Retrieve and Refine: Improved Sequence Generation Models For Dialogue

Jason Weston, Emily Dinan and Alexander H. Miller
Facebook AI Research
jase@fb.com, edinan@fb.com, ahm@fb.com

Abstract

Sequence generation models for dialogue are known to have several problems: they tend to produce short, generic sentences that are uninformative and unengaging. Retrieval models on the other hand can surface interesting responses, but are restricted to the given retrieval set leading to erroneous replies that cannot be tuned to the specific context. In this work we develop a model that combines the two approaches to avoid both their deficiencies: first retrieve a response and then refine it – the final sequence generator treating the retrieval as additional context. We show on the recent CONvAI2 challenge task our approach produces responses superior to both standard retrieval and generation models in human evaluations.

1 Introduction

Sequence generation models like Seq2Seq (Sutskever et al., 2014) are increasingly popular for tasks such as machine translation (MT) and summarization, where generation is suitably constrained by the source sentence. However, obtaining good performance on dialogue tasks, where the context still allows many interpretations, remains an open problem despite much recent work (Serban et al., 2016). Several authors report the issue that they produce short, generic sentences containing frequent words – the so-called “I don’t know” problem – as that response can work as a reply in many instances, but is uninformative and unengaging. Retrieval models (Ji et al., 2014) do not have this problem, but instead either produce engaging responses or else completely erroneous ones which they cannot tune to the specific context, as they can only produce a valid reply if it is in the retrieval set.

In this work we propose a Retrieve and Refine model to gain the advantages of both methods, and avoid both their disadvantages. Models that produce an initial prediction and then refine it are growing in traction in NLP. They have been used in MT and summarization either for refinement of initial predictions (Junczys-Dowmunt and Grundkiewicz, 2017; Niehues et al., 2016; Novak et al., 2016; Xia et al., 2017; Grangier and Auli, 2017) or combining with retrieval (Gu et al., 2017; Cao et al., 2018), as well as for sentence correction or refinement without context (Guu et al., 2017; Schmaltz et al., 2017). There is little work in applying these methods to dialogue; one work we are aware of has been done concurrently with ours is Pandey et al. (2018). The usefulness of our approach is shown with detailed experiments on the ConvAI2 dataset\footnote{http://convai.io/} which is a chit-chat task to get to know the other speaker’s profile, obtaining generations superior to both retrieval and sequence generation models in human evaluations.

2 Retrieve and Refine

The model we propose in this work is remarkably straight-forward: we take a standard generative model and concatenate the output of a retrieval model to its usual input, and then generate as usual, training the model under this setting.

For the generator, we use a standard Seq2Seq model: a 2-layer LSTM with attention. For the retriever, we use the Key-Value Memory Network (Miller et al., 2016) already shown to perform well for this dataset (Zhang et al., 2018), which attends over the dialogue history, to learn input and candidate retrieval embeddings that match using cosine
similarity. The top scoring utterance is provided as input to our Seq2Seq model in order to refine it, prepended with a special separator token. For both models we use the code available in ParlAI\(^2\). At test time the retriever retrieves candidates from the training set.

To train our model we first precompute the retrieval result for every dialogue turn in the training set, but instead of using the top ranking results we rerank the top 100 predictions of each by their similarity to the label (in embedding space). Following Guu et al. (2017) this should help avoid the problem of the refinement being too far away from the original retrieval. We then append the chosen utterances to the input sequences used to train Seq2Seq. We refer to our model as RetrieveNRefine, or RetNRef for short. We also consider two variants of the model in the following that we found improve the results.

**Use Retriever More** In our vanilla model, we noticed there was not enough attention being paid to the retrieval utterance by the generator. As the input to Seq2Seq is the dialogue history concatenated with the retrieval utterance, truncating the history is one way to pay more attention to the retrieval. In particular for the ConvAI2 dataset we clip the initial profile sentences at the start of the dialogue, forcing the model to more strongly rely on the retriever which still has them.\(^3\) We refer to this modification as RetrieveNRefine\(^+\).

**Fix Retrieval Copy Errors** Our model learns to sometimes ignore the retrieval (when it is bad), sometimes use it partially, and other times simply copy it. However, when it is mostly copied but only changes a word or two, we observed it made mistakes more often than not, leading to less meaningful utterances. We thus also consider a variant that exactly copies the retrieval if the model generates with large word overlap (we chose >60%). Otherwise, we leave the generation untouched.\(^4\) We refer to this as RetrieveNRefine\(^++\).

| RetNRef Retrieval Method          | PPL |
|-----------------------------------|-----|
| None (Vanilla Seq2Seq)            | 31.4|
| Random label                      | 32.0|
| Memory Network                    | 31.8|
| True label’s neighbor             | 25.9|
| True label                        | 9.2 |

Table 1: Perplexity on the ConvAI2 task test set with different types of retriever for RetNRef, see text.

3 Experiments

We conduct experiments on the recent ConvAI2 challenge dataset which uses a modified version of the PersonaChat dataset (Zhang et al., 2018) (larger, and with different processing). The dataset consists of conversations between crowdworkers who were randomly paired and asked to act the part of a given persona (randomly assigned from 1155 possible personas, created by another set of workers), chat naturally, and get to know each other during the conversation. There are around 160,000 utterances in around 11,000 dialogues, with 2000 dialogues for validation and test, which use non-overlapping personas.

3.1 Automatic Evaluation and Analysis

**Perplexity** Dialogue is known to be notoriously hard to evaluate with automated metrics (Liu et al., 2016). In contrast to machine translation, there is much less constraint on the output with many valid answers with little word overlap, e.g. there are many answers to “what are you doing tonight?”\(^5\). Nevertheless many recent papers report perplexity results in addition to human judgments. For the retrieve and refine case, perplexity evaluation is particularly flawed: if the retrieval points the model to a response that is very different but equally valid as the true response, the model might focus on refining that and get poor perplexity.

We therefore test our model by considering various types of retrieval methods: (i) the best performing existing retrieval model, the Memory Network approach from Zhang et al. (2018) (retrieving from the training set), (ii) a retriever that returns a random utterance from the training set, (iii) the true label given in the test set, and (iv) the closest nearest neighbor from the training set utterances to the true label, as measured by the embedding space of the Memory Network retriever model. While (iii) and (iv) cannot be used in a de-

\(^2\)http://parl.ai

\(^3\)Architectural changes might also deal with this issue, e.g. treating the two inputs as independent sources to do attention over, but we take the simplest possible approach here.

\(^4\)Other approaches might also help with this problem such as using an explicit copy mechanism or to use BPE tokenization (Fan et al., 2017), but we leave those for future work.
Table 2: Output sequence statistics for the methods. Seq2Seq generates shorter sentences with more common words than humans, which RetNRef alleviates.

| Method      | Word cnt | Char cnt | Rare Word % |
|-------------|----------|----------|-------------|
|             | <100     | <1k      |             |
| Seq2Seq     | 11.7     | 40.5     | 0.4% 5.8%   |
| RetNRef     | 11.8     | 40.4     | 1.1% 6.9%   |
| RetNRef+    | 12.1     | 45.0     | 1.7% 10.1%  |
| RetNRef++   | 12.7     | 48.1     | 2.3% 10.9%  |
| MemNet      | 13.1     | 54.5     | 4.0% 15.3%  |
| Human       | 13.0     | 54.6     | 3.0% 11.5%  |

Table 3: Word overlap between retrieved and generated utterances in RetNRef, and between Seq2Seq and the Memory Network retriever (first row).

| Method      | <30% | 30-60% | 60-80% | >80% |
|-------------|------|--------|--------|------|
| Seq2Seq     | 56%  | 34%    | 7%     | 3%   |
| RetNRef     | 41%  | 38%    | 13%    | 8%   |
| RetNRef+    | 26%  | 20%    | 12%    | 42%  |
| RetNRef++   | 26%  | 20%    | 0%     | 53%  |

Table 3 compares the word overlap between retrieved and generated sentences in the RetNRef variants in order to measure if RetNRef is either ignoring the retriever, or else paying too much attention to and copying it. As comparison, the first row also shows the overlap between the retriever and vanilla Seq2Seq which does not retrieve at all. The results show that RetNRef++ has >80% word overlap with the retriever output around half (53%) of the time, whereas Seq2Seq and RetNRef very rarely overlap with the retriever (3% and 8% of the time respectively have >80% overlap). This shows that our improved model RetNRef++ does use the retriever, but can also generate novel content when it wants to, which a standard retriever cannot.

3.2 Evaluation by Human Judgement Scores

Following the protocol in Zhang et al. (2018), we asked humans to conduct short dialogues with our models (100 dialogues each of 10-20 turns, so 600 dialogues in total), and then measure the engagingness, consistency, and fluency (all scored out of 5) as well as to try to detect the persona that the model is using, given the choice between that and a random persona. Consistency measures the coherence of the dialogue, e.g. “I have a dog” followed by “I have no pets” is not consistent.

The results are given in Table 4. They show engagingness scores superior to Seq2Seq for all RetNRef variants, and with RetNRef++ slightly outperforming the retriever which it conditions on. Importantly however, it maintains this performance whilst still being able to generate text which a retrieval model cannot. It also performs well in the other metrics, although like the Memory Network model, it is weaker at using the per-
| Method                  | Engagingness | Fluency   | Consistency | Persona |
|------------------------|--------------|-----------|-------------|---------|
| Seq2Seq (PPL)          | 2.70(1.17)   | 3.50(1.37)| 3.90(1.37)  | 0.90(0.29)|
| Seq2Seq (100 epochs)   | 2.76(1.15)   | 3.53(1.14)| 3.84(1.38)  | 0.85(0.35)|
| Memory Network         | 3.66(1.26)   | 3.83(1.26)| 3.61(1.36)  | 0.73(0.44)|
| RetrieveNRefine        | 2.94(1.26)   | 3.65(1.28)| 3.72(1.32)  | 0.90(0.30)|
| RetrieveNRefine+       | 3.50(1.33)   | 3.63(1.13)| 3.55(1.33)  | 0.71(0.45)|
| RetrieveNRefine++      | 3.80(1.18)   | 3.74(1.19)| 3.80(1.40)  | 0.65(0.47)|

Table 4: Human Evaluation scores for the models, scoring fluency, engagingness, consistency and persona detection, with standard deviation in parentheses. We consider engagingness to be the most important metric.

| Comparison (A vs. B) | Win Rate | A Wins | B Wins | Tie | p-value |
|----------------------|----------|--------|--------|-----|---------|
| RetrieveNRefine++ vs. Memory Network | 54.5% | 340 | 284 | 572 | 0.027 |
| RetrieveNRefine++ vs. Seq2Seq | 53.7% | 571 | 492 | 203 | 0.016 |
| RetrieveNRefine++ (retrieved) vs. Seq2Seq | 53.8% | 290 | 249 | 87 | |
| RetrieveNRefine++ (generated) vs. Seq2Seq | 53.6% | 281 | 243 | 116 | |
| RetrieveNRefine++ vs. Memory Network | 51.63% | 492 | 461 | 243 | |
| RetrieveNRefine++ vs. Human | 30.13% | 69 | 160 | 14 | |
| Seq2Seq vs. Human | 26.84% | 62 | 169 | 22 | |

Table 5: A/B testing by humans comparing model pairs, reporting win rates for A over B (excluding ties).

| Person 1 | Person 2 | Seq2Seq | MemNet | RetNRef++ |
|----------|----------|---------|--------|-----------|
| hey, george here, let's be friends | hi george, i'm regina, do you enjoy dance? | i do not but i do have a lot of time to dance | hello george, my name is pink | i do, i enjoy dance. |
| yeah, well, i wish you luck there, we bought a cute 2 story house. | nice, i don’t think i have ever left ohio. | i am sorry, i am so scared of twins. | i am from mississippi, just moved back here from austin texas | i am from tennessee, just moved here from austin texas. |
| drinking coffee? what time is it there? | so he works really late nights | thank you for that, you are what you believe? | took energy isn't physical, only spiritual | |
| 830 pm haha my dad is a construction worker | i am a big guy so i can help you | i am not sure what that means | do you do any sports? swimming helps me keep my energy up. | do you do any sports? swimming helps me keep my energy up. |
| yep, moonlight is amazing but northern light grab you too | oh wow, that is a long time | do you do any sports? swimming helps me keep my energy up. | |

Table 6: Example dialogues (left) where RetNRef++ outperforms MemNet, and (right) where RetNRef++ outperforms Seq2Seq – by either paying attention to the retriever or not. The first two rows are the previous two dialogue turns by Person 1 & 2, the following rows are the predictions for the next utterance of Person 1 by the various models.

Some example dialogues of the RetNRef++ model performing well (as scored by the evaluators) are shown in Tables 7 and 8. Longer sentences from the bot (person 2) with more nuanced entity information typically come from attending to the retriever, whereas the generator can also produce shorter replies independent of the retriever that fit the context well. There are still issues however, such as repeated phrases by the generator, and some tendency to copy the speaking partner’s phrases that could be improved.

3.3 A/B Testing by Humans

To compute a statistically more meaningful test, and to evaluate models more clearly against each other, we also tested models by pairing them against each other (different to the protocol previously used for this dataset). We took human-human paired dialogues from the test set, up to a randomly chosen turn, and then computed the response of two rival models. The resulting dialogue history and the two choices are given to a third annotator to evaluate and asked which model is best; they can also choose "unsure". We collected over 1000 such examples, with each conversation being...
Table 7: Example dialog between a human (Person 1) and the RetNRef++ model (Person 2). References to never eating vegetables, listening to audiobooks and speaking one Asian language and owning a Nissan Sentra appeared in the conditioned retrieved sentences, improving the overall engaging quality of the dialogue.

Table 8: Example dialog between a human (Person 1) and the RetNRef++ model (Person 2). References to Dr. Pepper and wishing to have a parrot appeared in the conditioned retrieved sentences, improving the overall engaging quality of the dialogue. There are still issues however, such as repeated phrases by the generator, failure to use the persona adequately, and some tendency to copy the speaking partner’s phrases that could be improved.

Evaluated twice. The results are given in Table 5. RetrieveNRefine obtains statistically significant wins over the retriever Memory Network model and the generator Seq2Seq model using a binomial two-tailed test, with win rates ~54%. Breaking down the wins between when RetNRef++ exactly copies the retrieval utterance vs. generates we see that it chooses them about equally, with wins about equal in both cases. This shows it can effectively learn when to choose the retrieval utterance (when it is good), and when to ignore it and generate instead (when it is bad). Table 6, which shows example outputs of our model, illustrates this.

RetNRef++ sometimes loses out when making small changes to the retrieved text, for example it made changes to “I once broke my nose trying to peak in on a jazz concert!” by replacing peak with glacier. Recall that RetNRef++ fixes this problem by exactly copying the retrieved text when there is insignificant word overlap with the generated text; as such, it has a correspondingly larger win rate against Memory Networks (54.5% versus 51.63%).

We also computed a small sample of A/B tests directly against humans rather than models, and again see the win rate is higher for RetNRef++.
quent words that ultimately are not engaging. Our RetNRef++ model has similar statistics to human utterances and provides more engaging conversations according to human judgments.

Future work should investigate improved ways to incorporate retrieval in generation, both avoiding the heuristics we used here to improve performance, and seeing if more sophisticated approaches than concatenation plus attention improve the results, for example by more clearly treating the inputs as independent sources, or training the models jointly.

References

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 152–161.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. arXiv preprint arXiv:1711.05217.

David Grangier and Michael Auli. 2017. Quickedit: Editing text & translations via simple delete actions. arXiv preprint arXiv:1711.04805.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2017. Search engine guided non-parametric neural machine translation. arXiv preprint arXiv:1705.07267.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. arXiv preprint arXiv:1709.08878.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. arXiv preprint arXiv:1706.04138.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael 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. arXiv preprint arXiv:1603.08023.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. arXiv preprint arXiv:1606.03126.