Utterance Alignment in Custom Service by Integer Programming

Guirong Bai[1,2], Shizhu He[1], Kang Liu[1,2], and Jun Zhao[1,2]

[1] National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{guirong.bai,shizhu.he,kliu,jzhao}@nlpr.ia.ac.cn

Abstract. In customer service (CS), customers pose questions that will be answered by customer service staff, and the communication in CS is a typical multi-round conversation. However, there are no explicit correspondences among conversational utterances, and obtaining the explicit alignments of those utterances not only contributes to dialogue analysis but also provides valuable data for learning intelligent dialogue systems. In this paper, we first present a study on utterance alignment (UA) in CS. We divide the alignment of utterances into four types: *None*, *One-to-One*, *One-to-Many* and *Jump*. The direct design models such as rule-based and matching-based methods are often only good at solving part of types, and the major reason is that they ignore the interactions of different utterances. Therefore, to model the mutual influence of different utterances as well as their alignments, we propose a joint model which models the UA as a task of joint disambiguation and resolved by integer programming. We conduct experiments on a dataset of an in-house online CS. And the results indicate that it performs better than baseline models, especially for *One-to-Many* and *Jump* alignments.

Keywords: Utterance alignment · Integer programming · Customer service.

1 Introduction

Customer service (**CS**), which provides an irreplaceable platform for sellers to answer the questions and solve the problems of customers in the shopping, plays important roles in e-commerce websites. Recently, with the development of the artificial intelligence techniques, automatic CS conversation analysis of CS has attracted more and more attention, including intention analysis [3], emotion identification [6], suggestion mining [12], etc.

In general, in CS, **customer**s pose questions that will be answered by customer service staff (**server**s), and the communication between a customer and a server is a typical multi-round conversation. In this paper, we focus on the utterance alignment (**UA**), which is to align the utterances between different sides (customer and server) in a dialogue. This task is very important and useful

for practical online services but was seldom addressed before in our knowledge. As shown in Figure 1, based on the alignments, the questions posed by a customer are connected with the corresponding responses from the server. It could benefit the server to perform quality control and check whether the customer is satisfied with the service. Moreover, most current intelligent dialogue models such as deep learning methods [5, 9] need sufficient aligned question-answer (response) pairs to train the model. Obviously, such question-answer pairs could be automatically acquired through UA.

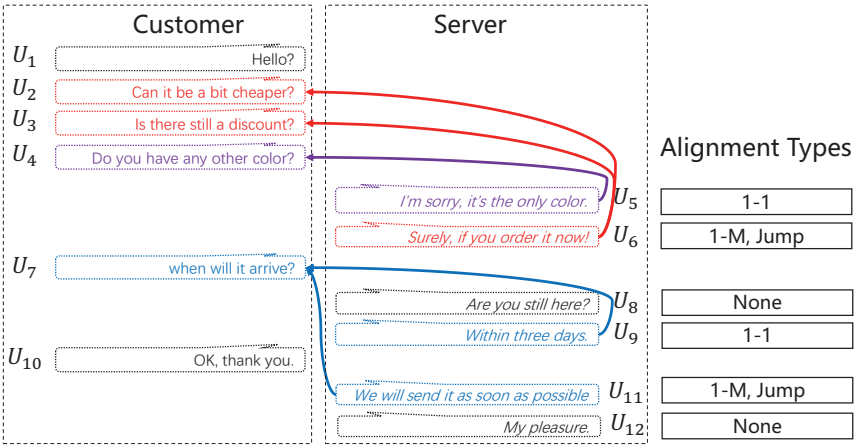

Fig. 1. Utterance alignments of a sample dialogue in CS. The left and right utterances are raised by customer and server respectively.

Although UA is a valuable task, it is full of challenges. In the conversation of CS, a response is may not adjacent the corresponding customer's question (e.g., as shown in Figure 1, U_8 does not align with U_7, and U_{11} does not align with U_{10}). And the orders of the utterances are possibly out of turns. Moreover, sometimes the customer even consults one consultation with multiple similar questions. According to our observation from the server side, there are usually four types of alignments among utterances, including ***None***, ***One-to-One*** (***1-1***), ***One-to-Many*** (***1-M***) and ***Jump***.

Intuitively, a response usually follows a corresponding question, and it is easy to think of employing the position information to align utterances as other alignment models such as IBM models [2]. However, in the types of *1-M* and *Jump*, there are freedom and disorder in a dialogue, which make the position information is too weak to align the correct utterances. It is naturally to further consider the content of utterances, where two utterances in an aligned pair must have a semantic connection or relatedness [8, 18]. Nevertheless, solely considering the semantic connection and position information of a pair may still cause a local optimization. The information of different utterances and their alignments

in a dialogue should be considered globally. In fact, the alignments of different utterances are correlated and interactional with each other. For example, if question Q_1 (e.g., U_2 in Figure 1) and Q_2 (U_3) are very *similar* and the question Q_2 (U_3) aligns with answer A_1 (U_6), the question Q_1 (U_2) should also *align* with answer A_1 (U_6). But if question Q_1 (U_3) and Q_2 (U_4) are very *dissimilar* and the question Q_2 (U_4) aligns with answer A_1 (U_5), the question Q_1 (U_3) should *align* with the answer A_1 (U_6) rather than A_1 (U_5), even though A_1 (U_5) is more closer to Q_1 (U_3) than A_1 (U_6).

To this end, we propose to find all alignments of a conversation in a joint model. We model the utterance alignment as the progress of joint disambiguation (whether U_5 aligns with U_4 or not), consider the correlatives of different utterances as well as their relationship by integer constraints (if U_2 and U_3 are similar, and U_3 aligns with U_6, U_2 should align with U_6), and resolve them by integer programming (**IP**). Moreover, two neural models are proposed to capture the semantic representation of the utterance content, which are also able to incorporate the position information. Based on the learned semantic representation, the content-based alignments are calculated. Finally, we fuse the above information of all pairs in a dialogue into an integer programming algorithm. In this way, all possible alignments could affect each other and the final results are optimized globally.

We create a dataset to verify the feasibility of the proposed model, and the experimental results demonstrate the effectiveness of the proposed model. Compared with the best matching model, the F1 is increased by 7% in totally. In special, it obtains 3.6%, 3.7% improvements on the more challenging alignments of *1-M* and *Jump*, respectively.

In brief, the main contributions are as follows:

- We propose a new task, named utterance alignment (UA), which contributes to dialogue analysis and provides valuable data for learning intelligent dialogue systems.
- We propose a joint model for UA by integer programming (IP), which considers the correlatives of different utterances as well as their relationship by integer constraints and make effects on experiments.
- We collect dialogues from a real CS and construct a dataset for UA with human-annotation.

2 Problem Definition

2.1 Utterance Alignment

In most cases, the customer poses the questions that will be answered by the server. In this work, we mainly focus on helpful and crucial question-answer pairs. Therefore, we only need to consider which customer utterances (question, Q for short) are aligned for each server utterance (answer, A for short). We formulate the task as a joint disambiguation task based on classification models: whether A_i aligns with Q_j or not.

2.2 Alignment Types

There are a number of consultations in a CS conversation as shown in Figure 1). Considering the dialogue process, we can align utterances by their posing orders (e.g., U_5 aligns with U_4, U_8 aligns with U_7). However, in the actual scenario, this simply alignment strategy is not enough. A customer may pose a number of questions at a time (e.g., U_2, U_3, U_4), and the server may answer them in different orders. In addition, each server may need to talk to multiple customers at the same time, and each customer's question may not be answered immediately, it will aggravate the above situation. On the other hand, people may express the same intention in a number of short and simple utterances in the oral communicating environment (e.g., U_2 and U_3 express the similar meaning), therefore, some answers should align with more than one questions (e.g., U_6 should align with both U_2 and U_3). Moreover, there may be some chat messages (e.g., U_8 and U_{12}) interspersed in the consulting process, and they should not be aligned with any utterance.

Therefore, considering the different alignments which may be suitable for different models, We divide the alignments into four types: *None*, *One-to-One* (1-1), *One-to-Many* (1-M) and *Jump*. The *None* type means that there is no alignment for a given utterance of the server (e.g., U_8 and U_{12} in Figure 1). The *1-1* means a response aligns with only one question (e.g., U_5, U_9 and U_{11}), which is the simplest and most intuitive alignment type. The *1-M* means a response aligns with more than one questions (e.g., U_6 should align with two utterances: U_2 and U_3). And the *Jump* means a response replies to a question which is posed serval turns ago, and their alignments cross some other questions (e.g., U_{11} should align with U_7, which crosses the closest question: U_{10}). The *1-M* and *Jump* alignments violate the regular order in a dialogue and provide main difficulties for UA, which are the main focus of this paper.

2.3 Data

We create a Chinese dataset from an online CS. We first sample 10,000 conversations from a human-to-human customer service system, which owns about 6-20 utterances for each conversational episode. we invited five annotators for the explicit alignments. For example, U_8 should be independently annotated whether or not to align with one or more of U_1, U_2, U_3, U_4 and U_7. If the server utterance is a meaningless utterance or cannot answer any customer question, it will be annotated a *None* label. For example, in Figure 1, U_{12} is a meaningless utterance as a chat message, and none of the customer utterances can semantically match U_8. So both of them are annotated *None* label. The coincidence rate of the five annotators is about 85% with another annotator reviewing.

In the end, we obtain 5,741 labeled conversations with average 6.0 turns from the server and 4.5 turns from the customer. Every turn has average 22.7 and 6.2 words on each side respectively. Moreover, for a given utterance of the server, there are average 2.8 customer utterances as alignment candidates. And the alignments of *None*, *1-1*, *1-M* and *Jump* account for 57%, 31%, 12%, and 9% respectively.

3 Utterance Alignment Models

Let $CS = [(U_1, t_1), (U_2, t_2), ..., (U_n, t_n)]$ denotes the conversation of CS, $U_i = [w_1, ..., w_{L_{U_i}}]$ indicates the word sequence of a utterance and t_i indicates the role of speaker (Customer (C) and Server (S)). For each server utterance $(U_i, t_i), t_i = S$, we should find all customer raised utterances before i ($\{j|j < i, t_j = C\}$) which could be answered by U_i.

3.1 Matching-Based Alignment

We first learn different Q-A matching models that utilize deep neural networks scoring the matching degree between a customer utterance and a server utterance. The matching score $s_{q,a}$ of a customer's question q and a server's answer a is calculated as: $s_{q,a} = \mathbf{q}^T \cdot \mathbf{M} \cdot \mathbf{a}$, where $\mathbf{q}$ and $\mathbf{a}$ are the semantic representation of them, and the matrix $\mathbf{M}$ is the parameter of the matching model. Then, we can obtain the alignments of utterance pairs when their matching scores large than a threshold. We utilize the following margin-based ranking loss to train the matching models: $L = max(0, s_{q,a'} + \gamma - s_{q,a})$ (or $L = max(0, s_{q',a} + \gamma - s_{q,a})$) , where q' and a' indicate the random selected nonaligned utterances. In specific, we obtain the representation of utterance by Convolutional Neural Networks (CNN) [1] and Long Short-Term Memory (LSTM) [7].

Position Embedding:We consider position information among utterances and design the following three features to encode the position information: 1) Index: indicates the absolute position of a server utterance in a dialogue; 2) All-Distance (*A-Dist*): records the number of utterances in the dialogue between two alignment utterances; 3) Customer-Distance (*C-Dist*): records the number of customer utterances in the dialogue between two alignment utterances. For example, as shown in Figure 1, the index, A-Dist, and C-Dist are 6, 3, 2 while judging the alignment between U_6 and U_2 (the candidate alignment utterances of U_6 are from U_1 to U_4) respectively. Each position feature is represented by a fixed-dimension vector and concatenated with sentence representation.

3.2 Utterance Alignment with Joint Disambiguation

The above-mentioned methods independently judge the alignment of each utterance pair, which cause the alignments are local optimum results. In fact, the alignments of different utterance pairs in a dialogue have coherence and interaction. As shown in Figure 2, similar questions are usually aligned with the same answer and vice versa. Therefore, we propose a joint disambiguation model with some global constraints by integer programming (IP).

We define three types of 0-1 variables ($\in \{0, 1\}$): 1) A_{ij} indicates whether the i-th customer utterance align with j-th server utterance (final results). 2) MQ_{ij} indicates whether the i-th customer utterance is semantically similar with j-th customer utterance. 3) MA_{ij} indicates whether the i-th server utterance is semantically similar with j-th server utterance.

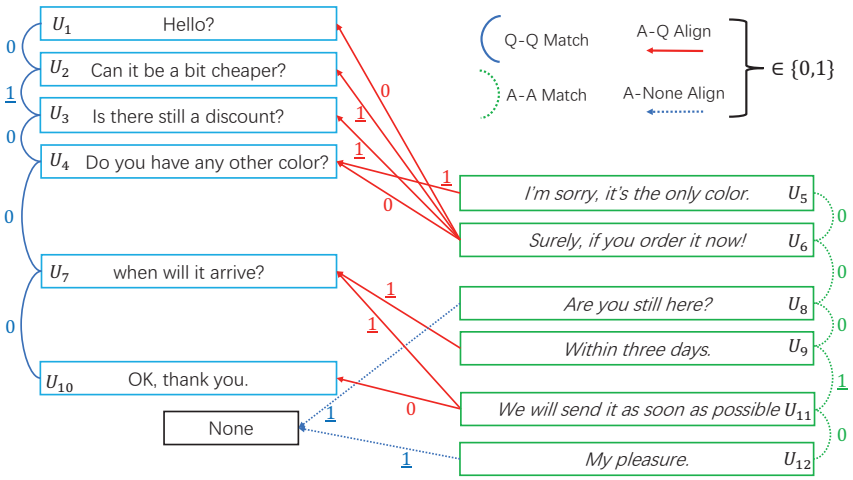

Fig. 2. A dialogue example of utterance alignments with joint disambiguation. The relations of Q-Q, A-A and Q-A/A-Q with 0/1 variables denote matching or not.

Basic models: We train three scoring models for the above three variables. The first one is adopt the matching-based methods. We define $s_{q,q}$ and $s_{a,a}$ to indicate the similarity of two questions and two answers. And they are modeled by the following function: $s_{q,q}(i,j) = \sigma(\mathbf{W}_{qq}[\mathbf{q}_i, \mathbf{q}_j])$ and $s_{a,a}(i,j) = \sigma(\mathbf{W}_{aa}[\mathbf{a}_i, \mathbf{a}_j])$, where σ indicates the *sigmoid* function, $\mathbf{W}_{qq}$ and $\mathbf{W}_{aa}$ are the model parameters trained by minimizing the cross entropy. In addition, we define $n(a) = \sigma(\mathbf{a}\mathbf{W}_n)$ to indicate the probability of the server's answer a not aligning with any utterance by minimizing the cross entropy.

Objective of the joint disambiguation model: The objective contains four parts as follows:

1) The question-answer alignment scores: the probability questions align with answers: $T_1 = \sum_{j=1}^{J}\sum_{k=1}^{K} A_{ij} \cdot (s_{q,a}(i,j) - \beta_1)$, where J and K indicate the utterance numbers of customer and server, respectively.

2) The question similarity scores: the probability questions are aligned with a same server's answer: $T_2 = \sum_{(q_j,q_{j'})\in L_q} MQ_{jj'} \cdot (s_{q,q}(j,j') - \beta_2)$, where L_q contains all candidate question pairs from the customer utterances.

3) The answer similarity scores: the probability answers are to be connected to a same customer's question: $T_3 = \sum_{(a_k,a_{k'})\in L_a} MA_{kk'} \cdot (s_{a,a}(k,k') - \beta_3)$, L_a contains all candidate answer pairs from the server utterances.

4) The *None* alignment probabilities: the probability they have no alignment with any question. $T_4 = \sum_{j=1}^{J}\sum_{k=1}^{K} A_{jk} \cdot (n(k) - \beta_4)$.

Then, the final objective function is as follows:

$$maximize \quad T = \alpha_1 T_1 + \alpha_2 T_2 + \alpha_3 T_3 + \alpha_4 T_4 \tag{1}$$

where α_1, α_2, α_3, α_4, β_1, β_2, β_3, β_4 are the hyper-parameters.

Global constraints: To model the interaction among different decisions, we additionally set a series of global constraints on the three types of binary variables:
C1) A response could only reply to the former questions. The constraint could be formulated as:

$$A_{jk} = 0, \ \forall \ index(j) >= index(k) \tag{2}$$

where *Index(U)* denotes the index of the utterance U.
C2) If two customer utterances are similar, they must be aligned to at least one same server utterance. Conversely, they could not be aligned to a same server utterance if they are dissimilar.

$$MQ_{jj'} \cdot QQ_{jj'} + (1 - MQ_{jj'}) \cdot (1 - QQ_{jj'}) \geq 1 \tag{3}$$

where $QQ_{jj'} = \sum_{k=1}^{K} A_{jk} \cdot A_{j'k}$. In fact, it is an XNOR gate between $MQ_{jj'}$ and $QQ_{jj'}$. Therefore, C2 is a nonlinear operation, which is different from integer linear programming (ILP) models in other NLP tasks.
C3) Similar with **C2)** about another side.

$$MA_{kk'} \cdot AA_{kk'} + (1 - MA_{kk'}) \cdot (1 - AA_{kk'}) \geq 1 \tag{4}$$

where $AA_{kk'} = \sum_{j=1}^{J} A_{jk} \cdot A_{j'k}$.

4 Experiment

In this section, we present our experiment settings and results, which devote to answering the following questions: 1) *Is the joint disambiguation model able to obtain a better performance of utterance alignments compared with rule-based and matching-based methods?* 2) *Is the proposed model able to resolve the types of `1-M` and `Jump` alignments?*

4.1 Configurations

The dataset is randomly split into training (4741 dialogues, about 80%), validation (500 dialogues, about 10%) and testing set (500 dialogues, about 10%). The utterances are segmented into word sequences with Jieba[3] tool after some basic preprocessing such as convert all URLs to a special label. Hyper-parameters γ, α_1, α_2, α_3, α_4, β_1, β_2, β_3, β_4 are set to 0.5, 1.0, 1.0, 0.05, 1.0, 0.3, 0.5, 0.5, 0.5 respectively. The word embedding size and the hidden size are 128 for all deep learning models. Each position embedding size is 4. For a fair comparison, CNN based models set filter sizes as [3,4,5] and employ 42 filters. We used the Adam with learning rate 0.001.

[3] https://github.com/fxsjy/jieba

4.2 Baselines

The multi-round conversation in CS has some specific characteristics: 1) there are only two participants (customer and server); 2) the customer mainly poses the questions; and 3) the server mainly answer customer's questions. Therefore, we can simply obtain the utterance alignments using some heuristic rules based on the posing order of utterances. We utilize the following manual rules to align utterances which only consider the position information.
Rule-1: For a given server utterance, we choose the closest customer utterance as its alignment.
Rule-2: Rule-1 lacks the ability to handle *None* alignment type, which occupies a large proportion of alignments. Thus, another rule is proposed: if the closest customer utterance has been aligned to other utterances, we directly give the current server utterance the *None* label.

Rules	Utterance ID	Golden Alignments	Predicted Alignments
Rule-1	U_5	[U_4]	[U_4]
Rule-1	U_{12}	[None]	[U_{10}]
Rule-2	U_{12}	[None]	[None]
Rule-2	U_6	[U_2,U_3]	[None]

Table 1. Sample utterance alignments obtained by heuristic rules.

Two rules are adopted and their sampling results for the dialogue in Figure 1 are given in Table 1.
Matching-based model: We utilize semantic composition models such as C-NN and RNN (LSTM) for learning the representations of utterances, which also incorporate the position embeddings of the multi-turn dialogue into the representations.

4.3 Evaluation metrics

Based on the human-labeled alignments for each server utterance, we could calculate the precision (`P`), recall (`R`) and `F1` for utterance alignments. Considering that there are multiple alignments, we utilize the micro averaging to obtain the overall metrics for equally treating all utterance pairs.

4.4 Results and Discussion

The overall experimental results are shown in Table 2. The last two rows are the results of our proposed joint disambiguation models with integer programming (IP).

From the overall results, we can observe that: 1) The rule-based methods are not very bad, the overall F1 even exceeds the results of Match-CNN. 2) The matching-based methods have a better recall, that is, they have an advantage in obtaining more valid alignments. 3) From the above four rows, we believe

	Overall			None			One-to-One			One-to-Many			Jump		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Rule-1	45.2	42.1	43.6	14.7	14.7	14.7	**93.1**	**93.1**	**93.1**	**96.1**	42.1	58.6	0	0	0
Rule-2	58.0	54.0	55.9	**55.0**	**55.0**	**55.0**	61.9	61.9	61.9	66.9	29.3	40.8	0	0	0
Match(CNN)	52.4	59.7	55.8	35.5	39.9	37.6	**69.8**	**92.2**	**79.5**	**92.8**	70.2	79.9	**31.1**	53.9	39.4
Match(LSTM)	55.7	60.9	58.2	42.4	46.1	44.2	68.9	87.0	76.9	92.0	64.8	76.0	26.3	43.1	32.7
IP(CNN)	**58.9**	**66.6**	**62.5**	48.5	53.1	50.7	67.1	87.7	76.0	88.6	**76.8**	**82.3**	**32.4**	**65.4**	**43.3**
IP(LSTM)	**61.5**	**69.3**	**65.2**	**55.8**	**60.3**	**58.0**	62.8	82.3	71.3	89.8	**78.1**	**83.5**	29.8	**66.0**	**41.1**

Table 2. The precision (P), recall (R) and F1 (%) for overall and different alignment types on test data.

that it is very hard to obtain a satisfactory result merely relying on the position information or the utterance texts. 4) The proposed methods obviously exceed other methods, which demonstrates that the alignments of different utterances are correlated and interactional with each other. 5) In most cases, LSTM has a better composition semantics on spoken utterances.

From the result of different alignment types, we can observe that: 1) For the `None` type, the `Rule-1` is very bad because it always obtain an alignment for all utterances in any case. The `Rule-2` is the most competitive model which even better than all matching models except the proposed method. It indicates that the extra information such as utterance texts will help to work on *None* type. 2) For the `1-1` type, the `Rule-1` is outstanding. It is because that an answer usually follows its corresponding question in a dialogue. Our proposed joint model still outperforms better. The extra classification information such as restrictions on other utterances can help to judge whether to choose an alignment or not. 3) For the `1-M` type, the Rule-1 has the best precision but a worse recall. The proposed joint model performs best for recall and F1. It demonstrates that joint models are able to capture more potential alignments by utilizing global restrictions. 4) For the `Jump` type, the rule-based methods are broken because their assumption always chooses a nearest one. By modeling the utterance contents, the matching-based methods are able to deal with this alignment type in some extent. The proposed models outperform other models, which indicate that joint models are able to consider all relations among different utterance pairs.

In total, the proposed joint model obtain the best performance for overall results , especially for *1-M* and *Jump* alignments which are very hard for rule-based and matching-based methods.

4.5 Detailed Analysis

In this section, we analyze the effects of some core components in joint models.

At first, we validate the importance of position embeddings in matching-based methods. The experimental results in Table 3 compare the models with and without (w/o) it. Because of they can absorb the advantages of manual rules based on position information, the results with them perform better on most types (*Overall*, *One-to-One*, *One-to-Many* and *Jump*), except *None*.

With *None* type occupying the largest proportion 57%, we then compare different methods for it in IP. We compare the model with and without (w/o)

	Overall			None			One-to-One			One-to-Many			Jump		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
CNN o pos	45.1	**62.3**	52.3	44.1	58.0	50.1	42.5	71.6	53.3	63.2	58.9	61.0	14.6	**61.4**	23.5
LSTM o pos	45.8	61.4	52.4	**45.6**	**58.3**	**51.2**	42.3	68.9	52.4	63.2	56.7	59.7	13.9	56.9	22.4
CNN w pos	52.4	59.7	55.8	35.5	39.9	37.6	**69.8**	**92.2**	**79.5**	**92.8**	**70.2**	**79.9**	**31.1**	53.9	**39.4**
LSTM w pos	**55.7**	60.9	**58.2**	42.4	46.1	44.2	68.9	87.0	76.9	92.0	64.8	76.0	26.3	43.1	32.7

Table 3. The effects of position embeddings in matching-based methods.

	Overall			None			One-to-One			One-to-Many			Jump		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
IP(C) o None	52.9	60.2	56.3	34.6	39.0	36.7	**73.3**	**92.9**	**81.9**	**91.0**	76.8	83.3	**38.8**	63.7	**48.2**
IP(L) o None	55.0	62.1	58.3	39.6	44.4	41.9	70.9	89.1	79.0	90.7	76.8	83.2	35.9	62.7	45.7
IP(C)+Pipe	62.7	65.5	64.0	70.9	73.5	72.2	46.5	55.3	50.5	74.4	53.7	62.4	16.9	45.1	24.6
IP(L)+Pipe	**62.8**	65.9	64.3	**70.9**	**73.6**	**72.2**	46.7	56.0	50.9	74.6	54.9	63.3	17.7	48.4	25.9
IP(C) w None	58.9	66.6	62.5	48.5	53.1	50.7	67.1	87.7	76.0	88.6	76.8	82.3	32.4	65.4	43.3
IP(L) w None	61.5	**69.3**	**65.2**	55.8	60.3	58.0	62.8	82.3	71.3	89.8	**78.1**	**83.5**	29.8	**66.0**	41.1

Table 4. The effects of modeling coherence among utterances in UA. C and L denote CNN and LSTM respectively.

considering such part in IP (contains T_4 or not). In addition, we design a pipeline model (*+Pipe*), which first judges whether the utterance should align with *None* based on a threshold, and next utilize the IP models. Table 4 shows the experimental results. It demonstrates the IP models effectively deal with *None* alignment and overcome the problem of *1-M* and *Jump* alignments in other models.

5 Related Work

There are many tasks on dialogue analysis such as dialogue analysis state tracking [20], dialogue act classification [15],the speaker and addressee recognition [13], response generation [16] and other tasks. However, there is little research work paying attention to utterance alignments.

Utterance alignment relates to other alignment tasks in NLP, such as word alignment in machine translation. However, it is to align words rather than utterances and has fixed word size with infinite space of generated utterances. As a result, related approaches such as the HMM model [19] could not be directly applied to our task. Some previous approaches transform words into continuous space to achieve it [23, 17], but the utterances in dialogues still have different distributions from words. Moreover, the utterance alignment in CS deal has a larger linguistic unit and focus more on conversation analysis rather than sentence analysis.

And IP the proposed models employ to combine local features and global restrictions has received wide attention in other NLP tasks, such as semantic role labeling [14], syntactic and semantic dependency parsing [4], named entity disambiguation [10], sentiment analysis [11], summarization [21] and question answering [22, 24], etc. However, most of the aforementioned approaches apply linear constraints in joint disambiguation models. By contrast, there are nonlinear constraints in our model.

6 Conclusion

In this paper, we define a new task in real CS, utterance alignment, which devotes to aligning the utterances between customer and server in a dialogue. Where utterance alignments are divided into four types: *None*, *One-to-One*, *One-to-Many* and *Jump*. To model the mutual influence of different utterances as well as their alignments for *One-to-Many* and *Jump* alignments, we propose a joint model for UA, which models the task as a joint disambiguation problem with integer programming resolving and obtain better results.

7 Acknowledgement

This work is supported by the National Natural Science Foundation of China (No.61533018), the Natural Key R&D Program of China (No.2017YFB1002101), the National Natural Science Foundation of China (No.61702512, No.61806201) and the independent research project of National Laboratory of Pattern Recognition. This work was also supported by CCF-DiDi BigData Joint Lab and CCF-Tencent Open Research Fund.

References

1. Blunsom, P., Grefenstette, E., Kalchbrenner, N.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
2. Brown, P.E., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2) (1993), http://www.aclweb.org/anthology/J93-2003
3. Carlos, C.S., Yalamanchi, M.: Intention analysis for sales, marketing and customer service. Proceedings of COLING 2012: Demonstration Papers pp. 33–40 (2012)
4. Che, W., Li, Z., Hu, Y., Li, Y., Qin, B., Liu, T., Li, S.: A cascaded syntactic and semantic dependency parsing system. In: CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning. pp. 238–242. Coling 2008 Organizing Committee (2008), http://www.aclweb.org/anthology/W08-2134
5. Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., Zhou, M.: Superagent: A customer service chatbot for e-commerce websites. Proceedings of ACL 2017, System Demonstrations pp. 97–102 (2017)
6. Herzig, J., Feigenblat, G., Shmueli-Scheuer, M., Konopnicki, D., Rafaeli, A., Altman, D., Spivak, D.: Classifying emotions in customer support dialogues in social media. In: Meeting of the Special Interest Group on Discourse and Dialogue (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735 (1997)
8. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems. pp. 2042–2050 (2014)
9. Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. Computer Science (2014)

10. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 457–466. KDD '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1557019.1557073, http://doi.acm.org/10.1145/1557019.1557073
11. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: Proceedings of the 20th international conference on World wide web. pp. 347–356. ACM (2011)
12. Negi, S., Buitelaar, P.: Towards the extraction of customer-to-customer suggestions from reviews. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2159–2167. Association for Computational Linguistics (2015). https://doi.org/10.18653/v1/D15-1258, http://www.aclweb.org/anthology/D15-1258
13. Ouchi, H., Tsuboi, Y.: Addressee and response selection for multi-party conversation. In: EMNLP. pp. 2133–2143 (2016)
14. Punyakanok, V., Roth, D., Yih, W.t., Zimak, D.: Semantic role labeling via integer linear programming inference. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics (2004), http://www.aclweb.org/anthology/C04-1197
15. Reithinger, N., Klesen, M.: Dialogue act classification using language models. In: EuroSpeech (1997)
16. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
17. Tamura, A., Watanabe, T., Sumita, E.: Recurrent neural networks for word alignment model. ACL (1) **52**, 1470–80 (2014)
18. Tan, M., dos Santos, C., Xiang, B., Zhou, B.: Improved representation learning for question answer matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 464–473 (2016)
19. Vogel, S., Ney, H., Tillmann, C.: Hmm-based word alignment in statistical translation. In: Proceedings of the 16th conference on Computational linguistics-Volume 2. pp. 836–841. Association for Computational Linguistics (1996)
20. Williams, J., Raux, A., Ramachandran, D., Black, A.: The dialog state tracking challenge. In: Proceedings of the SIGDIAL 2013 Conference. pp. 404–413 (2013)
21. Woodsend, K., Lapata, M.: Automatic generation of story highlights. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 565–574. Association for Computational Linguistics (2010), http://www.aclweb.org/anthology/P10-1058
22. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Natural language questions for the web of data. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 379–390 (2012)
23. Yang, N., Liu, S., Li, M., Zhou, M., Yu, N.: Word alignment modeling with context dependent deep neural network. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 166–175. Association for Computational Linguistics (2013), http://www.aclweb.org/anthology/P13-1017
24. Zhang, Y., He, S., Liu, K., Zhao, J.: A joint model for question answering over multiple knowledge bases. In: Thirtieth AAAI Conference on Artificial Intelligence. pp. 3094–3100 (2016)