words in u_q, and the IDF is calculated based on the full SemEval data (all 200K questions and all 2M answers), we then repeat the procedure to obtain a vector for c_i, and finally we compute the cosine between these two vectors;

We also define a variant of each of the $r(x, y)$ functions above, where the similarity score is further summed with the TF×IDF-cosine similarity between the question and the comment (**+qc-sim**). Finally, we define yet another metric, **Avg**, as the average of all $r()$ functions defined in this section.

5 Experiments

We compare the model selection approaches described in Section 4.2 above, with the goal to devise a seq2seq system that gives fluent, *good* and informative answers, i.e., avoids answers such as *"I don't know"*.

5.1 Setup

Our model is based on the seq2seq implementation in TensorFlow. However, we differ from the standard setup in terms of preprocessing, postprocessing, model selection, and evaluation.

First, we learned subword units using byte pair encoding (Sennrich et al., 2015) on the full data. Then, we encoded the source and the questions and the answers using these learned subword units. We reversed the source sequences before feeding them to the encoder in order to diminish the effect of vanishing gradients. We also applied padding and truncation to accommodate for bucketing. We then trained the seq2seq model using stochastic gradient descent.

Every 2,000 iterations, we evaluated the current model with the metrics from Section 4.2. These metrics are later used to select the model that is most suitable for our task, thus avoiding overfitting on the training data.

In our experiments, we used the following general parameter settings: (*i*) vocabulary size: 40,000 subword units; (*ii*) dimensionality of the embedding vectors: 512; (*iii*) RNN cell: 2-layered GRU cell with 512 units; (*iv*) minibatch size: 80; (*v*) learning rate: 0.5; (*vi*) buckets: [(5, 10), (10, 15), (20, 25), (40,45)].

5.2 Results and Discussion

In our first experiment, we explore the performance of seq2seq models produced by optimizing

	Optimizing for	MAP	BLEU	Iteration	Ans. Len.
1	MAP	63.45	9.18	192,000	10.56
2	BLEU	62.64	8.16	16,000	16.31
3	seq2seq loss	62.81	7.00	200,000	8.73
4	Baseline	52.80	-	-	-

Table 1: Evaluation results using the seq2seq model and optimizing during training for MAP (on DEV) vs. BLEU (on DEV) vs. the seq2seq loss (on TRAIN). The following columns show some results on TEST when selecting the best training model on DEV (for MAP and BLEU) and on TRAIN (for the seq2seq loss). We report BLEU and MAP, as well as the iteration at which the best value was achieved on DEV/TRAIN, and the average length of the generated answers on TEST.

	MAP	
Ranking Metric	**Dev**	**Test**
TF×IDF+qc-sim	63.56	63.45
TF×IDF	62.46	62.03
cos-embeddings+qc-sim	62.97	62.90
cos-embeddings	62.21	62.13
bm25+qc-sim	62.81	61.96
bm25	62.88	61.77
BLEU+qc-sim	62.67	62.73
BLEU	59.94	59.82
Avg	62.84	62.33

Table 2: MAP score for the ranking strategies defined in Section 4.2, evaluated on the development and on the test datasets.

MAP using the different variants of the similarity function $r()$ from Section 4.2, with MAP for model selection. The results in Table 2 show that **TF×IDF+qc-sim** performs best. The results are consistent on the development (63.56) and on the test datasets (63.45). The absolute improvement with respect to **cos-embeddings+qc-sim** is +0.59 on the development and +0.55 on the test dataset, respectively.

In a second experiment, we compared model selection strategies when optimizing for MAP (**TF×IDF+qc-sim**) vs. BLEU vs. seq2seq loss. We further report the results for a baseline for the SemEval2016 Task 3, subtask A (Nakov et al., 2016), which picks a random order for the answers in the target question-answer thread. The results are shown on Table 1. For each model selection criterion, we report its performance and statistics on the test dataset about the model that was best-performing on the development dataset.

We can see that doing model selection according to MAP yielded not only the highest ranking performance of 63.45 but also the best BLEU score. This is even more striking if we consider that BLEU tends to favor longer answers, but the average length of the seq2seq answers is 10.56 for MAP and 16.31 for BLEU score. Thus, we have shown that optimizing for an extrinsic evaluation measure that evaluates how good we are at telling *Good* from *Bad* answers works better than optimizing for BLEU.

6 Manual Evaluation and Error Analysis

We evaluated the three approaches in Table 1 on 100 relatively short questions.[5] First, we randomly selected 50 questions from the test set of SemEval-2016 Task 3. However, we did not use the original questions, as they can contain multiple sentences and thus can be too long; instead, we selected a single sentence that contains the core of the question (and in some cases, we simplified it a bit further). We further created 50 new questions, which are more personal and conversational in nature, but are still generally related to Qatar. The answers produced by the three systems for these 100 questions were evaluated independently by four annotators, who judged whether each of the answers is good.

6.1 Quantitive Analysis

Table 3 reports the number of good answers that each of the annotators has judged to be good when the model is selected based on BLEU, MAP, and seq2seq loss. The average of the four annotators suggests that optimizing for MAP yields the best overall results. Note that all systems perform slightly worse on the second set of questions. This should be expected as the latter are different from those used for training the models. Overall, the MAP-based system appears to be more robust,

[5]The questions and the outputs of the different models are available at http://goo.gl/w9MZfv

	Optimizing for	# *Good* answers according to Ann. 1	Ann. 2	Ann. 3	Ann. 4	Avg.
	questions 1-50					
1	MAP	23	29	21	26	24.75 (49.50%)
2	BLEU	8	15	11	8	10.50 (21.00%)
3	seq2seq loss	18	26	21	28	23.25 (46.50%)
	questions 51-100					
4	MAP	28	20	13	29	22.50 (45.00%)
5	BLEU	9	5	2	6	5.50 (11.00%)
6	seq2seq loss	25	14	11	24	18.50 (37.00%)
	questions 1-100					
7	MAP	51	49	34	55	47.25 (47.25%)
8	BLEU	17	20	13	14	16.00 (16.00%)
9	seq2seq loss	43	40	32	52	41.75 (41.75%)

Table 3: Number of good answers according to manual annotation of the answers to 50+50 questions by the three models from Table 1.

with only 2.25 points absolute decrease in performance (compared to 5 and 4.75 for the systems using BLEU and seq2seq loss, respectively).

6.2 Qualitative Analysis

We now analyze the quality of the generated answers from the manual evaluation. Instead of looking at overall numbers, here we look at some interesting cases, shown in Tables 4 and 5.

First, we can confirm that the answers generated by the model that was optimized for BLEU seem to be the worst. We attribute this to the relatively early iteration when the optimal BLEU occurs and also to the nature of the BLEU metric. BLEU tries to optimize for n-gram matching. Thus, the selected model ultimately prefers longer utterances, while the other two models focus on providing a short focused answer; this is especially true for the first part of the manual test set as shown in Table 4, where we can find "safe" answers with stopwords, which do not have much informative content, but are a good bet, e.g., "good luck", "I think that", "it is good to", etc.

In examples 1 and 6, we can see that only the MAP-based model addressed the question directly. The other models are a better fit to the language model, and thus failed to produce the target named entity. In example 1, they produced a generic answer, which can be given in response to many questions.

Example 5 shows how the models have trouble handling exclusion/negation. The model was able to copy the named entity, which is generally a good thing to do, but not here. If the question was simply "Where can I go running?", the answer would have been good.

Note that the responses to the questions from the second group are more personal, e.g., they start with "I think", "I will", etc. Finally, we can see the well-known problem with seq2seq models: repetition. This is evident in examples 3, 13, 14.

7 Conclusion

Building dialog systems, e.g., in the form of chatbots, has attracted a lot of attention recently and thus has become a crucial investment for many companies. Progress in neural networks, especially in seq2seq models, has made it possible to quickly and directly learn chatbots from data. However, the availability of domain-specific training data coming from real dialogs is a critical bottleneck for the development of dialog systems.

We addressed the problem by producing training data from Community Question Answering (CQA) forums. We further applied sentence selection based on word embeddings in order to retain only meaningful pairs of short texts, which can be more effectively used for training seq2seq models.

Moreover, we introduced the use of extrinsic evaluation based on a CQA task using MAP to select the most effective models among those generated during training, using a development set of good user answers available from the cQA data.
We also used MAP to perform automatic evaluation of system accuracy against the test set annotated for the CQA task. This was not explored

#	Model	Example
1		*How good are Karwa services?*
	BLEU	good luck
	seq2seq	good luck
	MAP	**karwa is good .**
2		*Which is the best spa in Doha?*
	BLEU	lulu
	seq2seq	**al sharq spa**
	MAP	**al sharq spa**
3		*Which is better: Qtel or Vodafone?*
	BLEU	go for the bank
	seq2seq	**vodafone**
	MAP	vodafone is better than vodafone .
4		*Can you recommend a good Vet in Doha?*
	BLEU	good luck
	seq2seq	**dr . choc**
	MAP	**dr . choc**
5		*Where can I go running; other than the Corniche?*
	BLEU	i think it is a good place to go to doha .
	seq2seq	corniche
	MAP	corniche
6		*Are there any cameras on the roundabouts in Qatar?*
	BLEU	i think it is a good offer for a few months .
	seq2seq	there are many cars in qatar .
	MAP	**there is no radar in qatar .**
7		*What is the best mall in Doha to buy good furniture?*
	BLEU	lulu is good
	seq2seq	carrefour
	MAP	**ikea**

Table 4: Output of our chatbot when using different model selection strategies for some simplified questions from Qatar Living, which we used for the first part of our manual evaluation. The good answers are shown in bold.

#	Model	Example
8		*Do you like fast food?*
	BLEU	**i like the one in doha**
	seq2seq	**i do n ' t eat food .**
	MAP	**i hate food .**
9		*What is your favorite beach in Qatar?*
	BLEU	i think it is a good place
	seq2seq	**sealine beach**
	MAP	**sealine beach**
10		*Tell me bot, what do you know about Qatar?*
	BLEU	i think it is a good offer to qatar .
	seq2seq	qatar living
	MAP	**qatar is a country .**
11		*What are the things about Qatar you like the most?*
	BLEU	i think it ' s a good one ...
	seq2seq	qatar is a country .
	MAP	**qatar is the best place to live .**
12		*Are you looking forward to the FIFA World Cup in 2022?*
	BLEU	i think it was a good one in doha
	seq2seq	i will win the world cup .
	MAP	i will win the world cup .
13		*Do you think the fast food in Qatar is of good quality?*
	BLEU	i think it is a good place to go to doha .
	seq2seq	**food is good .**
	MAP	food is good for food .
14		*What is the most useful thing that you can tell me about life in Qatar?*
	BLEU	i love my life ...
	seq2seq	life is life ... : - rrb -
	MAP	**life is beautiful ...**
15		*Do you need to Google to give me an answer?*
	BLEU	i think you can get a good one in your home
	seq2seq	i will give you the answer .
	MAP	**i would like to answer .**

Table 5: Output of our chatbot for some new, conversational questions related to Qatar, which we created and used for the second part of our manual evaluation. The good answers are shown in bold.

before.

Finally, we carried out manual evaluation with four different annotators on two different sets of questions: the first set used simplified questions taken from the CQA data, whereas the second one was composed of new, conversational-style questions that we generated. Thus, the questions of the second set are rather different from those used to train the systems; yet, they are about topics that are generally discussed in the training data.

We have found that the seq2seq model can learn from CQA-derived data, producing accurate answers when answering forum questions according to automatic and manual evaluation, with MAP of 63.45, and accuracy of 49.50, respectively. Moreover, the accuracy on completely new questions drops by only few points, i.e., to 47.25, according to our manual evaluation.

Interestingly, our model selection approach is more accurate than using the loss of the seq2seq model, and performs much better than using BLEU score. Indeed, the latter seems not to be very appropriate for evaluating chatbots (at least such trained on CQA data), as our manual analysis shows.

In future work, we would like to study new methods for selecting data, so that the overall system accuracy can improve further. We also plan to explore sub-word embedding representations (Bojanowski et al., 2016) that could better capture typos, which are common in Web forums. Last but not least, we plan to experiment with other languages such as Arabic.

Acknowledgments

This research was performed by the Arabic Language Technologies group at Qatar Computing

Research Institute, HBKU, within the Interactive sYstems for Answer Search project (IYAS).

References

Peter Liu Christopher Manning Abigail See. 2017. Get to the point: Summarization with pointer-generator networks. In *Advances in neural information processing systems*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR* abs/1607.04606. http://arxiv.org/abs/1607.04606.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany, pages 1631–1640.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pages 1746–1751.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic, StatMT '07, pages 228–231.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL*. San Diego, California, pages 110–119.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, pages 2122–2132.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 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*. Prague, Czech Republic, pages 285–294.

Todor Mihaylov and Preslav Nakov. 2016. Semanticz at semeval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*. San Diego, California, pages 804 – 811.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia, pages 746–751.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community question answering. pages 525–545.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania, ACL '02, pages 311–318.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.* 3(4):333–389. https://doi.org/10.1561/1500000019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR* abs/1508.07909. http://arxiv.org/abs/1508.07909.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *CoRR* abs/1512.05742. http://arxiv.org/abs/1512.05742.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of ACL-IJCNLP*. Association for Computational Linguistics, Beijing, China, pages 1577–1586.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of HLT-NAACL*. Denver, Colorado, pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR* abs/1506.05869. http://arxiv.org/abs/1506.05869.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR* abs/1411.4555. http://arxiv.org/abs/1411.4555.