Customer Support on Twitter

Weitao Lin

January 16, 2018

1 Dataset

The Customer Support Dataset is a large, modern corpus for study of modern customer support practices and impact. It is publicly accessible on Kaggle.

The dataset is a csv file with 2811774 rows and 7 columns (tweet_id, author_id, inbound, created_at, text, response_tweet_id, in_response_to_tweet_id). Each row is a tweet with an unique tweet_id and we can recover the conversation between customer and agent by referencing response_tweet_id and in_response_to_tweet_id. For example, we can see below how a customer is complaining about sprintcare's customer service and how tweet_id gets referenced.

tweet_id	author_id	inbound	created_at	text	response_tweet_id	in_response_to_tweet_id
8	115712	True	Tue Oct 31 21:45:10 +0000 2017	@sprintcare is the worst customer service	9,6,10	NaN
10	sprintcare	False	Tue Oct 31 21:45:59 +0000 2017	@115712 Hello! We never like our customers to ...	NaN	8.0
9	sprintcare	False	Tue Oct 31 21:46:14 +0000 2017	@115712 I would love the chance to review the ...	NaN	8.0
6	sprintcare	False	Tue Oct 31 21:46:24 +0000 2017	@115712 Can you please send us a private messa...	5,7	8.0
7	115712	True	Tue Oct 31 21:47:48 +0000 2017	@sprintcare the only way I can get a response ...	NaN	6.0
5	115712	True	Tue Oct 31 21:49:35 +0000 2017	@sprintcare I did.	4	6.0
4	sprintcare	False	Tue Oct 31 21:54:49 +0000 2017	@115712 Please send us a Private Message so th...	3	5.0
3	115712	True	Tue Oct 31 22:08:27 +0000 2017	@sprintcare I have sent several private messag...	1	4.0
1	sprintcare	False	Tue Oct 31 22:10:47 +0000 2017	@115712 I understand. I would like to assist y...	2	3.0
2	115712	True	Tue Oct 31 22:11:45 +0000 2017	@sprintcare and how do you propose we do that	NaN	1.0

Figure 1: Example conversation

More details to know about this dataset:

1. The inbound column basically indicates whether this tweet is posted by a customer (True) or agent (False).
2. There are 108 different companies listed in this dataset. The 10 most active ones and least active ones are:

	author_id	count
8	AmazonHelp	169840
10	AppleSupport	106860
85	Uber_Support	56270
77	SpotifyCares	43265
40	Delta	42253
80	Tesco	38573
9	AmericanAir	36764
78	TMobileHelp	34317
99	comcastcares	33031
33	British_Airways	29361

	author_id	count
59	MOO	630
95	ask_progressive	612
56	KeyBank_Help	555
21	AskRobinhood	432
103	mediatemplehelp	305
53	JackBox	266
69	OfficeSupport	218
15	AskDSC	210
34	CarlsJr	196
51	HotelTonightCX	152

Figure 2: 10 most active (left) and least active (right) companies

3. Among 2811774 tweets, 1273931 (around 45%) of them are posted by customers and the rest are posted by company agents.
4. Column in_response_to_tweet_id only contains one float value or NaN, while Column response_tweet_id contains one or more float value or NaN.
5. As column in_response_to_tweet_id being NaN indicates the beginning of a dialogue, we can calculate that there are 794335 dialogues.
6. We also find that in some dialogues (around 0.3%) agents from more than one companies may show up.

tweet_id	author_id	inbound	created_at	text	response_tweet_id	in_response_to_tweet_id
1299	115951	True	2017-10-31 22:22:31	@115913 I'm a #TMobile customer getting the ru...	1298,1300,1301	NaN
1301	TMobileHelp	False	2017-10-31 22:23:27	@115951 @115913 We always want to keep you con...	NaN	1299.0
1300	sprintcare	False	2017-10-31 22:24:30	@115951 Hello please go to the link I will nee...	NaN	1299.0
1298	sprintcare	False	2017-10-31 22:25:24	@115951 Get started here: https://t.co/dYSbRQ3...	NaN	1299.0

Figure 3: Example conversation showing more than one company may appear in a conversation

7. On average, the length (number of tweets) in a conversation is around 3.5. The smallest number of tweets is 2, which means at least we have a question answer pair. Actually, around 55% percent of dialogues only contain 2 tweets (around 0.1% of these dialogues may suffer from loss of data).

2 Future Work

2.1 Classification Task

Build an AI that given some question asked by a customer, it can either choose to select an existing answer or, when it decides there isn't a good match, release a signal so that a human agent can come to take care of this question. Basically this is a classification problem where the machine try to classify the given question with an additional class "Don't know".

One of the questions to think about here is how we organize the labels. Apparently there are many similar answers such as "Please reboot your device" and "Could you please try restarting the device". We need some preprocessing to map these two answers to one target. (Probably with some keyword synonym or pre-trained word embedding?)

2.2 Generation Task

Build a generative chatbot that can intelligently generate a answer based on the knowledge it learns from historical data, rather than pick an answer. This is more challenging but also more ideal because as new products and services come out, many questions just cannot be answered perfectly with historical answers.

The most popular approach to generative nlp problem these days is to construct seq2seq RNN model. This has been widely used in machine translation, question answering, automatic summarization, etc.

2.3 Sequential Dialogue

Currently the way most chatbot models get trained is learning from question answering pairs. However, in this dataset, we have dialogues with long sequence. Some dialogues contain more than 10 tweets. How can we train a chatbot that can learn from this kind of sequential dialogues is another question to think about. (For example some attention mechanism that allows the model makes decision based on what it hears and says. A complicated model such as hierarchical RNN may work, but that might take long time to build and train.)

3 Literature Review

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

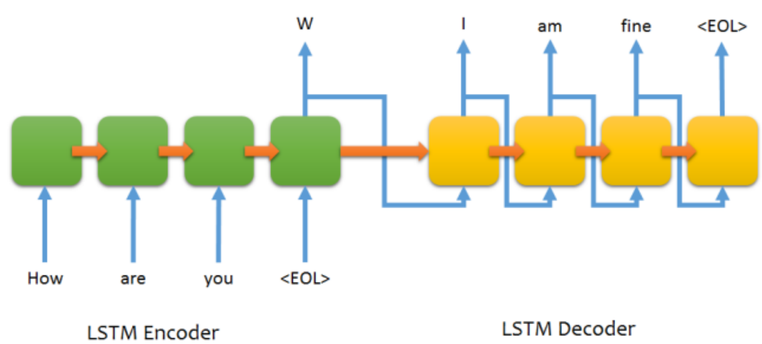

Figure 4: layout of a simple seq2seq model

Recently some effort has also been put into merging IR and generation model. For example AliMe[Qiu et al., 2017] first use an IR model to retrieve a set of QA pairs and use them as candidate answers, and then rerank the candi- date answers using an attentive Seq2Seq model: if the top candidate has a score higher than a certain threshold, it will be taken as the answer; otherwise the answer will be offered by a generation based model.

References

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

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

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 498–503, 2017.

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

Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Zhao Yan, Nan Duan, Jun-Wei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *ACL (1)*, 2016.