AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen,
Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu
Alibaba Group, Hangzhou, China
{minghui.qmh,fenglin.lfl}@alibaba-inc.com

Abstract

We propose *AliMe Chat*, an open-domain chatbot engine that integrates the joint results of Information Retrieval (IR) and Sequence to Sequence (Seq2Seq) based generation models. *AliMe Chat* uses an attentive Seq2Seq based rerank model to optimize the joint results. Extensive experiments show our engine outperforms both IR and generation based models. We launch *AliMe Chat* for a real-world industrial application and observe better results than another public chatbot.

1 Introduction

Chatbots have boomed during the last few years, e.g., Microsoft's XiaoIce, Apple's Siri, Google's Google Assistant. Unlike traditional apps where users interact with them through simple and structured language (e.g., "submit", "cancel", "order", etc.), chatbots allow users to interact with them using natural language, text or speech (even image).

We are working on enabling bots to answer customer questions in the E-commerce industry. Currently, our bot serves millions of customer questions per day (mainly Chinese, also some English). The majority of them is business-related, but also around 5% of them is chat-oriented (several hundreds of thousands in number). To offer better user experience, it is necessary to build an open-domain chatbot engine.

Commonly used techniques for building open-domain chatbots include IR model (Ji et al., 2014; Yan et al., 2016b) and generation model (Bahdanau et al., 2015; Sutskever et al., 2014; Vinyals and Le, 2015). Given a question, the former retrieves the nearest question in a Question-Answer (QA) knowledge base and takes the paired answer, the latter generates an answer based on a pre-trained Seq2Seq model. Often, IR models fail to handle long-tail questions that are not close to those in a QA base, and generation models may generate inconsistent or meaningless answers (Li et al., 2016; Serban et al., 2016).

To alleviate these problems, we propose a hybrid approach that integrates both IR and generation models. In our approach, we use an attentive Seq2Seq rerank model to optimize the joint results. Specifically, for a question, we first use an IR model to retrieve a set of QA pairs and use them as candidate answers, and then rerank the candidate answers using an attentive Seq2Seq model: if the top candidate has a score higher than a certain threshold, it will be taken as the answer; otherwise the answer will be offered by a generation based model (see Fig. 1 for the detailed process).

Our paper makes the following contributions:

- We propose a novel hybrid approach that uses an attentive Seq2Seq model to optimize the joint results of IR and generation models.
- We conducted a set of experiments to assess the approach. Results show that our approach outperforms both IR and generation.
- We compared our chatbot engine with a public chatbot. Evidence suggests that our engine has a better performance.
- We launched *AliMe Chat* for a real-world industrial application.

The rest of the paper is structured as follows: Section 2 presents our hybrid approach, followed by experiments in Section 3, related work is in Section 4, and Section 5 concludes our work.

2 A Seq2Seq based Rerank Approach

We present an overview of our approach in Fig. 1. At first, we construct a QA knowledge base from the chat log of our online customer service cen-

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 498–503
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2079

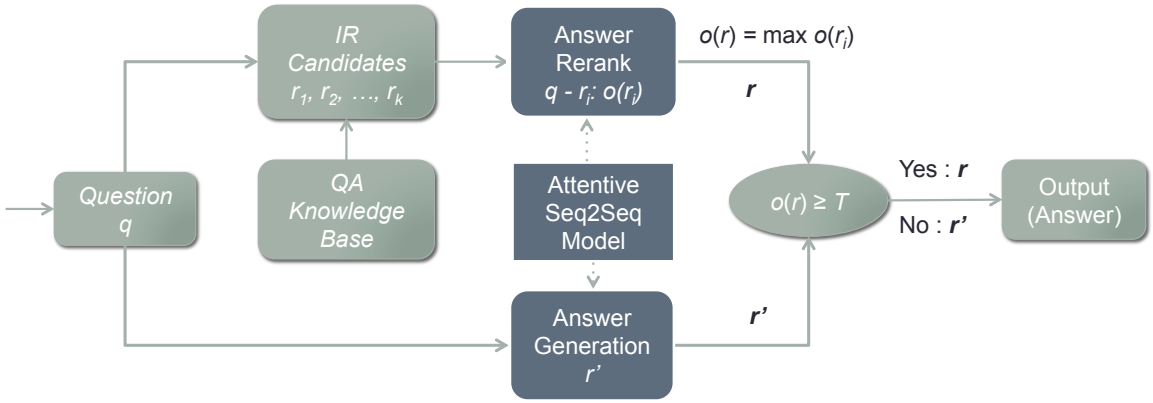

Figure 1: Overview of our hybrid approach.

ter. Based on this QA knowledge base, we then develop three models: an IR model, a generation based model and a rerank model. There are two points to be noted: (1) all the three models are based on words (i.e., word segmentation is needed): the input features of IR model are words, while those of generation model and rerank model are word embeddings, which are pre-trained using fasttext (Bojanowski et al., 2016) and further fine-tuned in the two models; (2) our generation based model and rerank model are built on the same Seq2Seq structure, the former is to generate an output while the latter is to score the candidate answers with regarding to an input question.

Given an input question q and a threshold T, the procedure of our approach is as follows:

- First, we use the IR model to retrieve a set of k candidate QA pairs $\langle q_{\text{kb}_i}, r_i \rangle_{i=1}^{k}$ ($k = 10$).
- Second, we pair q with each candidate answer r_i and calculate a confidence score $o(r_i) = s(q, r_i)$ for each pair using the scoring function in Eqn. 2 of the rerank model.
- Third, we consider the answer r with the maximal score $o(r) = \max o(r_i)$: if $o(r) \geq T$, take the answer r; otherwise output a reply r' from the generation based model.

Here, the threshold T is obtained through an empirical study, to be discussed in Section 3.2.

2.1 QA Knowledge Base

We use the chat log of our online customer service center between 2016-01-01 and 2016-06-01 as our original data source (the conversations are taken between customers and staff). We construct QA pairs from conversations by pairing each question with an adjacent answer. When needed, we flatten consecutive questions (resp. answers) by concatenating them. After that, we filter out QA pairs that contain business related keywords. Finally, we obtained 9,164,834 QA pairs.

2.2 IR Model

Our retrieval model employs search technique to find the most similar question for each input and then obtain the paired answer. With word segmentation, we build an inverted index for the set of all 9,164,834 questions by mapping each word to a set of questions that contain that word. Given a question, we segment it into a set of words, remove stop words, extend the set with their synonyms, and use the refined set to call back a set of QA candidate pairs. We then employ BM25 (Robertson et al., 2009) to calculate similarities between the input question and the retrieved questions, and take the paired answer of the most similar one as the answer.

2.3 Generation based Model

Our generation based model is built on the attentive Seq2Seq structure (Bahdanau et al., 2015).

Let $\theta_i = \{y_1, y_2, \cdots, y_{i-1}, c_i\}$, the probability of generating a word y_i at position i is given by Eqn. 1, where f is a nonlinear function that computes the probability, s_{i-1} is the hidden state of the output at position $i-1$, c_i is a context vector that depends on $(h_1, h_2, \cdots, h_m)$, the hidden states of the input sequence: $c_i = \sum_{j=1}^{m} \alpha_{ij} h_j$, $\alpha_{ij} = a(s_{i-1}, h_j)$ is given by an alignment model that scores how well the input at position j matches to the output at $i-1$ (Bahdanau et al., 2015). An example is shown in Fig. 2, where $i = 3$ and $m = 4$.

$$\begin{aligned} p(y_i = w_i|\theta_i) &= p(y_i = w_i|y_1, y_2, \ldots, y_{i-1}, c_i) \\ &= f(y_{i-1}, s_{i-1}, c_i) \end{aligned} \quad (1)$$

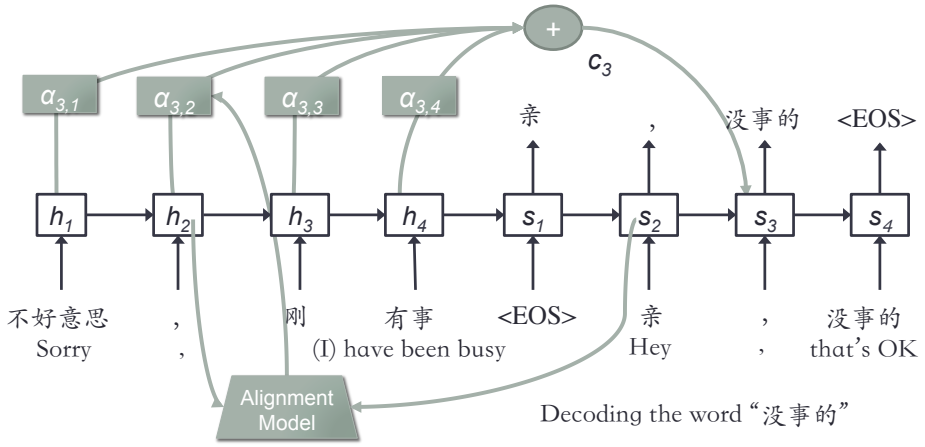

Figure 2: Attentive Seq2Seq model. Our model is mainly for Chinese.

We choose Gated Recurrent Units (GRU) as our Recurrent Neural Network (RNN) unit. A few important implementations are discussed below.

Bucketing and padding. To handle questions and answers of different lengths, we employ the bucket mechanism proposed in Tensorflow [1]. We use five buckets $(5, 5)$, $(5, 10)$, $(10, 15)$, $(20, 30)$, $(45, 60)$ to accommodate QA pairs of different length, e.g., a question of length 4 and an answer of length 8 will be put in bucket $(5, 10)$, and pad questions and answers with a special symbol "_PAD" when needed.

Softmax over sampled words. To speed up the training process, we apply softmax to a set of sampled vocabulary words (the target word and 512 random ones) rather than the whole set. The idea is similar with the importance sampling strategy in (Jean et al., 2014).

Beam search decoder. In the decode phase, we use beam search, which maintains top-k ($k = 10$) output sequences at each moment t, instead of greedy search, which keeps only one at each time t, to make our generation more reasonable.

2.4 Attentive Seq2Seq Rerank Model

Our rerank model uses the same attentive Seq2Seq model to score candidate answers with regarding to an input question. Specifically, we choose *mean probability*, denoted as $s^{\text{Mean-Prob}}$ in Eqn. 2, as our scoring function (a candidate answer is treated as a word sequence $w_1, w_2, \cdots, w_n$). We have also tried *inverse of averaged cross-entropy* and *harmonic mean*, but they had a poorer performance.

$$s^{\text{Mean-Prob}} = \frac{1}{n}\sum_{i=1}^{n} p(y_i = w_i|\theta_i) \qquad (2)$$

[1] https://www.tensorflow.org/tutorials/seq2seq

3 Experiments

In our experiments, we first examined the effectiveness of attentive Seq2Seq model with the scoring criterion *mean probability*; we then evaluated the effectiveness of IR, Generation, IR + Rerank, IR + Rerank + Generation (our approach); we also conducted an online A/B test on our approach and a baseline chatbot engine; lastly, we compared our engine with a publicly available chatbot.

For evaluation, we have business analysts go through the answer of each testing question (two analysts for the experiment comparing with another public chatbot, and one for the other experiments), and mark them with three graded labels: "0" for unsuitable, "1" means that the answer is only suitable in certain contexts, "2" indicates that the answer is suitable. To determine whether an answer is suitable or not, we define five evaluation rules, namely "right in grammar", "semantically related", "well-spoken language", "context independent" and "not overly generalized". An answer will be labeled as suitable only if it satisfies all the rules, neutral if it satisfies the first three and breaks either of the latter two, and unsuitable otherwise.

We use top-1 accuracy (P_{top_1}) as the criterion because the output of some approaches can be more than one (e.g., IR). This indicator measures whether the top-1 candidate is suitable or neutral, and is calculated as follows: $P_{top_1} = (N_{suitable} + N_{neutral})/N_{total}$, where $N_{suitable}$ means the number of questions marked as suitable (other symbols are defined similarly).

3.1 Evaluating Rerank Models

We first compared two Seq2Seq models (the basic one proposed in (Cho et al., 2014), the attentive one presented in Section 2.4), on three scor-

ing criteria (*mean probability*, *inverse of averaged cross-entropy* and *harmonic mean*) using a set of randomly sampled 500 questions. We show the P_{top_1} result in Table 1, which suggests that the attentive Seq2Seq model with $s^{\text{Mean-Prob}}$ has the best performance. We use it in our rerank model.

	IR+Rerank			IR
	$s^{\text{Mean-Prob}}$	$s^{\text{Cross-Ent}}$	s^{HM}	
Basic	0.48	0.48	0.47	0.47
Attentive	**0.54**	0.53	0.53	

Table 1: Comparison of different rerank models.

3.2 Evaluating Candidate Approaches

We then evaluated the effectiveness of the following four approaches with another set of 600 questions: IR, Generation, IR + Rerank, IR + Rerank + Generation. We present the result in Fig. 3. Clearly the proposed approach (IR + Rerank + Generation) has the best top-1 accuracy: with a confidence score threshold $T = 0.19$, $P_{top_1} = 60.01\%$. Here, questions with a score higher than 0.19 (the left of the dashed line, 535 out of 600), are answered using rerank, and the rest is handled by generation. The P_{top_1} for the other three alternatives are 47.11%, 52.02%, and 56.23%, respectively. Note that a narrowly higher P_{top_1} can be achieved if a higher threshold is used (e.g., 0.48), or, put differently, rerank less and generate more. We use the lower threshold because of the uncontrollability and poor interpretability of Seq2Seq generation: with an elegant decrease at the P_{top_1}, we gain more controllability and interpretability.

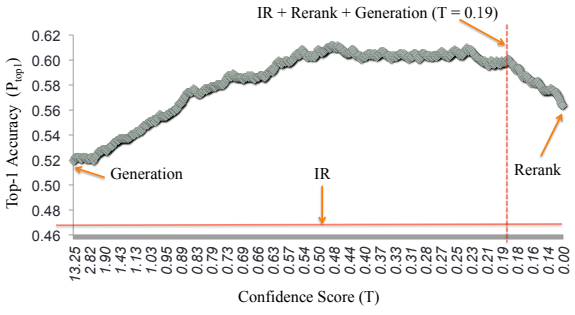

Figure 3: Top-1 accuracy of candidate approaches.

3.3 Online A/B Test

We implemented the proposed method in *AliMe Chat*, our online chatbot engine, and conducted an A/B test on the new and the existing IR method (questions are equally distributed to the two approaches). We randomly sampled 2136 QA pairs, with 1089 questions answered by IR and 1047 handled by our hybrid approach, and compared their top-1 accuracies. As shown in Table 2, the new approach has a P_{top_1} of 60.36%, which is much higher than that of the IR baseline (40.86%).

Model	N_{total}	$N_{unsuitable}$	$N_{neutral}$	$N_{suitable}$	P_{top_1}
IR	1089	644	384	61	40.86%
Hybrid	1047	415	504	128	60.36%

Table 2: Comparison with IR model in A/B test.

3.4 Comparing with a Public Chatbot

To further evaluate our approach, we compared it with a publicly available chatbot [2]. We select 878 out of the 1047 testing questions (used in the A/B test) by removing questions relevant to our chatbot, and use it to test the public one. To compare their answers with ours, two business analysts were asked to choose a better response for each testing question. Table 3 shows the averaged results from the two analysts, clearly our chatbot has a better performance (better on 37.64% of the 878 questions and worse on 18.84%). The Kappa measure between the analysts is 0.71, which shows a substantial agreement.

	Win	Equal	Lose
Number	330	382	165
Percentage	37.64%	43.52%	18.84%

Table 3: Comparison with another chatbot.

3.5 Online Serving

We deployed our approach in our chatbot engine. For online serving, reranking is of key importance to run time performance: if k candidate QA pairs are ranked *asynchronously*, the engine has to wait for the last reranker and it will get worse when QPS (questions per second) is high. Our solution is to bundle each k QA pairs together and transform it into a $k \times n$ matrix (n is the maximal length of the concatenation of the k QA pairs, padding is used when needed), and then make use of parallel matrix multiplication in the rerank model to accelerate the computation. In our experiments, the batch approach helps to save 41% of the processing time when comparing with the *asynchronous* way. Specifically, more than 75% of questions take less than 150ms with rerank and less than 200ms with generation. Moreover, our engine is able to support a peak QPS of 42 on a cluster of 5 service instances, with each reserving 2 cores and 4G memory on an Intel Xeon E5-2430 server. This makes our approach applicable to industrial bots.

[2] http://www.tuling123.com/

We launched *AliMe Chat* as an online service and integrated it into *AliMe Assist*, our intelligent assistant in the E-commerce field that supports not only chatting but also customer service (e.g., sales return), shopping guide and life assistance (e.g., book flight). We show an example chat dialog generated by our chat service [3] in Fig 4

Figure 4: An example chat dialog of *AliMe Chat*.

4 Related Work

Closed-domain dialog systems typically use rule- or template- based methods (Williams and Zweig, 2016; Wen et al., 2016), and dialog state tracking (Henderson, 2015; Wang and Lemon, 2013; Mrksic et al., 2015). Differently, *open-domain* chatbots often adopt data-driven techniques. Commonly used include *IR* and *Seq2Seq generation*.

IR based techniques mainly focus on finding the nearest question(s) from a QA knowledge base for an input question, e.g., (Isbell et al., 2000), (Ji et al., 2014), (Yan et al., 2016b). A recent work (Yan et al., 2016a) has tried a neural network based method for matching. Usually, IR based models have difficulty in handling long-tail questions.

Seq2Seq based generation models are typically trained on a QA knowledge base or conversation corpus, and used to generate an answer for each input. In this direction, RNN based Seq2Seq models are shown to be effective (Cho et al., 2014; Sutskever et al., 2014; Ritter et al., 2011; Shang et al., 2015; Sordoni et al., 2015; Serban et al., 2016). A basic Seq2Seq model is proposed in (Sutskever et al., 2014), and enhanced with attention by (Bahdanau et al., 2015). Further, Sordoni et al. (2015) considered context information, Li et al. (2016) tried to let Seq2Seq models generate diversified answers by attaching a diversity-promoting objective function. Despite many efforts, Seq2Seq generation models are still likely to generate inconsistent or meaningless answers.

Our work combines both IR based and generation based models. Our work differs from another recent combinational approach (Song et al., 2016) in that they use an IR model to rerank the union of retrieved and generated answers. Furthermore, we found that our attentive Seq2Seq rerank approach helps to improve the IR results significantly.

5 Conclusion

In this paper, we proposed an attentive Seq2Seq based rerank approach that combines both IR and generation based model. We have conducted a series of evaluations to assess the effectiveness of our proposed approach. Results show that our hybrid approach outperforms both the two models. We implemented this new method in an industrial chatbot and released an online service.

There are many interesting problems to be further explored. One is *context*, which is of key importance to multi-round interaction in dialog system. Currently, we use a simple strategy to incorporate context: given a question, if less than three candidates are retrieved by the IR model, we enhance it with its previous question and sent the concatenation to the IR engine again. We have tried other context-aware techniques, e.g. context sensitive model (Sordoni et al., 2015), neural conversation model (Sutskever et al., 2014), but they do not scale up well in our scenario. We are still exploring scalable context-aware methods. Also, we are working on *personification*, i.e., empowering our chatbot with characters and emotions.

Acknowledgments

The authors would like to thank Juwei Ren, Lanbo Li, Zhongzhou Zhao, Man Yuan, Qingqing Yu, Jun Yang and other members of Alibaba Cloud for helpful discussions and comments. We would also like to thank reviewers for their valuable comments.

[3] Interested readers can access *AliMe Assist* through the Taobao App by following the path "(My Taobao)→(My AliMe)".

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *In Proceedings of ICLR 2015* .

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*. pages 1724–1734.

Matthew Henderson. 2015. Machine learning for dialog state tracking: A review. In *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*.

Charles Lee Isbell, Jr. Michael Kearns, Dave Kormann, Satinder Singh, and Peter Stone. 2000. Cobot in lambdamoo: A social statistics agent. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. AAAI Press, pages 36–41.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *CoRR* abs/1412.2007.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. *An information retrieval approach to short text conversation*. arXiv preprint.

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

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2015. Multi-domain dialog state tracking using recurrent neural networks. *CoRR* abs/1506.07190.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of EMNLP 2011*. pages 583–593.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *FTIR* 3(4):333–389.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *the Proceedings of the Thirtieth AAAI 2016*. pages 3776–3784.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *the Proceedings of ACL 2015*. pages 1577–1586.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval- and generation-based dialog systems. volume abs/1610.07149.

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 *the Proceedings of NAACL HLT 2015*. pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *the Proceedings of NIPS 2014*. pages 3104–3112.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *ICML Deep Learning Workshop 2015* CoRR,abs/1506.05869.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *the Proceedings of the SIGDIAL 2013*. pages 423–432.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. *A network-based end-to-end trainable task-oriented dialogue system*. arXiv preprint.

Jason Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. Technical report.

Rui Yan, Yiping Song, and Hua Wu. 2016a. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In Proceedings of SIGIR '16, pages 55–64.

Zhao Yan, Nan Duan, Jun-Wei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016b. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *the Proceedings of ACL 2016*.