Multi-Point Semantic Representation for Intent Classification

Jinghan Zhang,[1,2]* Yuxiao Ye,[2,4] Yue Zhang,[3] Likun Qiu,[2,5]†
Bin Fu,[2] Yang Li,[2] Zhenglu Yang,[1] Jian Sun[2]
[1]Nankai University, China, [2]Alibaba Group, China, [3]Westlake University, China
[4]University of Cambridge, United Kingdom, [5]Minjiang University, China
jhzhang@mail.nankai.edu.cn, yy477@cl.cam.ac.uk, yue.zhang@wias.org.cn, likun.qiu@foxmail.com
{bingo.fb, ly200170, jian.sun}@alibaba-inc.com, yangzl@nankai.edu.cn

Abstract

Detecting user intents from utterances is the basis of natural language understanding (NLU) task. To understand the meaning of utterances, some work focuses on fully representing utterances via semantic parsing in which annotation cost is labor-intentsive. While some researchers simply view this as intent classification or frequently asked questions (FAQs) retrieval, they do not leverage the shared utterances among different intents. We propose a simple and novel multi-point semantic representation framework with relatively low annotation cost to leverage the fine-grained factor information, decomposing queries into four factors, i.e., topic, predicate, object/condition, query type. Besides, we propose a compositional intent bi-attention model under multi-task learning with three kinds of attention mechanisms among queries, labels and factors, which jointly combines coarse-grained intent and fine-grained factor information. Extensive experiments show that our framework and model significantly outperform several state-of-the-art approaches with an improvement of 1.35%-2.47% in terms of accuracy.

Introduction

As intelligent assistants and customer service robots have been widely applied, there is a need to improve their capacity to understand spoken or written utterance of users. A challenging step is to represent the meaning of user utterances.

Recent research has been focusing on two approaches. One is to fully represent the meaning of sentences in tree or graph structures, such as Combinatory Categorial Grammar(CCG) (Steedman 2000) and Abstract Meaning Representation(AMR) (Banarescu et al. 2013). We refer to this line of research as the *full representation framework*. The full representation framework is usually viewed as a semantic parsing task, which relies on sufficient labeled data. However, data annotation for semantic parsing is labor-intensive and time consuming. Considering the heavy workload of the annotation task, it is unimaginable to construct an annotated corpus for each domain or customer.

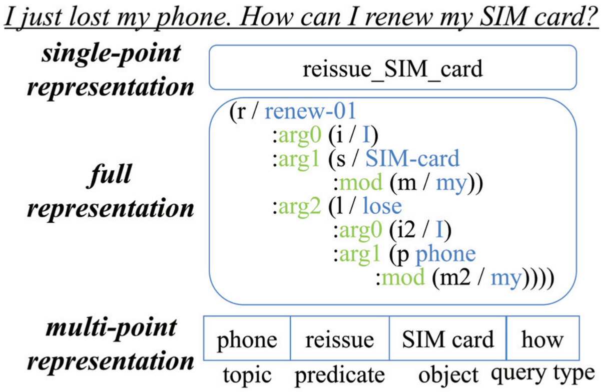


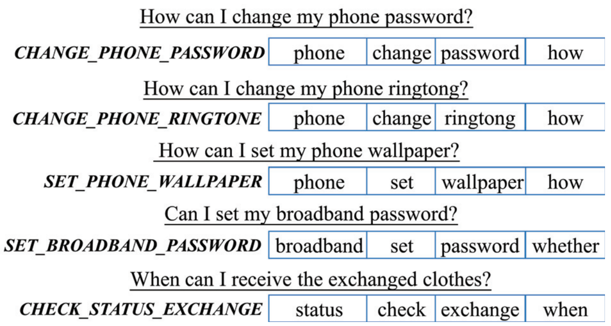
Figure 1: One sentence is represented by three representation frameworks. In “full representation”, words in green are the semantic roles, while in blue are the concepts.

The other is the *single-point representation framework*. Some work (Niu et al. 2019; Zhang et al. 2019) views language understanding as a query classification problem (and in some cases jointly trained with additional tasks such as entity detection and slot filling), where class labels are intent names or standard FAQs (frequently asked questions). We refer to this line of research as the *single-point representation framework*. The single-point representation framework is usually understood as intent classification or FAQ retrieval task, in which data annotation is relatively easy. However, since the meaning of a sentence is mainly represented by an intent label, we can not utilize the shared text spans of different sentences belonging to difference intents, e.g., *ORDER* as in *ORDER_PRODUCT* and *ORDER_TAXI*.

We proposes a simple and novel annotation framework, called the *multi-point representation framework*, which lies somewhere between the two aforementioned frameworks. Under this framework, we try to distinguish different intents through four types of key factors, i.e., topic, predicate, object/condition, and query type. We focus on differentiating intents by four key concepts instead of fully representing

[*] This work was done when Jinghan Zhang was interning at Alibaba Group,China.

[†] Likun Qiu is the corresponding author of the paper.



How can I change my phone password?

CHANGE_PHONE_PASSWORD	phone	change	password	how

How can I change my phone ringtong?

CHANGE_PHONE_RINGTONE	phone	change	ringtong	how

How can I set my phone wallpaper?

SET_PHONE_WALLPAPER	phone	set	wallpaper	how

Can I set my broadband password?

SET_BROADBAND_PASSWORD	broadband	set	password	whether

When can I receive the exchanged clothes?

CHECK_STATUS_EXCHANGE	status	check	exchange	when

Figure 2: The multi-point annotation scheme. Intents are in a bold, italic font. The boxes are “topic”,“predicate”,“object/condition”,“query type” from left to right.

the main meaning of sentences. The difference of three representation frameworks is illustrated in Figure 1.

Among the four factor categories, both “predicate” and “object” are the main elements of a proposition in logic and semantics. “Condition” stands for the predicate of another possible proposition, e.g., “exchange” in “check_status_exchange”. “Topic” is used to differentiate different topics. “Query type” indicates different ways to ask a question. Examples of our annotation framework are illustrated in Figure 2.

Our idea is inspired by the fact that the number of possible sentences is infinite, and thus to fully represent the meaning of all sentences is infeasible, or deep semantic representations must be too general. However, in a certain domain or scenario, the possible semantic space is limited, and thus we could differentiate all the intents by a limited number of key concepts and do not need to represent each intent directly with a full representation.

Our framework has several advantages. First, the annotation cost is relatively small and thus it can be applied in various scenarios. Second, by decomposing each intent into several key factors, utterances standing for the same factor but belonging to different intents can be shared among different intents in training. For example, *SET_PHONE_WALLPAPER* and *SET_BROADBAND_PASSWORD* have the same predicate *SET*. Although these two intents belongs to different classes, utterances about *SET* in two intents can both help training the model. Third, our multi-point framework enhances the differentiation of queries belonging to similar intents, which share many factors and only have one different factor. For example, *CHANGE_PHONE_PASSWORD* and *CHANGE_PHONE_RINGTONG* fall into distinct object/condition categories (e.g., *password* and *ringtong*), resulting in differentiating intents.

To leverage the advantage of multi-point representation, we propose a Compositional Intent Bi-Attention (CIBA) classification model, which consists of query-factor attention, query-label attention and label-factor aligned attention mechanisms. The query-factor attention mechanism measures the compatibility of query and factors, and captures the implicit shared information of factors among different intents. The query-label attention module extracts the relation between factor-attention query representations and intents. The label-factor aligned attention mechanism directly matches intents with predicted factors and thus distinguishes intents through the difference of factors.

A comprehensive evaluation is conducted on the public China National Conference on Computational Linguistics(CCL) dataset and three real-world datasets. The experimental results demonstrate that our proposed framework significantly outperforms several state-of-the-art approaches with an improvement of 1.35%-2.47% in terms of accuracy. The contributions of our work can be summarized as follows:

1. We propose a novel multi-point annotation framework to represent the main meaning of queries at a relatively fine-grained level without heavy annotation or parsing cost.
2. Based on this framework, we propose a composition-intent bi-attention classification model to jointly utilize the information of shared factors and different factors.

Related Work

Full representation framework Combinatory Categorial Grammar(CCG) (Steedman 2000) and Abstract Meaning Representation(AMR) (Banarescu et al. 2013) are employed to fully represent the deep semantic meaning of sentences in tree or graph structures. Perera et al.(2018) proposed the Alexa Meaning Representation Language (AMRL), which is a compositional graph-based semantic representation to fully represent sentences including fine-grained types, properties, actions and roles. In AMRL, the intent is composed of some of those factors, for example, *PlaybackAction object@MusicRecording*. However, it is designed for parsing and not utilizes the fine-grained compositional information for text classification. Compared with AMRL, our annotation framework is not only suitable for task-oriented dialog, but also question answering task and knowledge base question answering(KBQA) task.

Single-Point Representation Framework Most work views language understanding as a query classification problem, where class labels are intent names or standard FAQs. Deep learning models (Kim 2014; Yang et al. 2016; Lai et al. 2015) have dominated the literature. Joulin et al.(2016) used bag of words and linear network as the encoder. Kim(2014) introduced convolutional nerual network (CNN) to extract local features for text classification. Yang et al.(2016) explored a hierarchical attention model with Long-Short Term Memory networks (LSTMs) for document classification. Lai et al.(2015) and Zhou et al.(2016) combined recurrent and convolution layers to acquire the sentence representation. Graph Convolutional Network (GCN) and its variations (Zhang, Liu, and Song 2018; Yao, Mao, and Luo 2018; Wu et al. 2019; Haonan et al. 2019) have been applied in text classification. Recently, some pre-trained language models (Peters et al. 2019; Devlin et al. 2019; Yang et al. 2019) have achieved remarkable improvements.

Recent work (Tang, Qu, and Mei 2015; Zhang et al. 2018; Wang et al. 2018) also made use of label embeddings to leverage label information. For example, Wang et al.(2018) proposed the Label-Embedding Attentive Model (LEAM), which directly use label information in constructing the text-sequence representation. We take this idea further that we leverage coarse-grained label and fine-grained factor information together to capture the textual and compositional information of intents.

Multi-Point Representation Framework Some work (Gupta et al. 2018; Vedula et al. 2019) explores the multi-point representation framework, which decomposes query with action and object to represent a simple label such as *BUY_TICKETS* and *BOOK_RESTAURANT*. However, the annotation frameworks can be insufficiently informative in real-world tasks. Despite of the efforts, recent models still simply use intents as class labels and ignore the compositional information. Different from those works, our multi-point annotation framework aims to differentiate intents by four factor categories and enhance the fine-grained shared and different information between intents. Our proposed framework is easy to reuse in new scenarios due to low annotation cost. We also investigate a model to exploit fine-grained compositional information, which no existing work considers.

Multi-Point Annotation Framework

In our annotation framework, an intent or a query is decomposed into four factors:

- **Topic** represents the specific topics among the whole intents collection, such as *phone*, *broadband* in telecom service, and *accumulation fund* and *medical insurance* in city service.
- **Predicate** represents the most important action in the intent, such as *cancel*, *change*, *charge*, et. al.
- **Object/Condition** represents the most important object or condition of the intent, such as *password* (object), *tickets* (object) and *exchange* in *check_status_exchange* (condition).
- **Query Type** represents the type of queries. Each intent is combined with a query type to construct a query, so that each query only belongs to one type. In our framework, there are 10 query types as shown in Table 1, such as *how*, *what* or *why*.

In the annotation framework, each intent is decomposed into three factors, i.e., topic, predicate and object/condition, while each query is decomposed into the above four factors. The topic, predicate and object/condition of the query are inherited from the intent it belongs to. For a query or intent, each factor can be default. The factor in this situation is defined as *none*. The topic, predicate and object/condition often depend on the specific scenario (e.g. news, banking, telephone business).

The cost of annotation and decomposition is very low. Only the factors of intents need to be annotated, and the query type is automatically tagged with linguistic rules. For example, there are 10859 queries and 53 intents in the CCL dataset, but we merely annotate 53 intents in our framework and 10859 queries inherit three factors from intents. While in previous framework, 10859 queries need to be decomposed.

Query Type	Example Query
WHATIS	What is three-point shot in a basketball game?
HOW	How can I change my password?
WHERE	Where to change my password?
WHEN	When can I change my password?
NUMBER	How long is my password valid for?
ENUMERATE	What passwords have you ever used?
INFORMATION	What kind of information should I provide if I want to change my password?
WHY	Why can't I change my password?
WHETHER	Can I change my password?
NONE	None of above.

Table 1: Query types and examples.

The factor category and composition in our framework are flexible and reconfigurable, which can be omitted to adapt different scenarios. For example, if there is no definite topic existing, the topic category can be omitted.

Model

As illustrated in Figure 3, our model achieves query classification by combining attentions between query and intent compositions. Each query is encoded by two CNN layer, in which one is a normal convolutional layers and the other is viewed as a gate. Each factor is fed into an average pooling layer to obtain a dense representation. A query-factor attention layer captures the semantic relation between query and factor categories which leverages the shared fine-grained factor information. A query-label attention layer extracts the label-level feature of query with coarse-grained label information. A label-factor aligned attention layer, which combines the fine-grained factor level and coarse-grained label level information, focuses on aligning predicted factors and intents. Finally, in the output layer, the intent classification and four factor classifications are synthesized to yield the final loss under multi-task learning.

Encoder Layer

Query Encoder Pre-trained embeddings such as BERT (Devlin et al. 2019) and word2vec (Mikolov et al. 2013) are employed to construct the word embeddings. BERT and word2vec embeddings are connected to embed the query which is represented as $Q_{emb} \in \mathbb{R}^{l \times e}$, where l is the query length and e is the total word embedding dimension.

Gated Tanh Units (GTU) and Gated Linear Units (GLU) have shown the effectiveness of gating mechnisms in language modeling (Dauphin et al. 2017). Given a query embeddings $Q_{emb} \in \mathbb{R}^{l \times e}$, we employ two CNN layers as GLU to encode the query. More specifically, we use 1D convolution layer with kernel $\{200, 200, 200\}$ of filter sizes $\{2, 3, 4\}$ to compute the n-gram features at different granularities at each position respectively, which are concatenated to form the query representation $Q_{cnn1} \in \mathbb{R}^{l \times 600}$. Another CNN connected with a ReLU function is employed as a gate

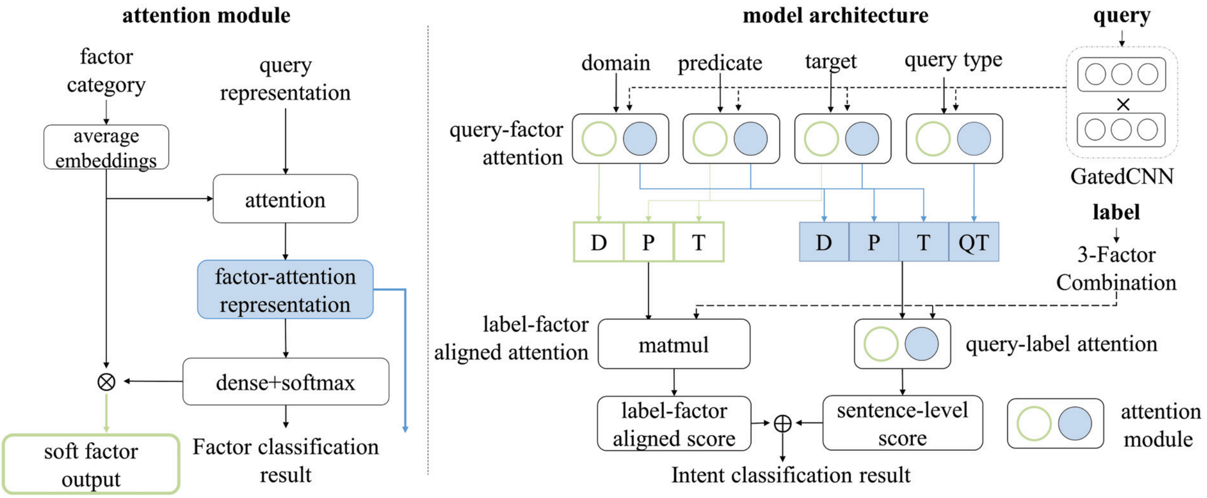


Figure 3: CIBA model architecture. T,P,O,QT means topic, predicate, object/condition, query type, respectively. The green and blue blocks on the right are the soft factor output and factor-attention representation on the left, respectively.

to acquire $G_{cnn2} \in \mathbb{R}^{l\times 600}$ which controls the feature relevance with query. The query representation is computed as $Q_{rep} = Q_{cnn1} \times G_{cnn2}$ and it is fed into a dense layer to get the final query representation $Q_{rep} \in \mathbb{R}^{l\times d}$, where d is the encoding size.

Labels and Factors According to (Wang et al. 2018), labels' lexical information directly contributes to constructing text representation in the text-label joint latent space, which brings a remarkable improvement for text classification. To utilize the lexical information in labels and factors, we encode factors and labels with an average embedding layer.

Given the token embeddings of each factor, we average them to get a factor representation. The factor embedding is the connection of BERT contextual sentence embeddings and the average word embeddings. We acquire the factors' representation of different factor categories, $T_{rep} \in \mathbb{R}^{n_d\times e}$, $P_{rep} \in \mathbb{R}^{n_p\times e}$, $O_{rep} \in \mathbb{R}^{n_t\times e}$, $QT_{rep} \in \mathbb{R}^{n_{qt}\times e}$, where we use T, P, O, QT to represent topic, predicate, object/condition, query type, respectively and n_t, n_p, n_o, n_{qt} represent the number of T, P, O, QT respectively.

As for the intent label, we leverage factor embeddings of each intent, rather than using intent's token embeddings directly, because factor combination representation can obtain more fine-grained information about each intent. We acquire label embeddings $L_{rep} \in \mathbb{R}^{n_l\times 3e}$ by the connection of three factors annotated with intents, where n_l is the number of labels.

Query-Factor Attention Layer

A challenging issue is to utilize factor information to capture the underlying relevance between queries and factors for improving intent classification. We introduce the bi-directional attention mechanism (Seo et al. 2016) to capture the relation between queries and labels. Given $Q \in \mathbb{R}^{l\times d}$ as Q_{ref} and $F \in \mathbb{R}^{n_f\times d}$ where n_f is the number of factor, for each factor category F, we first employ a dense layer with ReLU activation to fix the factor into the same encoding size as the query and compute a similarity matrix S between factor F and query Q representation as:

$$S_{t,j} = \alpha\left(Q_t, F_j\right), \alpha\left(q, f\right) = \omega \times [q; f; q \cdot f] \quad (1)$$

Q_t is the t-th column vector of query, F_j is the j-th column vector of factor, and ω is a trainable weight vector. $\cdot$ means element-wise multiplication and [;] is vector concatenation across rows.

Query-to-factor attention is made, and signifies which factors are the most relevant to each query words:

$$A^F = softmax(S) \in \mathbb{R}^{l\times n_f}, A_{Q2F} = A^F F \in \mathbb{R}^{l\times d} \quad (2)$$

where l is the query length and n_f is the factor category number.

Factor-to-query attention indicates which query words have the closest similarity to one of the factors. The attention weight on a query word is obtained by the max value of each column in S.

$$A^Q = softmax(maxcol(S)) \in \mathbb{R}^l \quad (3)$$

The attended query vector, which is the weighted sum of the most important query words related to the factors, is computed as below and then tiled n_f times:

$$\hat{q} = \sum_t A^Q Q_t \in \mathbb{R}^d, A_{F2Q} = tile_{n_f}(\hat{q}) \in \mathbb{R}^{n_f\times d} \quad (4)$$

Finally, we obtain the factor-aware query representation Q_F, which consists of the origin query representation, factor-attended query representation, factor-updated query representation and the attended query vector.

$$Q_F = [Q; A_{Q2F}; Q \cdot A_{Q2F}; Q \cdot A_{F2Q}] \in \mathbb{R}^{l\times 4d} \quad (5)$$

We summarize the process from equation 1 to equation 5 as an attention function $\Phi(Q, F) = Q_F$, whose inputs are query and factor representations and output is the factor-aware query representation.

For each factor category, we computed topic-aware query representation Q_T, predicate-aware query representation Q_P, object/condition-aware query representation Q_O, query type-aware query representation Q_{QT} via the attention function $\Phi(Q, T)$, $\Phi(Q, P)$, $\Phi(Q, O)$, $\Phi(Q, QT)$, respectively.

The factor-aware representation is fed into a dense layer with softmax activation for the factor classification:

$$P_F = softmax(WQ_F + b)$$
$$\mathcal{L}_F = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_d}(\mathbb{I}(y_F^i = j)log(p_j^i)) \tag{6}$$

where W and b is trainable parameters and $\mathbb{I}$ is an indicator function. Through the same function as (6), the P_T, P_P, P_O,P_{QT},$\mathcal{L}_T$, $\mathcal{L}_P$,$\mathcal{L}_O$ and $\mathcal{L}_{QT}$ are computed.

In addition, we acquire four factor-attention query representations, which are combined with factor-aware query representation to get the final query representation related to factors, computed as $Q_{factors} = [Q_T; Q_P; Q_O; Q_{QT}]$.

Query-Label Attention Layer

The query representation and labels are fed into an attention layer to extract the label-aware query representation Q_{label}, computed as $Q_{label} = \Phi(Q_{factors}, L_{rep})$.

Q_{att} serve as connection of $Q_{factors}$ and Q_{label} to acquire hierarchical query representation.

label-factor aligned Attention Layer

Through the query-factor attention layer, factor classifications predict four factors of each query. In consideration of the inter relation between factors and intent, a sum layer is applied to extract the relation between predicted factors and labels. The predicted factor representation is computed as $Q_{pre}^F = P_F F_{rep}$.

We acquire the soft intent representation by combining three soft predicated factor representations which intent contains, i.e., Q_{pre}^T for topic, Q_{pre}^P for predicate, Q_{pre}^O for object/condition. Then, we match the correct intent label by the correlation of soft intent representation and overall labels to get the final factor-level score.

$$Q_{pre}^L = [Q_{pre}^T; Q_{pre}^P; Q_{pre}^O]$$
$$att_F = Softmax(Q_{pre}^L L_{rep}) \in \mathbb{R}^{n_l} \tag{7}$$

Output Layer

Given the query representationQ_{att} related to overall factors and labels, the query-level score att_S is derived through a dense layer with softmax activation.

$$att_S = softmax(WQ_{att} + b) \tag{8}$$

The final intent scores P_L is summed of sentence-level score att_S and the factor-level score att_F.

$$P_L = softmax(\alpha_S att_S + \alpha_F att_F) \tag{9}$$

where W and b are trainable parameters. α_S and α_F are the weights of sentence-level and factor-level scores, which is 1.0 in our settings.

Given a set of training data, the intent label classification loss is computed as:

$$\mathcal{L}_L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_l}(\mathbb{I}(y_L^i = j)log(p_j^i)) \tag{10}$$

There are five classification processes in our architecture, i.e., topic classification, predicate classification, object/condition classification, query classification and intent classification, in which the former four classifications contribute to improving intent classification. The training objective is to minimize the weighted sum of overall losses:

$$\mathcal{L} = \mathcal{L}_L + \alpha_T\mathcal{L}_T + \alpha_P\mathcal{L}_P + \alpha_O\mathcal{L}_O + \alpha_{QT}\mathcal{L}_{QT} \tag{11}$$

α are weights of different tasks, which are 1.0 in our settings.

Experiments

We conduct experiments on four chinese intent classification datasets collected from real-world dialogue and QA system.

Settings

Datasets We evaluate our CIBA on four datasets. The detail statistics are presented in Tabel 2. The CCL dataset is a public Chinese query classification dataset in the domain of telecommunication released for the China National Conference on Computational Linguistics (CCL) 2018 Shared Task 1 (Sun et al. 2018), excluding samples with labels such as GREETINGS and COMPLAINTS. The TELE, ECOM and CitySrv are chinese task-oriented query datasets about telecommunication, e-commerce marketing promotions and city service, respectively, which are collected from real-world dialogue and QA systems.

In each dataset, intents are decomposed by trained annotators on the basis of our proposed annotation framework.

For each query, the intent label is annotated by trained annotators, and the labels of factor categories except query type are inherited from the intent. The label of query type is tagged automatically. Test sets of the last three datasets are selected from user queries in real-world dialogue and QA systems in one week.

Baselines We compare our proposed model with TextCNN (Zhou et al. 2016), LEAM (Wang et al. 2018) and L-Mixed (Sachan, Zaheer, and Salakhutdinov 2019), which are at the top of several text classification benchmarks. Besides, we also compare with GatedCNN for a fair comparison.

Implementation Details We use 300-dimensional word embeddings pre-trained with word2vec (Pennington, Socher, and Manning 2014) and 768-dimensional textual embeddings pre-trained by BERT-BASE (Devlin et al. 2019). For Chinese, we use character-level embeddings. We set the query sentence length to 40 for CitySrv dataset and 30 for other datasets. The hidden sizes of each unit are selected from the set [128,300]. A dropout layer is applied with its rate selected from [0.2,0.5]. Our model is optimized through Adam (Kingma and Ba 2014) with a learning rate

Dataset	Train	Test	Intent	T	P	O+C	QT
CCL	10859	2748	53	0	14	22	10
TELE	31188	1783	60	5	36	23	10
ECOM	23739	1321	107	11	64	63	10
CitySrv	27459	376	162	17	49	99	10

Table 2: Statistics of datasets. T,P,O+C,QT are topic, predicate, object/condition, query type, respectively.

Methods	**CCL**	**TELE**	**ECOM**	**CitySrv**
TextCNN	0.9308	0.8918	0.8388	0.8107
GatedCNN	0.9275	0.8917	0.8455	0.8160
LEAM	0.9287	0.8721	0.8441	0.8080
L-Mixed	0.9337	0.8979	0.8440	0.8053
BERT	0.9458	0.9058	0.8543	0.8165
CIBA(word only)	0.9436	0.9089	0.8546	**0.8373**
CIBA(word+BERT)	**0.9472**	**0.9226**	**0.8622**	0.8320

Table 3: Results on four datasets under ACC metric. Through t-test, there were significant differences between our proposed model and all baselines ($P<0.05$)

as 1e-3 and L2-regularizer with 1e-3. The kennel size of the encoding layer is [2,3,4], and the filter size is 200. For each dataset, we randomly select 10% training data as the development dataset, train each method 5 times for each dataset with the best result on the test dateset reported.

Results

Table 3 shows the results on the test data of each dataset. Our proposed model outperforms all the baselines with new state-of-the-art performance. For the purpose of fair comparison, we also evaluate CIBA with only word embeddings, which is represented as "CIBA(word only)". Compared with our baselines, "CIBA(word only)" yields remarkable improvement and outperforms the best result by 0.99% on the CCL dataset, 1.10% on the TELE dataset, 1.06% on the ECOM dataset, and 2.13% on the CitySrv dataset, respectively. The results reveal that CIBA strongly captures relation between query and intents and differentiate intents via textual and compositional intent information, i.e., coarse-grained label information and fine-grained factor information.

Besides, we further combine with BERT pre-trained textual embeddings to form the word embeddings which captures the underlying language feature. "CIBA" achieves best results which reaches 0.36%-1.37% improvement with "CIBA(word only)" on each dataset except CitySrv dataset. Within CitySrv data, the boundary of each factor category is the most definite in all datasets.

Discussion

Methods	**ACC**
CIBA	0.9089
-LA	0.9044
-LF	0.9060
-LA-LF	0.8983
-FA-LF	0.8977
CIBA(only topic)	0.8883
CIBA(only predicate)	0.8917
CIBA(only object/condition)	0.8962
CIBA(only query type)	0.8867

Table 4: Ablation study. "FA","LA","LF" represent query-factor attention mechanism, query-label attention mechanism and label-factor aligned attention mechanism, respectively.

Ablation Study As illustrated in Table 4, we conduct ablation experiments to evaluate different modules, i.e., query-factor attention mechanism(FA), query-label attention mechanism(LA) and label-factor aligned attention mechanism(LF). Once removing the query-label attention module or the label-factor aligned attention module, the performance drops with 0.45% and 0.29%, respectively. "-LA-LF" represents merely employing the fine-grained factor information, while "-FA-LF" represents only leveraging the coarse-grained label information. With the 1.12% drop when removing factor information, it is identified that intent composition effectively captures the relation between intents and queries and differentiates intents. The results indicate that coarse-grained label information and fine-grained factor information are complementary for query representation with multiple textual and compositional features.

We investigate factor effect by employing single factor independently in our model. CIBA under this setting only uses the query-factor mechanism without coarse-grained label information. The results reveal the effect of each factor categories. Specifically, the object/condition factor reserves most intent information thus achieves the best result, compared with other factor categories. In contrast, "CIBA(only query type)" performs worst without any label information.

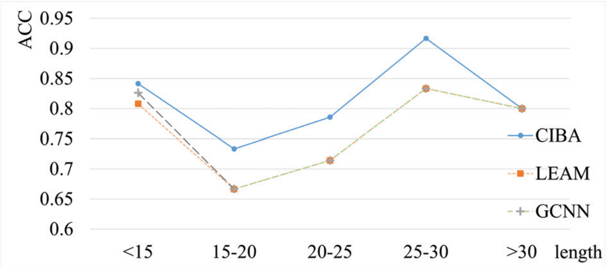


Figure 4: Analysis of the sentence length. The abscissa axis is the range of query length and the vertical axis means the accuracy of query with each range.

Analysis of Query Length We conduct some experiments to explore the effect on query length as illustrated in Figure 4. The query number drops quickly when the query length increases. Compared with LEAM and GCNN, our model achieves superior results on all query length scope. CIBA

Model	True Label	Predicted Label	MEC	EC
GatedCNN	endowment_insurance	endowment_insurance_employee	8	8
LEAM	endowment_insurance	endowment_insurance_employee	8	8
CIBA	endowment_insurance	endowment_insurance_employee	**0**	**0**
GatedCNN	medical_insurance_insure_children	medical_insurance_insure_urban_and_rural_residents	3	3
LEAM	medical_insurance_insure_children	medical_insurance_insure	2	5
	medical_insurance_insure_children	medical_insurance_insure_urban_and_rural_residents	2	
CIBA	medical_insurance_insure_children	medical_insurance_insure_urban_and_rural_residents	**1**	**1**

Table 5: Examples of confusing intents. The **MEC**(mapping error count) is the count of mapping true label to predicted label and **EC**(error count) is the count of predicting incorrect label.

Intents/Factors	number
endowment_insurance	52
endowment_insurance_employee	50
employee	1645

Table 6: Statistics of intents and factors of confusing intents pair.

captures the fine-grained factor information in the query and differentiates intents not only with textual feature, but also with compositional feature. The result demonstrates the effectiveness of CIBA on both long and short queries.

Confusing Intent A confusion intent pair is defined two intents which are different by one factor in one topic. In the Figure 5, the error count of the most confusing intent pair drops to 0 with the factor effect in CIBA, which suggests that our proposed model effectively differentiates intents with difference of factor composition. When the mapping error count(MEC) drops, the error count(EC) of this label drops synchronously. Compared with these baselines, our model effectively reduces the mistakes of confusing intents to improve the intent classification performance.
To deeply investigate the effect of intent composition, we also calculate the count of intents and factors of confusing intent pair, while the count of *endowment_insurance* and *endowment_insurance_employee* is low, the object *employee*'s number is 1645, which is the most reused in object/condition category. Thus, utterances about *employee* contribute to pointing "employee" in queries. As a result, *endowment_insurance* and *endowment_insurance_employee* are distinguished.

Factor Accuracy As shown in Figure 5, the accuracy of topic category is highest in three real-world datasets. The reason is that the boundary of topic category is clearest both in the data and intents. The accuracy of each factor category performs better than 0.83 in different datasets, which indicates that the factor classification provides precise fine-grained information.

Performance on English Dataset As shown in Table 7, we also evaluated our proposed model on SNIPS dataset(Coucke et al. 2018), and achieved 0.9772 under the metric of accuracy which outperformed state-of-the-art models by 0.42% without slot information. In the existing English public task-oriented datasets such as SNIPS and ATIS(Tur, Hakkani-Tür, and Heck 2010), the intent number is much smaller than ours. Specifically, SNIPS has 7 intents and ATIS has 18 intents, while the smallest number of intents in our datasets is 53. "Single-point representation" methods perform well in the existing English task-oriented datasets, which have few intents and is simpler. The result demonstrates that our model captures underlying relation and shares information between intents via "multi-point representation" although they are few.

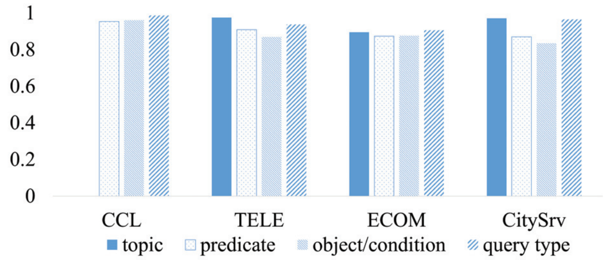


Figure 5: Factor accuracy.

Methods	**ACC**
Atten.-Based (Liu and Lane 2016)	0.9670
Joint Seq(Hakkani-Tür et al. 2016)	0.9690
Sloted-Gated(Goo et al. 2018)	0.9686
CAPSULE-NLU (Zhang et al. 2019)	0.9730
CIBA(word only)	0.9772

Table 7: Results on SNIPs dataset under the metric of ACC. The results of our baselines are as they reported on their papers.

Conclusion

We introduced a novel multi-point annotation framework and a Compositional Intent Bi-Attention (CIBA) classification model for intent classification. Our framework decomposes intents and queries into four factors, i.e., topic, predicate, object/condition, query type, and our model is designed to leverage such compositional information by jointly combining the coarse-grained intent and fine-grained factor information under multi-task learning. Extensive experiments validate the effectiveness of leveraging intent composition for query classification compared with the state-of-the-art approaches. Based on our annotation framework, an intent network could be constructed by linking intents via shared factors. The intent network contributes to inheriting contextual information from corresponding factors which are decomposed from user queries of multi-turn dialogue. We can also detect missing factors and ask users to do clarification to complete user intents when an utterance is incomplete or ambiguous.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant No.61572245, U1636116, 11431006, the Research Fund for International Young Scientists under Grant No. 61750110530, and the Ministry of education of Humanities and Social Science project under grant 16YJC790123.

References

Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract meaning representation for sembanking. In *LAW*, 178–186.

Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Dauphin, Y. N.; Fan, A.; Auli, M.; and Grangier, D. 2017. Language modeling with gated convolutional networks. In *ICML*, 933–941. JMLR. org.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 4171–4186.

Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL*, 753–757.

Gupta, S.; Shah, R.; Mohit, M.; Kumar, A.; and Lewis, M. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *EMNLP*, 2787–2792.

Hakkani-Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y.-N.; Gao, J.; Deng, L.; and Wang, Y.-Y. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, 715–719.

Haonan, L.; Huang, S. H.; Ye, T.; and Xiuyan, G. 2019. Graph star net for generalized multi-task learning. *arXiv preprint arXiv:1906.12330*.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. In *EACL*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*.

Liu, B., and Lane, I. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Niu, P.; Chen, Z.; Song, M.; et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Perera, V.; Chung, T.; Kollar, T.; and Strubell, E. 2018. Multi-task learning for parsing the alexa meaning representation language. In *AAAI*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2019. Deep contextualized word representations. In *NAACL*, 2227–2237.

Sachan, D. S.; Zaheer, M.; and Salakhutdinov, R. 2019. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *AAAI*, volume 33, 6940–6948.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. In *ICLR*.

Steedman, M. 2000. *The syntactic process*, volume 24. MIT press Cambridge, MA.

Sun, M.; Liu, T.; Wang, X.; Liu, Z.; and Liu, Y. 2018. *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer.

Tang, J.; Qu, M.; and Mei, Q. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, 1165–1174. ACM.

Tur, G.; Hakkani-Tür, D.; and Heck, L. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, 19–24. IEEE.

Vedula, N.; Lipka, N.; Maneriker, P.; and Parthasarathy, S. 2019. Towards open intent discovery for conversational text. *arXiv preprint arXiv:1904.08524*.

Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; and Carin, L. 2018. Joint embedding of words and labels for text classification. In *ACL*, 2321–2331.

Wu, F.; Zhang, T.; Souza Jr, A. H. d.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying graph convolutional networks. In *ICML*.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *NAACL*, 1480–1489.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yao, L.; Mao, C.; and Luo, Y. 2018. Graph convolutional networks for text classification. In *AAAI*.

Zhang, H.; Xiao, L.; Chen, W.; Wang, Y.; and Jin, Y. 2018. Multi-task label embedding for text classification. In *EMNLP*, 4545–4553.

Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. S. 2019. Joint slot filling and intent detection via capsule neural networks. In *ACL*.

Zhang, Y.; Liu, Q.; and Song, L. 2018. Sentence-state lstm for text representation. In *ACL*, 317–327.

Zhou, P.; Qi, Z.; Zheng, S.; Xu, J.; Bao, H.; and Xu, B. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *COLING*, 3485–3495.