Weakly Supervised Extractive Summarization with Attention

Yingying Zhuang
Amazon Inc.
San Francisco, USA
yyzhuang@amazon.com

Yichao Lu
Amazon Inc.
Seattle, USA
yichaolu@amazon.com

Simi Wang
Amazon Inc.
Seattle, USA
simiwang@amazon.com

Abstract

Automatic summarization aims to extract important information from large amounts of textual data in order to create a shorter version of the original texts while preserving its information. Training traditional extractive summarization models relies heavily on human-engineered labels such as sentence-level annotations of summary-worthiness. However, in many use cases, such human-engineered labels do not exist and manually annotating thousands of documents for the purpose of training models may not be feasible. On the other hand, indirect signals for summarization are often available, such as agent actions for customer service dialogues, headlines for news articles, diagnosis for Electronic Health Records, etc. In this paper, we develop a general framework that generates extractive summarization as a byproduct of supervised learning tasks for indirect signals via the help of attention mechanism. We test our models on customer service dialogues and experimental results demonstrated that our models can reliably select informative sentences and words for automatic summarization.

1 Introduction

Automatic summarization systems are useful in many applications where they aim to create a concise version of large amounts of textual data. Much effort has been devoted to developing automatic summarization systems in recent years by using deep learning, such as sentence compression with LSTMs (Filippova et al., 2015), sentence summarization with neural attention networks (Rush et al., 2015; Chopra et al., 2016), text summarization using sequence-to-sequence RNNs (Nallapati et al., 2016), end-to-end dialogue description generation (Pan et al., 2018), and summarization with deep reinforced models (Paulus et al., 2017). These approaches fall into one of two broad categories: extractive and abstractive. Extractive summarization directly chooses and assembles sentences and words from the original texts as the summary. Abstractive summarization collects high quality information and a summary is written in a concise manner. Central to both approaches is the availability of labeled data for training. For extractive summarization, training requires sentences and words being labeled as summary-worthy or not. For abstractive summarization, training requires document-summary pairs where each document has a summary available to supervise the training of a model that can produce such summaries to capture the highlights of the document. However, such labeled data may not be available in many applications. On the other hand, indirect signals for summarization are often accessible. For example, for dialogues, the resulting actions contain valuable signals for summarization. For a news article, its category (such as Politics, Sports, Technology, Weather, etc.) and its title could provide guidance on summary key points. For an Electronic Health Record (EHR), the concluding diagnosis can be a very important piece of information.

In this paper, we develop a general framework for automatic extractive summarization for scenarios where there are no pre-labeled sentences/words for summary-worthiness but other indirect signals are available. Imagine how a human annotator reads texts and produces summaries. Instead of reading through the entire texts, memorizing all information, and then writing up a summary based on memories, humans often read the texts word by word, sentence by sentence, and highlight those containing key information such as the resulting actions for a dialogue, the category for a news article, the diagnosis for an EHR, etc. Our approach mimics this human behavior on picking out important content by using an attention mechanism (Bahdanau et al., 2014; Xu et al., 2015; Yang et al.,

Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 520–529

2016). The model structure composes a hierarchical attention network (Yang et al., 2016) as the reader, a downstream ancillary prediction task of the indirect signal, and an extractor for identifying important words and sentences for automatic extractive summarization. We use a dataset for the ancillary task in the learning process to prediction the indirect signals. During the learning process, the reader first composes a sequence of word vectors into a sentence vector for each sentence, and then composes the sequence of sentence vectors into a document vector. It has an attention layer on both word level and sentence level to score each word and each sentence in order to locate the region of focus during prediction of the indirect signals. These attention scores are then used to extract informative sentences and words for summarization, which is a byproduct of the supervised learning process for the indirect signals.

The most distinguishing feature of our approach from other extractive summarization approaches is that it does not require a large training corpus of documents with labels indicating which sentences or words should be in the summary. We test our models on customer service dialogues. The results show that the trained attention scores reflect a linguistically plausible representation of the importance for each sentence and word. Therefore, it provides an intuitive way to extract summarization in the absence of pre-labeled sentences or words for supervised learning.

The main contributions of this work are:

- We propose a novel framework for the task of extractive summarization in the absence of labeled data.
- Previous literatures have focused on evaluation for model performance of prediction. In our work, we perform in-depth evaluation of the attention scores' linguistic plausibility and compare them to human performance.

We first formally define the task in Section 2 and then introduce the general framework in Section 3. We describe our experiment settings in Section 4 and present our results in Section 5. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 Task Definition

Assume that the input texts consist of a sequence of L sentences. Sentence i contains a sequence of T_i words $(w_{i1}, ..., w_{iT_i})$. The task is to extract the l most informative sentences and the k_j most informative words for each of the selected sentence j. We first rank each sentence in the document based on its informativeness using attention scores, and then select a subset of the l most informative sentences (where $l \leq L$). We then rank the words in each of the selected l sentences and highlight the most informative k_j words for sentence j (where $k_j \leq T_j$).

3 Attention Based Extractive Summarization

In this section, we propose a novel architecture that generates extractive summarization as a byproduct of the supervised learning tasks for indirect signals via the help of attention mechanism. The sentences and words that have provided strong signals to the supervised learning tasks will naturally have high attention weights and become good candidates for the summary. We call this process weakly supervised extractive summarization with attention. The architecture consists of a Hierarchical Attention Network (HAN) (Yang et al., 2016) reader that composes the source texts into a continuous-space vector representation, a downstream ancillary prediction task that takes the representation and generates the output for the indirect signal, and an extractor for identifying important words and sentences for automatic extractive summarization.

3.1 Hierarchical Attention Network Reader

One of the key components of our summarization model is a hierarchical attention network reader that is structured by four elements: a word encoder, a word-level attention layer, a sentence encoder and a sentence-level attention layer. The reader first operates at the word level and reads the sequences of source texts as the input, leading to the acquisition of sentence-level representations. Next, it composes the sequence of sentence vectors into a document vector that is then used for our downstream supervised learning task. The two attention layers, a word-level attention layer and a sentence-level attention layer, locate the region of focus when acquiring the representation vectors. Those attention weights are learned based on the downstream supervised learning task and will be used for extracting summaries. The reader architecture is illustrated in Figure 1, panel A. It mimics the process of human annotation. When reading

a document, humans often distill the highlights by writing down the keywords and key sentences that give the document its context and generate the summary based on these highlighted words and sentences.

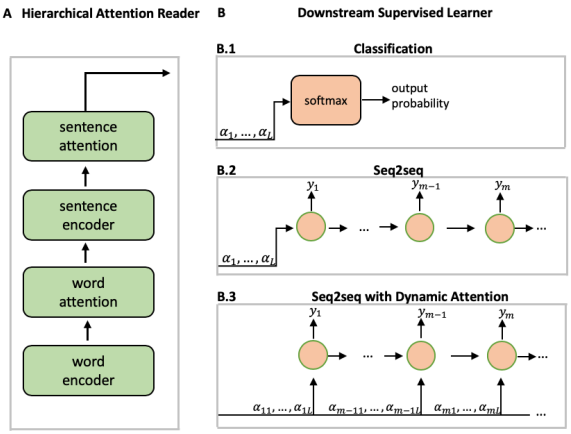

Figure 1: Model architecture. Panel A is the hierarchical attention network reader. Panel B.1 is the downstream supervised learner for a classification model. Panel B.2 is the downstream supervised learner for a seq2seq model. Panel B.3 is the downstream supervised learner for a seq2seq model with dynamic attention.

Assume the source texts contain L sentences and sentence i contains T_i words $(w_{i1}, ..., w_{iT_i})$. We let x_{it} denote the input vector for the t$^{\text{th}}$ word in the i$^{\text{th}}$ sentence. The word encoder maps $(x_{i1}, ..., x_{iT_i})$ to a sequence of word annotations $(h_{i1}, ..., h_{iT_i})$ using a recurrent neural network where $h_{it} = f(x_{it}, h_{it-1})$. Here f is some nonlinear function such as LSTM or GRU. For instance, Yang et al. (2016) used a bi-directional GRU (Chung et al., 2014; Cho et al., 2014b) for f where h_{it} is obtained by concatenating the forward hidden state $\overrightarrow{GRU}(x_{it})$ and the backward one $\overleftarrow{GRU}(x_{it})$: $h_{it} = \left[\overrightarrow{GRU}(x_{it}), \overleftarrow{GRU}(x_{it})\right]$ for $t = 1, ..., T_i$.

To apply an attention mechanism and extract important words in the sentence, we let $u_{it} = \tanh(W_w h_{it} + b_w)$; $\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)}$; $s_i = \sum_t \alpha_{it} h_{it}$ where $t = 1, ..., T_i$. Here u_w is the context vector at the word level. It is randomly initialized and jointly learned during the training process. Similarly, the sentence encoder maps the sequence of sentence vectors $(s_1, ..., s_L)$ to a sequence of sentence annotations $(h_1, ..., h_L)$ using a recurrent neural network. Then we use a sentence level context vector u_s to measure the importance of the sentences: $u_i = \tanh(W_s h_i + b_s)$; $\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)}$ where $i = 1, ..., L$. Similar to u_w, u_s is randomly initialized and jointly learned during the training process.

3.2 Downstream Supervised Learner

The downstream supervised learner is an ancillary prediction task for the indirect signal. A byproduct of this supervised learning task is the attention scores from the attention layers on both word level and sentence level. These attention scores reflect how strong of a signal they provide to the downstream supervised learning task, therefore, those with a high attention score naturally are good candidates for the summary. We present three types of downstream supervised learners, suitable for different formats of the indirect signal.

3.2.1 Classification

When the indirect signal is a categorical variable, the downstream supervised learner takes on the form of a classifier. The fixed-state vector representation of the source texts is calculated as $v = \sum_{i=1}^{L} \alpha_i h_i$. It can then be fed into a softmax layer to output a label for classification, as shown in Figure 1, panel B.1. For instance, the downstream ancillary task can be news category classification when the input texts are news articles, or disease classification when the inputs are EHRs.

3.2.2 Seq2seq

When the indirect signal is a sequential output, $(y_1, ..., y_M)$, the downstream supervised learner takes on the form of a recurrent neural network decoder whose initial hidden state is set to the fixed length representation v. The decoder is trained to generate the output sequence by predicting the next symbol y_m given the hidden state of the decoder at time m, which is computed by $h_m = f'(h_{m-1}, y_{m-1}, v)$. Choices for f' include LSTM, GRU, BiRNN, or any other variations of a recurrent neural network. The decoder architecture is shown in Figure 1, panel B.2.

One potential issue with this approach is that the use of the fixed-length vector v is a bottleneck in improving the performance of this encoder-decoder architecture. Cho et al. (2014a) showed that because all the necessary information of a source input needs to be compressed into the fixed-length vector, the performance of such architecture deteriorates as the length of input increases.

3.2.3 Seq2seq with Dynamic Attention

In order to address the bottleneck issue, we propose to add a dynamic attention (Bahdanau et al., 2014; Wu et al., 2016; Gehring et al., 2017) to the seq2seq decoder. The dynamic attention enables every position in the decoder to search through all positions in the input texts for important information, which are subsequently used to form the summarization.

As shown in Figure 1, panel B.3, at each step m the model attends over all sentence annotations $(h_1, ..., h_L)$ and calculates the hidden state as $h_m = f'(h_{m-1}, y_{m-1}, v_m)$ where v_m is computed as $v_m = \sum_{i=1}^{L} \alpha_{mi} h_i$, and $\alpha_{mi} = \frac{\exp(u_i^T u_{ms})}{\sum_i \exp(u_i^T u_{ms})}$. It should be noted that unlike the seq2seq task in Section 3.2.2, here a distinct set of attention scores $\alpha_{m1}, ..., \alpha_{mL}$ is calculated for each target word y_m. This is similar to the "encoder-decoder attention" layer in the transformer (Vaswani et al., 2017). The attention scores $\alpha_{m1}, ..., \alpha_{mL}$ show how important each sentence is in deciding the next state and generating the output word y_m. The context vector u_{ms} can be seen as a high-level representation of a fixed query "what are the informative sentences for the output y_m" for $m = 1, ..., M$, similar to those used in memory networks (Sukhbaatar et al., 2015; Kumar et al., 2016). Here u_{ms}s are randomly initialized and jointly learned during the training process.

3.3 Sentence and Word Extractor

For the classification ancillary task (presented in 3.2.1) and the seq2seq ancillary task (presented in 3.2.2), we rank each sentence by its corresponding α_i. For the seq2seq with dynamic attention ancillary task (presented in 3.2.3), because we calculate a distinct set of attention scores $\alpha_{m1}, ..., \alpha_{mL}$ for each target word y_m, we rank each sentence by the total attention scores received for all output words $(y_1, ..., y_M)$ where sentence j's total attention score is $\sum_{m=1}^{M} \alpha_{mj}$. Lastly, we rank each word within sentence j by its corresponding α_{jt}, where $t = 1, ..., T_j$. For extractive summarization, we select the l highest ranked sentences and the k_j highest ranked words for each of the selected sentence j.

4 Experimental Setup

In this paper, we conduct experiments to evaluate the plausibility of the attention scores to extract informative words and sentence for the use case of summarizing Amazon customer service dialogues. In a customer service context, dialogue summaries are especially useful in terms of providing contexts and highlights for contact transfers, escalations, and offline analysis. Our proposed approach addresses the issue of absence of labeled data and solves the problem for automatic extractive summarization. Table 1 gives an example of a customer agent dialogue from a customer service chat contact. A customer service contact summary typically contains information on what the customer's question or issue was, and what answer or solution the agent offered. Often labels indicating which sentences or words from the dialogue should be in the summary are not available while indirect signals on customer issue and agent action are stored and accessible. For example, for the customer service contact in Table 1, the customer issue code is "cancel order" and the agent action code is "full refund". Therefore, we can use the customer issue code and agent action code as the indirect signals for downstream ancillary prediction and obtain extractive summarization as a byproduct of the supervised learning tasks for indirect signals with the help of attention mechanism. Even though the customer issue code and the agent action code can already provide a high level summary themselves, they often lack some key information, such as the amount of the full refund, how long it takes for the customer to receive the refund, whether the refund is issued to a credit card or gift card, etc. Extractive summarization is especially valuable in this case because it can locate the sentences and words from the original dialogue that are summary-worthy and they contain much more comprehensive information than the customer issue codes and agent action codes themselves. Another advantage of this approach is that the model is flexible for extending to different applications. For instance, depending on the specific application of the summary, we may require information on customer sentiment to be included, in which case we can use the customer post contact survey responses as our indirect signals for downstream ancillary prediction. In the example in Table 1, the customer's post contact survey response is 5 out of 5 for satisfaction ratings.

4.1 Dataset

We collect transcripts between customers and agents from 1,681,809 anonymized Amazon customer service chat contacts. Word vocabulary size is 87,694 for customers and 113,446 for agents.

Agent:	Hello, how can I help you today?
Customer:	I accidentally bought a kindle book with 1 click and want the order to be cancelled.
Agent:	I see that there are 4 other e-books. Do you want to cancel all the items?
Customer:	Yes please.
Agent:	Thank you for confirming. Let me check with that, allow me a moment. Thank you for your patience. I've cancelled your order and issued a full refund.
Customer:	Fantastic. Thank you so much.
Agent:	You're welcome. Is there anything else I can assist you with today?
Customer:	No, that's it. You have been so helpful. I really appreciated it.
Agent:	My pleasure. Thank you for contacting Amazon. We hope to see you again. Have a great rest of your day.

Table 1: An Example of a Customer Agent Dialogue

Figure 2 demonstrates the distribution of number of sentences per dialogue and number of words per sentence. On average, each dialogue has 27 sentences in total, among which 11 are from the customer and 16 are from the agent. Each sentence has an average of 12 words. Agents also tend to speak longer sentences than the customer, where the average number of words in a sentence is 14 for agents while 9 for customers. We split the dataset into approximately 80% for training, 10% for validation, and 10% for testing to be used in the ancillary prediction task.

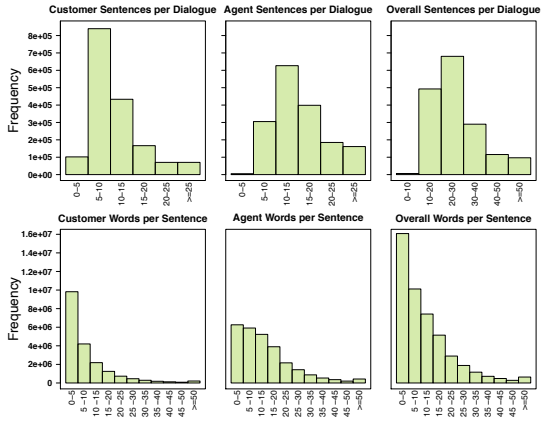

Figure 2: The first row demonstrates the distribution of number of sentences per dialogue in customer utterances, agent utterances, and customer utterances + agent utterances. The second row demonstrates the distribution of number of words per sentence in customer utterances, agent utterances, and customer utterances + agent utterances.

4.2 Evaluation

We focus on in-depth evaluation of the attention scores' linguistic plausibility to extract sentences and words for summarization. To create our evaluation data, we select a random sample of 1,000 customer service contacts from the testing dataset for manual annotation. We have two annotators, each of whom annotates 500 contacts, and a gold annotator who further validates the annotation to ensure quality and consistency between the two annotators. The annotators are asked to do the following:

1. For each contact, select the l most informative sentences from the dialogue and assemble them as the sentence-level summary for this contact. l is calculated as the ceiling of $20\% \times L$, which is the smallest integer greater than or equal to 20% of the total number of sentences in the dialogue.

2. For each of the l selected sentences, select the k_j most informative words for sentence j and assemble them as the word-level summary. k_j is calculated as the ceiling of $20\% \times T_j$ for the selected sentence j, where T_j is the total number of words in sentence j.

These sentence-level summaries are the reference summaries for our sentence extraction methods and the word-level summaries are the reference summaries for our word extraction methods. We use the popular automatic summarization metric ROUGE (Lin and Hovy, 2003) to evaluate the quality of the summarization. We report unigram overlap (ROUGE-1) and bigram overlap (ROUGE-2) as the metrics for informativeness and the longest common sub-sequence overlap (ROUGE-L) as the metric for fluency. To our knowledge, this is the first large scale dataset of customer service dialogues that are manually labeled specifically for quantitative evaluation of the attention scores' plausibility for extractive summarization.

4.3 Implementation Details

We use Bidirectional GRUs (Yang et al., 2016) for both the word encoder and sentence encoder in our hierarchical attention network reader. For the seq2seq and the seq2seq with dynamic attention

downstream supervised learners, we use GRUs as the decoders. In our experiments, we use the tokenization script from Stanford's CoreNLP toolkit (Manning et al., 2014). The 100,000 most frequent words (87.5% of total vocabulary) are used to train our models. Any word not included in the shortlist is mapped to a special token ([UNK]). We do not apply any other special preprocessing, such as stop words deleting or stemming, to the data. We use 200 for word embedding dimension and 50 hidden units for each GRU. Each of the context vectors u_w and u_s has a dimension of 100, and is initialized at random. We train our models with Adam (Kingma and Ba, 2014) with an initial learning rate of 0.001. The two momentum parameters are set to 0.9 and 0.999 respectively. We use a mini-batch size of 64.

4.4 Comparison Methods

We implement and compare several summarization models.

- Classification (base): A typical customer service dialogue starts with a customer describing an issue or asking a question, followed by several conversation rounds for more context, and ends with the agent taking actions to resolve the issue or escalating to another channel. Therefore most of the information to predict the customer issue lies in the customer's utterance while most of the information for agent action lies in the agent's utterance. For these reasons, we build two separate classification models (as defined in Section 3.2.1) for customer issue and agent action, respectively. The first model takes the concatenated customer messages as the input and predicts the category for the customer issue. In our dataset, we group customer issues into 19 categories. Similarly, the second model takes the concatenated agent messages to predict the category for the agent action. We group agent actions into 16 categories in our dataset.

- Classification (ensemble): We also built an ensembled classification model where we concatenate all utterances in the dialogue using their original order to predict a combined category. Since there are 19 classes in customer issue and 16 classes in agent action, there are 304 classes in total for the ensembled label. As pointed out in Section 3.2.1, as customer service contacts get longer and more complex, the number of classes for this approach could grow drastically and a classifier model will no longer suffice. Another pain point for the classification models is that we need to come up with a manual mapping to group all customer issues and agent actions into a fixed number of categories for each foreign language we expand to. The seq2seq models will address these issues.

- Seq2seq (base) is the set of two seq2seq models as defined in Section 3.2.2 to predict customer issue as a sequential output, such as "cancel order", from customer utterance and to predict agent action as a sequential output, such as "full refund", from agent utterance, respectively.

- Seq2seq (ensemble) is the ensembled seq2seq model where the input is the concatenated utterances from both customer and agent and the output is customer issue concatenated with agent action. For example, for the dialogue in Table 1, the sequential output is "cancel order full refund".

- Seq2seq + Att (base) is the set of two seq2seq models with an attention mechanism as defined in Section 3.2.3 to predict customer issue from customer utterance and agent action from agent utterance.

- Seq2seq + Att (ensemble) is the ensembled seq2seq model with an attention mechanism using concatenate utterances from both customer and agent to predict concatenated customer issue and agent action as a sequential output.

5 Results

After each of the two annotators finish annotating 500 contacts, the gold annotator has validated their results and verified that the standards and qualities are consistent across all 1,000 contacts. Table 2 summarizes our evaluation results using ROUGE-1,ROUGE-2, and ROUGE-L based on these 1,000 contacts. It is clear from the table that among all models, the Seq2seq + Att models outperform the rest with a significant margin with one exception of ROUGE-2 in sentence extraction. It is interesting to note that scores for the base models are generally higher compared to the ensembled models for Classification and Seq2seq. This is due to

the fact that transcripts (from either customer or agent) in the base models are significantly shorter than transcripts (from both customer and agent) in the ensembled models, therefore it is easier for the base models to compress all information into a fixed-length vector. On the other hand, the ensembled model outperforms the base model for Seq2seq + Att. This is due to two factors: 1) there is still valuable information in agent's utterance to infer customer's issue and valuable information in the customer's utterance to infer agent's action; 2) Seq2seq + Att models have great advantage in generating summaries with complicated and long dialogues.

The word extraction models are less promising. This is somewhat expected given that our models select a pre-determined number (proportional to sentence length) of words for each sentence while the true number of key words in a sentence could vary largely for sentences with the same length but in different contexts. This suggests that an alternative to our network would be to employ a word extractor that can learn the optimal number of words to extract given the context in the sentence and in the entire dialogue. We leave this to future work.

One of the motivations to use an attention mechanism in the Seq2seq + Att models was to overcome the bottleneck of a fixed-length context vector in the basic encoder–decoder Seq2seq approach. In Figure 3, we compare model performance for varying length of dialogues. We observe that the performance of all models except for Seq2seq + Att (base) and Seq2seq + Att (ensemble) dramatically decreases as the length of the dialogue increases. For shorter dialogues, Seq2seq + Att (base) and Seq2seq + Att (ensemble) are slightly better than the other models while for longer dialogues they significantly outperform the others. They show no significant performance deterioration even with dialogues of 50 or more sentences, which is critical for customer service as the need for a good summary increases as the length of a conversation grows.

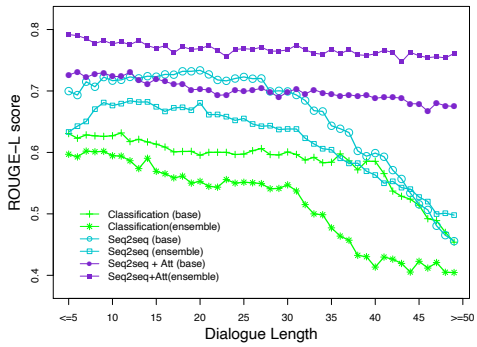

Figure 3: Model Performances with Respect to Dialogue Length (Total Number of Sentences in a Dialogue)

6 Related Work

Much effort has been devoted to automatic summarization in recent years due to an increasing need to access and digest large amounts of textual data. An ideal summarization system would understand each document and generate an appropriate summary directly from the results of that understanding, which is the abstractive summarization approach. However, a more practical approach to this problem results in the use of an approximation where a summary is created by identifying and subsequently concatenating the most salient text units in a document, namely the extractive summarization approach. The idea of creating a summary by extracting text units directly from the source document was introduced by Banko et al. (2000) who viewed summarization as a problem analogous to statistical machine translation where the task is to generate summaries in a more concise language from a source document in a more verbose language. Our approach for the sequential output to predict target words of customer issues and agent actions is similar in spirit, however, our work focuses on locating important sentences and words in the original document using an attention mechanism.

Other sentence extraction methods heavily relied on human-engineered features such as sentence position and length (Radev et al., 2004), the words in the title, the presence of proper nouns, word frequency (Nenkova et al., 2006), and event features such as action nouns (Filatova and Hatzivassiloglou, 2004). Kobayashi et al. (2015) and Yogatama et al. (2015) developed a sentence extraction approach based on pretrained sentence embeddings. Rush et al. (2015) proposed a neural attention model for abstractive summarization for individual sentences which was trained on a corpus of pairs of headlines and first sentences in news articles. Cheng and Lapata (2016) extended this approach and developed a general framework for document summarization. To address the lack of

Model	Sentence Extraction			Word Extraction		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Classification (base)	62.7	53.1	59.9	33.9	27.8	30.9
Classification(ensemble)	50.7	24.5	41.8	24.4	13.5	29.2
Seq2seq (base)	69.1	**57.6**	70.2	36.2	29.6	33.8
Seq2seq (ensemble)	64.4	48.4	64.5	30.3	22.9	35.9
Seq2seq + Att (base)	74.5	51.2	69.6	43.7	26.6	36.4
Seq2seq + Att(ensemble)	**88.2**	52.3	**76.7**	**54.6**	**32.0**	**39.3**

Table 2: ROUGE Evaluation

training data issue, they retrieved hundreds of thousands of news articles and used the corresponding highlights from the DailyMail website as the labels.

Liu et al. (2019) introduced auxiliary key point sequences to automatically generate dialogue summaries for customer service contacts at Didi, a leading mobile transportation company in China. A key point sequence acts as an auxiliary label to help the model learn the logic of the summary. The model predicts the key point sequence first and then uses it to guide the prediction of the summary. Didi requires its customer service agents to write summaries about dialogues with users, therefore, lack of labeled data is not an issue in their use case.

Our work can be considered as a continuous form of the hierarchical attention network implemented in Yang et al. (2016). Unlike Yang et al. (2016) which was developed for document classification and the prediction had to be a categorical variable, we presented a few different types of decoders that can make predictions on either categorical outcomes (such as customer sentiment) or sequential outcomes (such as customer issues and agent actions). In this paper we explore the application of hierarchical attention mechanism in dialogue summarization in the absence of labeled data. To the best of our knowledge this is the first such instance.

7 Conclusion and Future Work

The conventional approach to summarize documents/texts does not apply to cases with lack of existing summaries to supervise a training process. In this paper, we propose a novel approach based on ancillary labels and attention mechanism to address this issue. We show that this approach generates intuitive summaries and the good performance does not deteriorate as the length of dialogue increases. We test the proposed models on Amazon customer service contacts and reveal that the attention mechanism can correctly locate and retrieve relevant sentences and words which are then used to form the summaries.

We leave several summarization challenges as open questions. For example, in our approach, we set the summary length threshold of selected sentences and words to 20% . Further evaluation can be performed to observe the summarization performance with respect to different summary lengths. Furthermore, an alternative model that can jointly learn the optimal number of sentences and words to extract during training would be worthy of interest. In our work, we rank the sentences/words with their attention scores and use the sentences/words with the highest scores as the summary. In other words, we are more interested in the relative ranking of each sentence/word rather than its exact scores. Therefore, another future work direction is to incorporate a ranking algorithm in attention retrieval. Lastly, machine-generated extractive summaries may contain multiple sentences which are similar in meaning, hence not a desirable factor. It is also worthwhile to explore a redundancy elimination approach that takes a machine generated summary as a rough summary, identifies the semantic similarity between sentences in the summary, and further refines the summary by removing redundant segments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 318–325, Hong Kong. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *ACL Workshop on Summarization*.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization based on embedding distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal. Association for Computational Linguistics.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387, New York, New York, USA. PMLR.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1957–1965, New York, NY, USA. Association for Computing Machinery.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580.

Haojie Pan, Junpei Zhou, Zhou Zhao, Yan Liu, Deng Cai, and Min Yang. 2018. Dial2desc: end-to-end dialogue description generation. *arXiv preprint arXiv:1811.00185*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu

Liu, et al. 2004. MEAD - a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive summarization by maximizing semantic volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966, Lisbon, Portugal. Association for Computational Linguistics.