A Two-stage User Intent Detection Model on Complicated Utterances with Multi-task Learning

Shuangyong Song, Xiangyan Chen, Chao Wang, Xiaoguang Yu, Jia Wang, Xiaodong He
{songshuangyong,chenxiangyan5,wangchao208,cdyuxiaoguang,cdwangjia5,hexiaodong}@jd.com
JD AI Research
Beijing 100176, China

ABSTRACT

As one of the most natural manner of human−machine interaction, the dialogue systems have attracted much attention in recent years, such as chatbots and intelligent customer service bots, etc. Intents of concise user utterances can be easily detected with classic text classification models or text matching models, while complicated utterances are harder to understand directly. In this paper, for improving the user intent detection from complicated utterances in an intelligent customer service bot JIMI (JD Instant Messaging intelligence), which is designed for creating an innovative online shopping experience in E-commerce, we propose a two-stage model which combines sentence *C*ompression and intent *C*lassification together with *M*ulti-task learning, called *MCC*. Besides, a dialogue-oriented language model is trained to further improve the performance of *MCC*. Experimental results show that our model can achieve good performance on both a public dataset and the JIMI dataset.

CCS CONCEPTS

• **Information systems** → **Query intent**.

KEYWORDS

dialogue system, user intent detection, multi-task learning

ACM Reference Format:
Shuangyong Song, Xiangyan Chen, Chao Wang, Xiaoguang Yu, Jia Wang, Xiaodong He. 2022. A Two-stage User Intent Detection Model on Complicated Utterances with Multi-task Learning. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524232

1 INTRODUCTION

During the last few years, question answering (QA) based intelligent dialogue systems have been very popular, such as chatbots and intelligent customer service bots. That is partly due to the progresses achieved in deep learning and big data techniques, and partly due to the growing requirements in the real-world.

Customer service is one of the promising fields that intelligent assistants can play a key role in, especially the E-commerce customer service. On one hand, there is a strong demand for customer service staff in the E-commerce field along with the fast-growing market. On the other hand, with more and more people paying attention to shopping experience and service quality nowadays, the inefficiency and long turn-on time of traditional customer service are becoming obvious, especially during some promotion seasons.

Due to the above situation, there is a strong demand for E-commerce websites, such as Amazon.com [1], Taobao.com [4], and JD.com [6], to build their own intelligent customer service bots, which can help relieve customer service staff from answering simple, common and repetitive questions, and let them focus on the cases that really need human participation.

User intent detection, which is crucial for providing appropriate responses, is the most essential module in intelligent customer service bots, and most user intent detection tasks are based on text classification models or text matching models [8, 11]. However, intents of concise single-turn user utterances can be easily detected with some classic models, while multi-turn utterances and complicated utterances are harder to understand directly. Some work has been done on the user intent detection from multi-turn utterances [5, 9], and in this paper we focus on the user intent detection from complicated utterances.

User utterances without detected specific intents are replied with general answers or transferred to manual customer service, which either hurts the user experience or increases manual workloads. Complicated utterances are precisely this kind of utterances, which are hard to be detected with specific intents due to the noisy words in utterance text. For weakening the effect of those noisy words and strengthening the effect of real intent-related words, we adopt the sentence compression (which is also called as 'in-sentence summarization') [7], and propose a two-stage user intent detection model with *M*ulti-task learning on sentence *C*ompression task and intent *C*lassification task, which is called *MCC*. Besides, a dialogue-oriented language model is trained to further improve the performance of *MCC*.

We evaluate the *MCC* on two datasets, a public CCL dataset and an in-house JIMI dataset. The CCL dataset is a public Chinese query classification dataset in the domain of telecommunication released for the China National Conference on Computational Linguistics 2018 shared task 1 [10], and the JIMI dataset is collected from the JD[1] Instant Messaging Intelligence, an intelligent service robot designed for E-commerce shopping. The experiment results show that our model can achieve good performance on user intent detection from complicated utterances.

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524232

[1] https://www.jd.com/

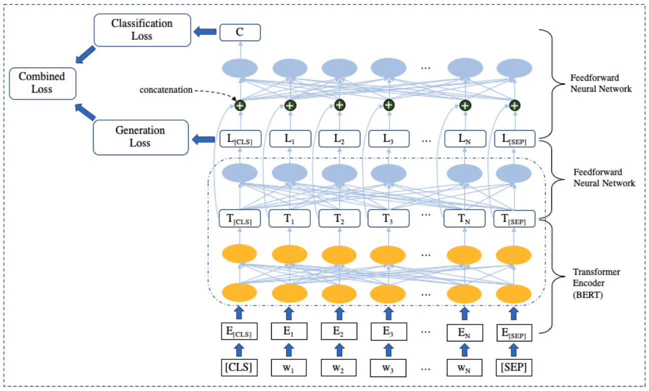

Figure 1: Model architecture.

2 OUR MODEL - *MCC*

2.1 Model Overview

Figure 1 gives the model architecture of the *MCC* with a dialogue oriented fine-tuned BERT [2] language model. We first use large amounts of data to fine-tune the original BERT model as the Transformer Encoder in our model. Then a sentence generation (compression) model is used to compress the complicated utterances into concise sentences. After that, a simple multi-layer perceptron based text classification is employed to complete the user intent detection task. Furthermore, the sentence generation loss and the text classification loss are optimized together as multi-task learning.

2.2 BERT fine-tuning

In this experiment, we used Google’s pre-training Chinese_L-12_H-768_A-12 model [2] to perform BERT encoding on each barrage, so that the word vector of each barrage text became 768 dimensions. We fine-tune it to a dialogue-oriented BERT (dBERT) with multi-tasks on large scale dialogue data. The first task is a masked language model (MLM), which use user utterances. The second task is a next sentence prediction (NSP), which use both user utterances and bot answers with a StructBERT [12] strategy, which incorporates language structures into language model pre-training, getting better generalizability and adaptability. Dialogue data from chatbots is a specific user generated content (UGC), riddled with typos and solecisms. StructBERT can well fit UGC by leveraging the structural information. The third task is a text matching task, which considering the semantic relations between user utterances. 170 million pieces of data are built for this dBERT training.

2.3 Complicated Utterance Compression

A sentence compression model is implemented on complicated utterances for generating concise sentences. In our work, we adopt the LaserTagger [7] model, which is a sequence tagging approach that casts text generation as a text editing task. LaserTagger is usually used on sentence compression (which is also called as ‘in-sentence summarization’), in which the target texts are reconstructed from the inputs using three main edit operations: *keeping* a token, *deleting* it, and *adding* a phrase before the token. Due to the strict limitation of system running speed, the LaserTagger used in this paper is the LaserTagger$_{FF}$, which is obviously faster than another version of LaserTagger, LaserTagger$_{AR}$, since the LaserTagger$_{FF}$ use the feedforward decoder instead of the autoregressive decoder in LaserTagger$_{AR}$, which is shown in Figure 1.

Table 1: Data length description

dataset	JIMI	CCL
1~5	16.16%	27.04%
6~10	26.37%	43.74%
11~15	20.17%	17.79%
16~20	10.23%	11.17%
> 20	27.07%	0.26%

For better adapting the LaserTagger to our utterance compression task, we design a modified LaserTagger (called as ‘mLaserTagger’), which just use the keeping and deleting tags, and disuse the adding tag. This make the generated sentence is a verbal subset of the original user utterance. The generation loss function of the mLaserTagger is denoted as $\mathcal{L}_g$.

2.4 Text Classification

Multi-layer perceptron is used to complete the last classification step. Cross-entropy (*CE*) is a standard classification loss function that is widely used in multi-class classification tasks, defined as:

$$\mathcal{L}_c = CE(y,p) = -\sum_{i=1}^{N} y_i log(p_i) = -log(p_k) \qquad (1)$$

where p is the estimated probability distribution, and y is the true probability distribution. For one-hot encoding, only the probability corresponding to the ground truth is 1.

2.5 Joint Training

We jointly optimize the sentence generation (compression) loss $\mathcal{L}_g$ and intent classification loss $\mathcal{L}_c$ to train our model with a loss $\mathcal{L} = \mathcal{L}_g + \mathcal{L}_c$, where the two tasks are also with a serial relation.

3 EXPERIMENTS

3.1 Collection of Datasets

3.1.1 *User Intent Classification Datasets.*

We evaluate the user intent classification task of *MCC* on a benchmark CCL dataset and an in-house JIMI dataset. The CCL dataset is a public Chinese query classification dataset, excluding samples with labels such as GREETINGS and COMPLAINTS, which contains about 20 thousand instances. The JIMI dataset is collected from real user utterances, and after several years of accumulation, about 596 thousand manually labelled user intent classification instances have been collected, with more than 2,000 fine-grained three-level user intents (classes), no matter if short or long utterances.

However, for long (complicated) utterances, the class distribution has a long-tail. The detail statistics of instance length are presented in Table 1. We can see that the long instances (length > 15) account for 37.30% and 11.43% in those two datasets respectively.

3.1.2 *Sentence Compression Datasets.*

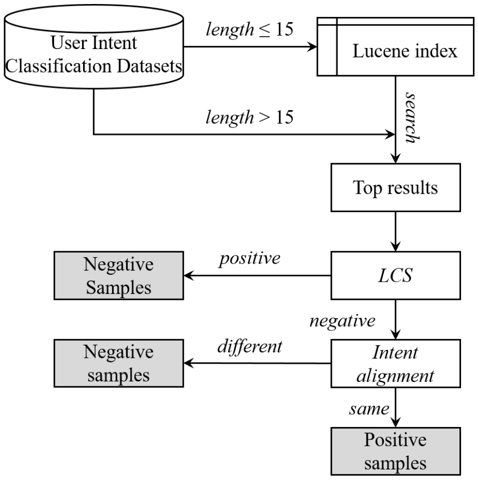

Figure 2: Building of sentence compression datasets.

In this paper, we roughly take the length of 15 as the boundary to distinguish whether an instance is complicated or concise. Figure 2 depicts the automatical building of the sentence compression datasets. We first create an indexing for all the short instances in the manually labelled user intent classification dataset using Apache Lucene[2]. Next, given a long instance I_l, we employ TF-IDF ranking algorithm [13] in Lucene to compute its similarities to all the short ones, and call back the top-K candidates. Then a variant of longest common substring (*LCS*) is used to further verify relation between I_l and the candidates: if the word set of a candidate is a subset of the word set of I_l, and meanwhile the word order has not been changed in I_l, it is labelled as 'positive', otherwise it is labelled as 'negative'. Finally, a candidate with 'positive' label and with the same intent label with I_l will be saved as a piece of sentence compression pair <*complicated instance*, *concise instance*> data.

In particular, we manually check 5,000 sentence compression pairs of JIMI to build a standard test set.

3.1.3 *BERT Fine-tuning Datasets.*

According to the BERT fine-tuning strategies mentioned in subsection 2.2, we build several datasets to train the dialogue-oriented dBERT: 1) Document-utterance dataset: similar with the BERT training data construction, a <utterance, utterance> pair starts with a [CLS] token, and for the NSP task, the [CLS] is labelled as 'positive' when a pair of utterances are from the same document and the [CLS] is labelled as 'negative' when they are from different documents. In particular, we denote a dialogue session as a document. 2) Utterance-Answer dataset: <utterance, answer> pairs are constructed in this dataset, and another annotation criterion of [CLS] is used as: if the utterance and the answer are from a same document and the answer is prior to the utterance, the [CLS] is labelled as 'prior'; If they are from a same document and the answer is behind the utterance, it is labelled as 'behind'; If they are from different documents, it is labelled as 'allogenetic'. 3) Utterance matching dataset: all the utterances are indexed using Apache Lucene, and with a random utterance we search top 200 similar candidate utterances with it. For this utterance and a candidate, we compare the second level intents and the third level intents of them, and the [CLS] label of this <utterance, utterance> pair matching is used as: if those two utterance are with a same third level intent, the [CLS] is labelled as 'same'; If they are with different third level intents but with a same second level intent, it is labelled as 'similar'; If they are with different second level intents, it is labelled as 'different'.

Table 2: Intent classification performance with BERT on user utterances of different lengths

utterance length	< 15	≥ 15
ACC	93.25	82.15
PSI	76.70	36.20

Table 3: The sentence compression performance

models	Exact	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	SARI	Keep	Deletion
MT{BERT+ mLaserTagger}	15.633	75.2	70.4	65.6	60.1	81.7	58.513	67.834	82.237
MT{dBERT+ mLaserTagger}	18.361	75.8	71.2	66.6	62.1	83.0	60.000	69.503	82.792

Finally, a dynamic masking technique [15] is used on those datasets and totally about 170 million instances are constructed.

3.2 Implementation

Our model is implemented in Python and TensorFlow. In experiments, all the BERT related training use the same parameters as the original version: hidden size: 768, max position embeddings: 512, num attention heads: 12, num hidden layers: 12, vocab size: 21128, and all the sentence compression related training use the same parameters as LaserTagger model, expect the nonuse of the 'adding' label in mLaserTagger model. Besides, the models are all trained on NVIDIA Tesla V100 GPUs.

3.3 Experimental Results

In Table 2, we show the intent classification performance (ACC and PSI, which means the proportion of specific intents) with original BERT on user utterances of different lengths. Since most of short utterances can express more concise intents than long utterances.

The main purpose of this paper is to verify the performance of the two-stage multi-task learning architecture, so we don't select multiple classic models as baselines, such as classification models HAN [14], textCNN [3], etc., since those general text classification models have poor performance on the complicated utterance intent classification task. In Table 3, we just show the compression results of JIMI, since we can't manually label the sentence compression results of CCL dataset without background knowledge. BERT and dBERT are both fine-tuned with the user intent classification data, and the MT{*} models means multi-tasks fine-tuned with both user intent classification data and sentence compression data. To allow for a good comparison, we denote the *MCC* as

[2] https://lucene.apache.org/core

Table 4: Examples of JIMI (translated from Chinese)

user utterance	intent	compression result	intent
I bought a box of batteries. I wanted to buy No. 5, but the order was No. 7. Now the goods have been received. Can I apply for replacement.	about the order (general answer)	Can I apply for replacement of batteries.	replacement (specific answer)
Hello, I bought two bags of diapers yesterday, because there was no discount when I bought them together, so I divided them into two orders. Today, I saw that there was a discount when I bought two bags at one time. I failed to apply for insurance.	about the price (general answer)	Order application price protection failed.	price protection (specific answer)
The card head of the Xiaomi hair dryer I bought on jd.com was burnt. I replaced it with a new one. After using it for more than 10 days, the card head was burnt again. The customer service contacted me to return it, but I didn't receive a refund after returning it.	about the refund (general answer)	The hair dryer didn't receive a refund.	refund not received (specific answer)

Table 5: The classification performance of user intents from complicated utterances

models	JIMI		CCL
	ACC	PSI	ACC
BERT	82.15	36.20	84.78
dBERT	82.96	55.34	86.96
MT{BERT+mLaserTagger}	82.44	42.22	89.13
MT{dBERT+mLaserTagger}	83.11	64.57	91.30

MT{dBERT+mLaserTagger}. Exact (exact match), BLEU and ROUGE-L are general evaluation metrics, which we won't explain the definition. Besides, SARI, KEEP (keep score), DELETION (deletion score) and ADDITION (addition score) are evaluation metrics proposed along with the LaserTagger model [7], and due to the nonuse of addition label in *MCC*, the ADDITION is also not evaluated.

Table 5 shows the final user intent classification results. The results show that dBERT performs better than original BERT, and with a multi-task mechanism, an associated sentence compression task with mLaserTagger can help to get even better user intent classification performance. This conclusion can be even better reflected by the CCL results than the JIMI results. Especially on the PSI metric, dBERT and multi-task mechanism can help improve the PSI from 36.20% to 64.57%, achieving an incredible increase. Since there is no distinction of 'specific' and 'general' intents in CCL dataset, we just evaluate the PSI metric on JIMI dataset.

In Table 4, we show some examples of comparison on user intent classification with or without the multi-task mechanism, and also the examples of sentence compression results. With the *MCC*, keywords can be better highlighted which makes the intent classification results to be spesific ones instead of general results.

With the proposed model, additional 28.37% general answers to long user utterances have been replaced with specific business answers, which brought the improvement of customer satisfaction degree with 1.32%, and the improvement of problem resolution rate with 1.37%.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a user intent detection model *MCC*, based on a multi-task learning mechanism and a fine-tuned language model. Besides, we proposed a automatic data construction method of sentence compression. In the future, we will try to realize a 'reduced model' to better meet the queries-per-second (QPS) needs of real online applications. How to accurately detect complicated utterances instead of roughly select them with text length will be an important improvement of the current work. Besides, since the architecture of user intent in most dialogue systems is hierarchical, future work contains combining the sentence compression and a hierarchical intent classification. Furthermore, considering both the origin utterances and compressed results on final intent detection is a possible optimization approach.

REFERENCES

[1] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *ACL 2017*. 97–102.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL 2019*. 4171–4186.

[3] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP 2014*. 1746–1751.

[4] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, and Wei Chu. 2017. *AliMe Assist* : An Intelligent Assistant for Creating an Innovative E-commerce Experience. In *CIKM 2017*. 2495–2498.

[5] Hang Liu, Meng Chen, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Conversational Query Rewriting with Self-Supervised Learning. In *ICASSP 2021*. 7628–7632.

[6] Ruixue Liu, Meng Chen, Hang Liu, Lei Shen, Yang Song, and Xiaodong He. 2020. Enhancing Multi-turn Dialogue Modeling with Intent Information for E-Commerce Customer Service. In *CCL 2020*. 65–77.

[7] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, Tag, Realize: High-Precision Text Editing. In *EMNLP 2019*. 5053–5064.

[8] Shuangyong Song, Haiqing Chen, and Zhiwei Shi. 2017. Intention classification of user queries in intelligent customer service system. In *IALP 2017*. 83–86.

[9] Shuangyong Song, Chao Wang, Qianqian Xie, Xinxing Zu, Huan Chen, and Haiqing Chen. 2020. A Two-stage Conversational Query Rewriting Model with Multi-task Learning. In *The Web Conference 2020*. 6–7.

[10] Maosong Sun, Ting Liu, Xiaojie Wang, Zhiyuan Liu, and Yang Liu (Eds.). 2018. *CCL 2018*.

[11] Chengyu Wang, Haojie Pan, Yuan Liu, Kehan Chen, Minghui Qiu, Wei Zhou, Jun Huang, Haiqing Chen, Wei Lin, and Deng Cai. 2021. MeLL: Large-scale Extensible User Intent Classification for Dialogue Systems with Meta Lifelong Learning. In *KDD '21*. 3649–3659.

[12] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. In *ICLR 2020*.

[13] Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, and Kui-Lam Kwok. 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.* 26, 3 (2008), 13:1–13:37.

[14] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL HLT 2016*. 1480–1489.

[15] Zewei Zhao and Houfeng Wang. 2020. MaskGEC: Improving Neural Grammatical Error Correction via Dynamic Masking. In *AAAI 2020*. 1226–1233.