Retrieval Based Response Letter Generation For a Customer Care Setting

Biplob Biswas[1], Renhao Cui[2], and Rajiv Ramnath[1]
[1]Department of Computer Science and Engineering, The Ohio State University
[2]Emplifi
{biswas.102, ramnath.6}@osu.edu, renhao.cui@emplifi.io

Abstract

Letter-like communications (such as email) are a major means of customer relationship management within customer-facing organizations. These communications are initiated on a channel by requests from customers and then responded to by the organization on the same channel. For decades, the job has almost entirely been conducted by human agents who attempt to provide the most appropriate reaction to the request. Rules have been made to standardize the overall customer service process and make sure the customers receive professional responses. Recent progress in natural language processing has made it possible to automate response generation. However, the diversity and open nature of customer queries and the lack of structured knowledge bases make this task even more challenging than typical task-oriented language generation tasks. Keeping those obstacles in mind, we propose a deep-learning based response letter generation framework that attempts to retrieve knowledge from historical responses and utilize it to generate an appropriate reply. Our model uses data augmentation to address the insufficiency of query-response pairs and employs a ranking mechanism to choose the best response from multiple potential options. We show that our technique outperforms the baselines by significant margins while producing consistent and informative responses.

1 Introduction

In modern business operations, customer care services are essential to support customers needing product information, making complaints, and in general, positively addressing their expectations. This support service plays a vital role in ensuring a good customer experience and is a key factor in developing goodwill. While non-specific, general knowledge about a product can now be conveniently retrieved through a web search, the exchange of specific information naturally entails a conversation between the customer and an agent who represents the organization. Traditionally, this task has been carried out by a trained human through chat or email exchanges. However, doing this manually at scale takes an enormous human effort. The process is time-consuming, labor-intensive and error-prone given the massive volume and diversity of customer queries.

Automation of the response generation process can go a long way toward solving this problem. Unfortunately, the rule-based systems that are in existence today struggle to capture the linguistic complexity of typical communications.

Recently, the advent of transformer-based pre-trained language models such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020) has brought about substantial progress in understanding and generating fluent text. Nevertheless, task-oriented dialog (TOD), a process that aims to assist a user to complete a certain task through response generation, is yet to be mastered due to the challenges in producing text that is informative and relevant to the prompt (Zhang et al., 2019; Ko et al., 2019). The task is especially difficult because of the scarcity of annotated datasets needed to train a supervised model.

Furthermore, creating an effective customer feedback system has additional challenges. In the existing and very popular TOD datasets (Rastogi et al., 2020; Wu, 2019), user utterances are usually fact-finding queries annotated with slot labels (e.g. Query: ‘Find a park near area51’, Slot-Values: {destination: ‘park’, close_to: ‘area51’}). A TOD framework (Young, 2000) first identifies the slots and then uses the slot-value pairs to retrieve facts from a knowledge base to reply. Unfortunately, these frameworks do not realistically address customer-care automation tasks where neither is the user prompt labeled with slot tags nor is there a knowledge base with relevant facts. Fur-

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 168 - 175
July 10-15, 2022 ©2022 Association for Computational Linguistics

thermore, a user prompt may not be limited to an inquiry about facts but may also include a complaint, suggestion, compliment, request, etc.

In this work, we factor in these challenges and present a response generation framework that automatically produces and ranks response letters addressing customers' queries or feedback, with a minimally annotated dataset. The contribution of this work is in two areas:

- We propose a retrieve and refine (Weston et al., 2018) based response generation model that is robust, efficient, and generalizable. A retrieval model fetches required knowledge from previous customer-agent letter exchanges and a generator refines the retrieved information to produce a coherent response tailored to the current context. By using historical knowledge retrieval, the model not only circumvents the requirement for an explicit knowledge base or slot-labeled dataset but can also augment and extend such datasets to enable a more diverse generation. In other words, we offer a practical solution.

- A Maximum Mutual Information (MMI) (Li et al., 2016; Zhang et al., 2018) driven approach to rank responses according to their relevance to the query. We also show that the model's loss function itself can indicate the MMI and save us the time and effort of developing yet another "backward" model (Zhang et al., 2020).

Our response generation framework is to be deployed in production as a part of an evolving pipeline for automating the customer service process. It will initially serve as a suggestion system to collect feedback from real human agents and is then expected to progress to enabling increased levels of automation.

2 Dataset

We use two proprietary datasets from the restaurant and adhesive tape industry, named ***DineCare*** (DC) and ***TapeTech*** (TT), consisting of 8448 and 14938 unique email exchanges between customers and agents respectively. Each exchange contains a case id, product and reason codes of the service, the customer query letter, and the human agent's response letter. Examples are included in Table 3. Both product codes and reason codes are alphanumeric strings defined by the corresponding business and may lack textual description. The reason code stands for the type of customer query; therefore we observe similar responses across queries having the same reason code.

DineCare

Items	Train	Validation	Test	All
Samples	5491	1267	1690	8448
Unique Product Code	257	99	138	308
Unique Reason Code	245	168	185	274
Mean Query Token	54	56	55	55
Mean Response Token	48	47	46	47

TapeTech

Items	Train	Validation	Test	All
Samples	8962	2988	2988	14938
Unique Product Code	656	393	371	851
Unique Reason Code	273	211	219	293
Mean Query Token	49	51	50	49
Mean Response Token	74	76	75	75

Table 1: The statistics for the data samples of the DineCare and TapeTech dataset.

Data Preparation We mask specific personally identifiable or proprietary information elements such as names, email addresses, phone numbers, prices, franchise names, and dates in our dataset with their corresponding generic tokens ("X-email", "X-phone" etc.). This serves two purposes. Firstly, this anonymization protects the privacy of the customers and the organization. Secondly, it forces the model to learn from and generate generic tags while avoiding noise in the form of irrelevant details such as specific names and values.

The average token count of queries in ***DineCare*** and ***TapeTech*** dataset are 55 and 49 respectively. For responses, the mean token counts of the corresponding datasets are 47 and 75. In both cases of query and response, these values are higher than that of a typical live chat. In terms of the type of the customer letters, 52% of *DC* letters are complaints, 33% are inquiries and the rest are of miscellaneous categories. Furthermore, they involve a diverse set of products (308 in DineCare, 851 in TapeTech) and reasons (274 in DineCare, 293 in TapeTech).

Each dataset is randomly divided into training (≈60%), validation (≈20%) and test (≈20%) sets. A summary of the dataset is presented in Table 1. Although the product and reason codes have a long-tailed distribution, samples involving them are present proportionally in each split. However, a fraction (2.5% in DineCare, 3.6% in TapeTech) of the test set contains product or reason codes that are absent in the training set. We retained them because their use helps us to understand the model's resilience in the event of unknown scenarios.

3 Framework

In this section, we first describe a potential baseline approach for response generation and then the retrieval-guided response generation framework.

3.1 Neural Response Generation

We utilize a pre-trained causal language model, GPT-2 (Radford et al., 2019), to train our baseline response generation model. We approach our goal of producing a response for a customer query as a conditional text generation task and hence adopt the pre-trained GPT-2 model to tune the parameters. Given a customer query letter with a token sequence: $q = (x_1, x_2, ..., x_m)$ where $x_i \in V$ for vocabulary V, product code: $d \in D$ and reason code: $s \in S$, the objective of our model is to generate response token sequence: $r = (y_1, y_2, ..., y_n)$ from the same vocabulary, i.e. $y_i \in V$. To this end, we first formulate the conditional probability of the response token sequence by factorizing the distribution using the chain rule:

$$p(r|q, d, s) = \prod_{i=1}^{n} p(y_i|y_{1:i-1}, q, d, s) \qquad (1)$$

Equation 2 gives us the negative log-likelihood $\mathcal{L}(E)$ that we want to minimize over a dataset E with parameters θ.

$$\mathcal{L}(E) = -\sum_{j=1}^{|E|} \log p_\theta(y_i|y_{1:i-1}, q, d, s) \qquad (2)$$

3.2 History Guided Generation

The GPT-2 baseline model (above) lacks access to factual information while responding to a query. Therefore, it tends to make up a safe or hallucinated reply. For instance, in response to a customer's question regarding a restaurant's service availability, the baseline model is seen to generate "don't know" or "open" although the dataset indicates its closure. To address this issue, a *Retrieve and Refine **(RetRef)*** (Weston et al., 2018) mechanism is employed. The idea is to retrieve valid responses for similar queries used in the recent past and utilize those responses in addition to the query to generate a refined and coherent response.

We split the whole task into three steps: 1. Knowledge Retrieval, 2. Response Generation and 3. Response Ranking. The framework is depicted in Figure 1 and detailed in the following subsections.

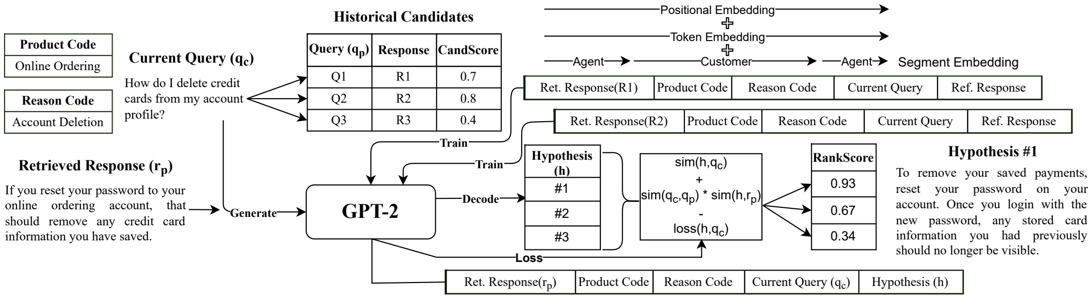
3.2.1 Knowledge Retrieval

Given a current customer query (q_c), knowledge can be extracted from agents' past responses (r_p) to similar past queries (q_p). To this end, we first select past conversations having the same reason code as the current one. This intuitively works as candidate generation and reduces our search space for potential knowledge. Then we assign a candidate score, $c = sim(q_c, q_p) + bleu(q_c, q_p)$ to these past query-response pairs where $bleu$ is BLEU-1 score between the corresponding queries and $sim(q_c, q_p) = \cos(E_{q_c}, E_{q_p})$, is cosine similarity between the embedding of the current and of the past query respectively. The embeddings are obtained using sentence-transformer (Reimers and Gurevych, 2019) and can be pre-computed to make the retrieval fast. While training, the similarity between the corresponding responses, $sim(r_c, r_p)$ is also added with a weight, $\gamma < 1$ to ensure that the model finds a relation between them to transfer knowledge (note that this response similarity is not used during testing as the reference response is unknown then). For training we choose (all) responses from candidate pairs that have $c > \tau$, where τ is a hyperparameter. Additional potential candidate responses were used to augment the training instances. We explain their use in the next section.

3.2.2 Response Generation

We use the same GPT-2 baseline model for retrieval-based training and generation. However, for the input to this model, a retrieved response is appended to the beginning of the current query (separated by a special token) as shown in the top-right corner of Figure 1. The objective is to teach the model to generate the reference response utilizing the retrieved knowledge in addition to the query.

Since only one retrieved response is used at a time, having more than one above the threshold ($c > \tau$) allows us to create more training instances with the same query-reference response pair (see Figure 1). In the absence of suitable retrievals, the reference itself is used as a retrieved response to make the model mirror the fetched response.



Query (q_p)	Response	CandScore
Q1	R1	0.7
Q2	R2	0.8
Q3	R3	0.4

Figure 1: The proposed retrieval based response generation framework.

This technique works like teacher forcing and is intended to avoid ignoring retrievals (as reported in Roller et al. (2021)). In the event of such a scenario during testing, we resort to the baseline model for a generation. We term this mix-model approach ***hybrid*** generation.

Moving on to prompt formation, it has three segments: the retrieved responses from the agent, the current query from the customer followed by the reference response from the agent. Even though the source request and response components are separated by a special token, a model does not have an idea of the author of a token. To address this, we add a segment embedding to the token embedding. The way positional embedding helps the model understand the relative position of the tokens, a segment embedding of the corresponding author is similarly reported to add more meaning to the model (Wolf et al., 2020).

3.3 Response Ranking

Even with the state-of-the-art decoding mechanisms, neural text generation is known to suffer from blandness or inconsistency (Zhang et al., 2019; Ko et al., 2019). Hence, we generate multiple responses using different sampling methods (e.g. top-k, nucleus, etc.) and employ a ranking mechanism to measure the context-awareness of generated responses (by evaluating them as hypotheses of the source query).

Such a hypothesis would indicate stronger correspondence when its probability of producing the query i.e. $p(query|hypothesis)$ is higher. To measure this probability, following the work of DialoGPT (Zhang et al., 2020), we trained an inverse model that considers the reference response as the input and the customer letter as the output. The loss of the model for a pair of queries and hypotheses was used to estimate the $p(query|hypothesis)$ score. The intuition is that a trivial and safe response is likely to appear frequently in different contexts and would usually contain less specific words, causing the inverse model to struggle to retrieve the source query from it, thus resulting in a higher loss. However, comparing the original model loss ($loss(h, q_c)$) with the inverse model loss for each query(q_c)-hypothesis(h) pair, we interestingly found a very high correlation between them. It indicates that we can avoid training an additional inverse model and perform the ranking process using forward loss only.

The final rank score of a hypothesis is computed using the following formula:

$$r_h = sim(h, q_c) - loss(h, q_c) \\ + sim(q_c, q_p) * sim(h, r_p) \quad (3)$$

where $sim(h, q_c)$ indicates similarity between query and hypothesis and $sim(q_c, q_p) * sim(h, r_p)$ takes into account the correspondence between retrieved response(r_p) and hypothesis(h) weighted by the query similarity($sim(q_c, q_p)$). The rationale behind the last product is: a generation that retains knowledge from a good retrieval is likely to offer a better response. Consequently, a higher rank score is expected to indicate better hypothesis quality.

4 Experiment Details

We used a small GPT-2 model of 124M parameters provided by Huggingface (Wolf et al., 2020). Grid search was used to tune the hyper-parameters to the following set of optimal values: {Weight Decay: 0.1, Warm-up steps: 1E2, Gradient Accumulation Steps: 16, Learning rate: 5E-4, Dropout rate: 0.1, Epoch: 5, optimizer: Adam, $\gamma = 0.4$, $\tau = 0.6$}. The training took around 1 hour for the DC dataset and 2.5 hours for the TT dataset on an NVIDIA Tesla V100-SXM2-16GB (GPU device). After training, 4 responses were generated for each query using 4 decoding combinations: 1. (top-k with temperature), 2. (top-p with temperature),

	DineCare						TapeTech					
Method	**S**	**B**	**N**	**M**	**R**	**C**	**S**	**B**	**N**	**M**	**R**	**C**
Baseline	0.59	0.36	4.33	0.25	0.39	2.22	0.84	0.56	6.51	0.38	0.66	3.59
Retrieve Only	0.63	0.34	4.27	0.24	0.38	2.11	0.84	0.53	6.45	0.36	0.64	3.47
RetRef	0.64	0.39	4.57	0.27	0.43	2.50	0.85	0.58	6.75	0.39	0.67	3.82
RetRef+Rank	**0.68±0.02**	**0.40±0.01**	**4.72±0.06**	**0.30±0.02**	**0.47±0.03**	**2.83±0.07**	**0.86±0.01**	**0.61±0.03**	**6.96±0.11**	**0.41±0.01**	**0.69±0.02**	**4.00±0.14**
Hybrid	0.62	0.39	4.51	0.27	0.42	2.43	0.84	0.56	6.61	0.38	0.65	3.58
Hybrid+Rank	0.67	0.40	4.62	0.30	0.46	2.73	0.85	0.59	6.79	0.40	0.67	3.72

Table 2: Test set results of the proposed response generation model on DineCare and TapeTech dataset. Baseline (§3.1) refers to fine-tuned GPT-2 model without knowledge retrieval. Automatic scoring metrics are: Average-SentenceSimilarity (**S**), BLEU-4 (**B**), NIST (**N**), METEOR (**M**), ROUGE-L (**R**) and CIDEr (**C**). Our best model's (*RetRef+Rank*) scores are averaged over 5 runs and have low standard deviation.

3. (top-k,top-p) and 4. (top-k,top-p with temperature) where k=20, p=0.8 and temperature=0.7. For preprocessing and evaluation, we use NLTK and nlg-eval (Sharma et al., 2017).

5 Evaluation

5.1 Retrieval Performance

A higher similarity between retrieved and reference response indicates a better retrieval. Our analysis finds that for 46% of 8382 queries, our retrieval model fetches at least one reference-like (similarity > 0.9) historical response within the top-10 candidates of each retrieval. Within top-5 and top-1 candidates, a retrieval with the above similarity is found in 38% and 21% cases respectively. Our manual evaluation on randomly sampled retrievals finds 49% of the retrieved responses suitable for generation and 20% as somewhat relevant. In the case of retrieval speed, with the pre-computed embeddings of 8448 records, it takes around 70 milliseconds using the aforementioned hardware (in §4) to fetch top-10 candidates of a query from the entire set.

5.2 Generation Quality

The automatic evaluation of all the methods is conducted on the held-out test set with the optimal hyper-parameter setting. The scores are listed in Table 2. Once again, note that the baseline model (§3.1) is a fine-tuned GPT-2 without retrieval and does not employ response ranking. For *Retrieve Only* method we consider only the fetched historical response (without refinement) as a hypothesis. For baseline, *RetRef*, and *Hybrid* method, we consider single hypothesis per query that is produced using the 4^{th} decoding setup (top-k,top-p with temperature) because its corpus-level score is better than other combinations. In ranking enforced versions of *RetRef* and *Hybrid*, for each query, we generate multiple responses using the aforementioned four decoding combinations (§4) and pick one with the highest rank score as the hypothesis for evaluation.

To assess our model, we utilize commonly used metrics such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015) and Average Sentence Similarity. The last metric is our measure of semantic similarity between reference response and hypothesis. For this measure, we use sentence-BERT (Reimers and Gurevych, 2019), a trained Siamese BERT-network to encode a reference and a hypothesis, and then calculate the cosine similarity of the resulting embeddings. The final similarity score is the mean value over the test set.

For both the datasets, our proposed retrieval-based response generation model (*RetRef+Rank*) outperforms all other baselines in all the metrics. Specifically, for *DineCare*, it achieves an improvement over the finetuned GPT-2 baseline model by 15.3% in similarity, 11.1% in BLEU-4, 9% in NIST, 20% in METEOR, 20.5% in Rouge-L and 27.5% in CIDEr. The performance increase with *TapeTech* dataset are 2.4%, 8.9%, 6.9%, 7.9%, 4.5% and 11.4% for the same order of metrics. Knowledge retrieval plays a key role in this rise. On the other hand, without refinement, *Retrieve only* approach yields the worst scores. *Hybrid* version can switch between the baseline and *RetRef* based on the availability of suitable retrieved responses. It was evaluated by including such test cases, and outperformed the baseline model by a significant margin across all metrics and datasets.

To measure the extent to which our model incorporates a retrieved knowledge in the generation, we

Label	**Source**	**Text**
DC#1 Inquiry	**Metadata**	Product Code: SERVICE RELATED, Reason Code: GC - PROMO - I
	Input	I just signed up for X-org this morning and have not received my coupon for free pancakes yet. When will I receive it?
	Reference	After signing up, it may take up to 24 hours to receive your initial offer.
	Our model	The day you sign up to create an account, you will be receiving your first email within 24 hours.
DC#2 Suggestion	**Metadata**	Product Code: SERVICE RELATED, Reason Code: OLO - SUGGESTION
	Input	I love your bourbon street foods. ... when ordering online it does not give much options to customize. ... Please try and correct this.
	Reference	We are glad that you suggested these changes to our site! This feedback has been shared with our X-org Online Ordering team.
	Our model	We love hearing from our passionate fans about ways to improve our online ordering. Comments like yours are made available to our Digital Operations team for consideration in future releases. Thank you for your suggestion.
DC#3 Complaint	**Metadata**	Product Code: SERVICE RELATED, Reason Code: GCARD - ORDER - C
	Input	I ordered the X-money gift card today but the 2 X-money bonus cards are not on the invoice or included.
	Reference	We are sorry you did not receive your 2 bonus cards. To make up for this, we will be sending you a X-money digital gift card that you will receive in a separate e-mail.
	Our model	We are sorry to hear you did not receive your bonus card. We will be sending you a X-money digital gift card that you will use on your next visit. You should receive it in a separate e-mail.
TT#4 Inquiry	**Metadata**	Product Code: 31714, Reason Code: USAPPLIC
	Input	I am looking for a duct tape to repair my pontoon boat cover. What tape would you suggest and where can I purchase it from?
	Reference	For your application, we would recommend our T-Rex Waterproof Tape. It is available at X-url; here is the link: X-url.
	Our model	T-Rex Waterproofing Tape would probably be best. You may also find it online on X-url, X-url.
TT#5 Inquiry	**Metadata**	Product Code: L134-P5, Reason Code: GNINGRED
	Input	Can you tell me if your Smooth Top Easy Liner Brand Shelf Liner - Grey Damask contains PVC?
	Reference	The smooth top easy liner is made of PVC coated polyester. Thank you for your inquiry.
	Our model	Our shelf liner is made of PVC coated polyester. Thank you for your inquiry.

Table 3: Sample response generation using our *RetRef+Rank* model

leverage previous work (Weston et al., 2018). Table 4 reports the word-overlap between generated and retrieved responses. For baseline method, overlap is computed between generated and reference response. The results show that our *RetRef+Rank* model retained >70% words from retrieval in 51% and 57% of the test generation of *DineCare* and *TapeTech* dataset respectively. This is a clear improvement over the baseline and the basic *RetRef* model which shows such overlap less frequently.

	DineCare			**TapeTech**		
Method	**<30%**	**30-70%**	**>70%**	**<30%**	**30-70%**	**>70%**
Baseline	48%	12%	40%	23%	27%	50%
Retref	42%	12%	46%	21%	25%	54%
RetRef+Rank	34%	15%	51%	17%	26%	57%

Table 4: Word overlap between retrieved and generated response.

Human Evaluation Three experts in the field manually assessed the relevance and informativeness of small-scale, randomly selected hypotheses. Relevance measures if a generated response is based on the corresponding product and reason whereas informativeness checks for its information consistency with respect to the reference response (Both scored out of 5). The result shows that responses produced by our *RetRef+Rank* model yield roughly 9% higher relevance (4.05 for DineCare, 4.49 for TapeTech) and 12% better informativeness (3.75 for DineCare, 4.24 for TapeTech) score than the baseline model, and for both the datasets.

5.3 Ablation Study

Apart from the inclusion of retrieved knowledge, two other notable contributors to the performance of the framework are data augmentation and response ranking. Our experiments reveal that the creation of more training instances with multiple candidate responses increases the automatic score by 12% in BLEU-4, 6% in CIDEr, and roughly 2% in other metrics. The role of ranking is also evident from the significant raise of *RetRef+Rank* and *Hybrid+Rank* model score from their base version as shown in Table 2. This can be attributed to the ranker's policy to penalize irrelevant generation while favoring the one that integrates quality retrieval.

5.4 Generation Examples and Discussion

Table 3 shows a few randomly selected generations from both datasets. It suggests that our model's responses are aligned with the type of the customer letter. For instance, letters of type inquiry (DC#1, TT#4, TT#5), suggestion (DC#2), and complaint (DC#3) are responded to accordingly with information, appreciation, and clarification. Secondly, having historical knowledge, our model is not only capable of producing an informed response but also refines that according to the query (DC#1). A few limitations of our model include its inability to verify time-sensitive historical information and handling multiple questions in the same letter. Additionally, any automated offer of a coupon (As shown in DC#3) or other follow-up commitment may put the company at risk. To resolve these issues, a risk or confidence measuring system can be introduced based on which human inspection may be sought before a response is dispatched. We leave this as future work.

6 Related Work

Research in machine-generated response systems originated at least four decades ago. At the end of the last century, Young (2000) introduced a concept of utterance recognizer and response generator for task-oriented dialog (TOD) systems. The past few years have witnessed several response generation models, particularly using neural approaches to conversational AI. Recently, combining the idea of GPT (Radford et al., 2018) and transfer-learning based training scheme, Wolf et al. (2020) produced improved dialog systems. Similarly, Zhang et al. (2020) presented DialoGPT, a tunable large-scale conversational response generation model based on GPT-2 (Radford et al., 2019). For TOD system, Kale and Rastogi (2020) and Du et al. (2020) proposed schema and template guided generation respectively which use slot-value tagged knowledge representations as input. Lately, several works (Weston et al., 2018; Roller et al., 2021; Kim et al., 2020; Lewis et al., 2020) have put forward a retrieve and refine approach to combine plain-text knowledge in conversational response generation. These works have inspired us to adopt similar generation ideas for our task.

7 Conclusion

The study proposes a neural response generation framework to reduce human labor in a real-world customer care setting, where a structured knowledge base is scarce. Our framework extracts knowledge from historical records of conversations to generate an informative response. Our evaluation shows the efficacy of the ranking system and provides evidence for the operational applicability of the framework. We plan to extend the framework with a response validation module for further improvement.

Acknowledgements

We would like to thank all the reviewers for their helpful comments. We are also grateful to Emplifi for their generous support of this research.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of HLTR*, HLT '02, page 138–145.

Yuheng Du, Shereen Oraby, Vittorio Perera, Minmin Shen, Anjali Narayan-Chen, Tagyoung Chung, Anushree Venkatesh, and Dilek Hakkani-Tur. 2020. Schema-guided natural language generation. In *Proceedings of the 13th International Conference on NLG*, pages 283–295.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of EMNLP*, pages 6505–6520, Online. ACL.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289, 1st virtual meeting. Association for Computational Linguistics.

Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. 2019. Linguistically-informed specificity and semantic plausibility for dialogue generation. In *Proceedings of NAACL-HLT*, pages 3456–3466.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of SMT*, StatMT '07, page 228–231.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

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-HLT*, pages 110–119. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. ACL.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, Pennsylvania, USA. ACL.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at dstc8.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*, pages 38–45.

Chien-Sheng Wu. 2019. Learning to memorize in neural task-oriented dialogue systems.

Steve Young. 2000. Probabilistic methods in spoken dialogue systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 358.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of NeurIPS*, NIPS'18, page 1815–1825.

Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Consistent dialogue generation with self-supervised feature learning. *arXiv preprint arXiv:1903.05759*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of ACL*, pages 270–278.