Auto-Dialabel: Labeling Dialogue Data with Unsupervised Learning

Chen Shi[1] **Qi Chen**[2] **Lei Sha**[1] **Sujian Li**[1] **Xu Sun**[1]
Houfeng Wang[1] **Lintao Zhang**[2]
[1]MOE Key Lab of Computational Linguistics, Peking University, Beijing, China
{shichen, shalei, lisujian, xusun, wanghf}@pku.edu.cn
[2]Microsoft Research Asia, Beijing, China
{cheqi, lintaoz}@microsoft.com

Abstract

The lack of labeled data is one of the main challenges when building a task-oriented dialogue system. Existing dialogue datasets usually rely on human labeling, which is expensive, limited in size, and in low coverage. In this paper, we instead propose our framework *auto-dialabel* to automatically cluster the dialogue intents and slots. In this framework, we collect a set of context features, leverage an autoencoder for feature assembly, and adapt a dynamic hierarchical clustering method for intent and slot labeling. Experimental results show that our framework can promote human labeling cost to a great extent, achieve good intent clustering accuracy (84.1%), and provide reasonable and instructive slot labeling results.

1 Introduction

Building a task-oriented dialogue system is challenging. In real world, unlabeled dialogue data is usually available for companies who has interactive platform with users. Based on these unlabeled data, the well-known sequence-to-sequence framework (Sutskever et al., 2014; Cho et al., 2014) is widely used in dialogue response generation (Vinyals and Le, 2015; Sordoni et al., 2015; Shang et al., 2015), but it can not handle task-oriented scenario well since it needs accuracy instead of fluency. Generally, a task-oriented dialogue system needs to realize user's *intent*, which means *user's current goal in a dialogue session*. To fulfill this intent, the system usually needs several key information (*slot*). As shown in Figure 1, a dialogue utterance is labeled with *intent* flight, and several *slots* such as its from_location, arrive_time. Training a task-oriented dialogue system usually needs abundant such labeled data.

Existing well-known dialogue datasets are mostly human-labeled, such as ATIS (Hemphill et al., 1990), DSTC (Williams et al., 2013), Frames (El Asri et al., 2017), and the Stanford dataset (Eric et al., 2017). Human-labeled datasets are expensive to produce, limited in size, and restricted to a specific domain, which make them difficult to extend. Moreover, the intent and slot label sets are usually decided by human experience. Since we usually do not know the exact intents or slots of a new unlabeled data, the assigned label names may be subjective in some extent. To better assist the human labeling process, Wen et al. (2017) proposed an improved version of *Wizard-of-Oz* (Kelley, 1984) data collection methods, which incorporate crowdsourcing to collect domain specific data. Instead of human-labeling, Cohn et al. (1995) proposed a semisupervised framework *active learning*, which can minimize the need for human annotation in a certain extent. However, these approaches are still mostly or selectively human-labeled, and may be distracted by the disadvantages raised above.

Thus, in this paper, we propose an unsupervised labeling method to automatically cluster dialogue intents and corresponding slots. Since the intent of a dialogue utterance may depend on its topic or some frequent key words, utterances in the same intent may share similar context features. Hence, we cluster these utterances into different intents. Given a new dialogue dataset whose number and type of intents are uncharted, the clustering process does not need any prior information and can derive a new set of intents. We modify dynamic clustering methods to automatically decide the number of clustering classes. The clustered intents are labeled as integer indices. Before clustering, for better utilization of extracted features, we leverage an autoencoder to map all features into the same space. For the slot labeling, since the phrases of same slots such as location and time may share same type of features, we leverage a similar way to cluster slots within each intent.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 684–689
Brussels, Belgium, October 31 - November 4, 2018.

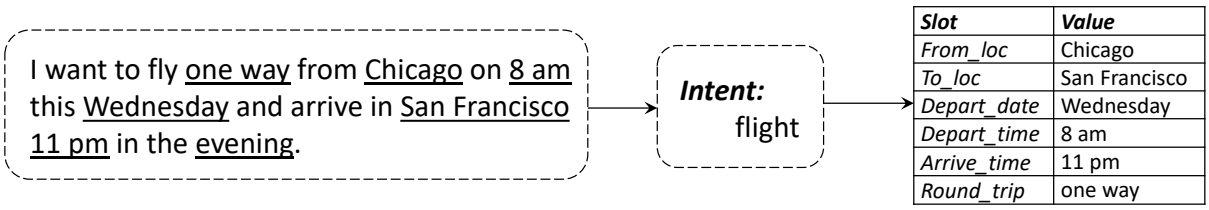

Figure 1: An example of the dialogue labeling task.

Experimental results on the ATIS dataset (Hemphill et al., 1990) show that our proposed methods can achieve 84.1% intent clustering accuracy, and provide reasonable and instructive slot labeling results. Moreover, since the whole process is unsupervised, it can be much faster and more consistent and objective than human labeling, and extended to other domains. We think the proposed methods can be a good attempt for the automatic dialogue labeling task.

2 Auto-Dialabel Framework

Formally, we treat a multi-turn dialogue session $\mathcal{D} = \{\mathcal{Q}_1, \mathcal{R}_1, \cdots, \mathcal{Q}_\mathcal{N}, \mathcal{R}_\mathcal{N}\}$ as a sequence of $\mathcal{N}$ query-response pairs between two interlocutors, in which *query* $\mathcal{Q}_*$ represents user's utterance and *response* $\mathcal{R}_*$ represents assistance's utterance. Each query utterance $\mathcal{Q}_t$ should be labeled with an intent $\mathcal{I}_t \in [0, \mathcal{K}]$, which represents the interlocutor's purpose in current utterance. $\mathcal{K}$ is the number of intent classes, and should be dynamically decided during the labeling procedure.

Since each intent has its corresponding slots, for intent $\mathcal{I}_t$, we set its slot as $\mathcal{S}_t = \{\mathcal{S}_{t,1}, \cdots, \mathcal{S}_{t,L_t}\}$, where $\mathcal{L}_t$ is number of slots in $\mathcal{I}_t$. $\mathcal{S}_t$ is also learned automatically and dynamically.

In detail, given a set of query utterances, the unsupervised dialogue auto-labeling system labels the intents and slots based on the following steps:

feature extraction and assembly, which extracts a set of context features $\mathcal{F}$ from query utterance, and leverages an autoencoder to compress each extracted feature into same size, then concatenate them as the assembled feature embedding $\mathcal{E}$.

intent clustering, which adopt dynamic hierarchical clustering to get intents based on $\mathcal{E}$.

slot clustering, which leverages the same clustering methods to get slots based on word-level features and labeled intents.

The process of intent labeling is shown in Figure 2. Slot labeling has a similar process to intent labeling. Features are a key to both intent labeling and slot labeling. We first introduce the feature extraction and assembly, which involves all the features used in our model. It is noted that we leverage all the features for intent clustering while we only use word-level features for slot labeling.

2.1 Feature Extraction and Assembly

We design a set of context features $\mathcal{F}_*$ at different levels of granularity to model the query utterance, including *word embedding*, *POS tag*, *frequent key words*, and *topic features*.

Word Embedding Given an n-words $\mathcal{Q} = \{w_1, \cdots, w_n\}$, intuitively, the feature of $\mathcal{Q}$ relied on each words within it. One frequently-used way to model a sentence by words is to use a mean pooling for all word embeddings: $\mathcal{F}_\mathcal{W} = \frac{1}{n}\sum_i^n$ *embedding of* w_i

POS Tag Since the distribution of the POS tag may effect the sentence's structure in syntactic level, we use *bag-of-POS* as the POS tag feature $\mathcal{F}_\mathcal{P}$. Given n_p types of POS tags, $\mathcal{F}_\mathcal{P} = \{p_1, \cdots, p_{n_p}\}$ is a discrete vector in which each dimension p_i represents the existence of a POS tag POS_i.

Frequent Key Words In several occasions, the intent of a sentence is decided by some *key words*. So we specially emphasize it by introducing the frequent key words feature $\mathcal{F}_\mathcal{X}$. $\mathcal{F}_\mathcal{X} = \{x_1, \cdots, x_c\}$ represents the information of key words in query utterance. To centralize the word information, we cluster all the noun words into c different classes by its word embedding, then count the occurrence frequency of each class as a discrete vector $\mathcal{F}_\mathcal{X}$.

Topic The topic-level feature denotes the topic information of query utterance. We leverage an unsupervised topic model to get the topic distribution $\mathcal{F}_\mathcal{T} \in \mathbb{R}^t$ as the topic-level feature, t is the number of topics. Since query utterance are short

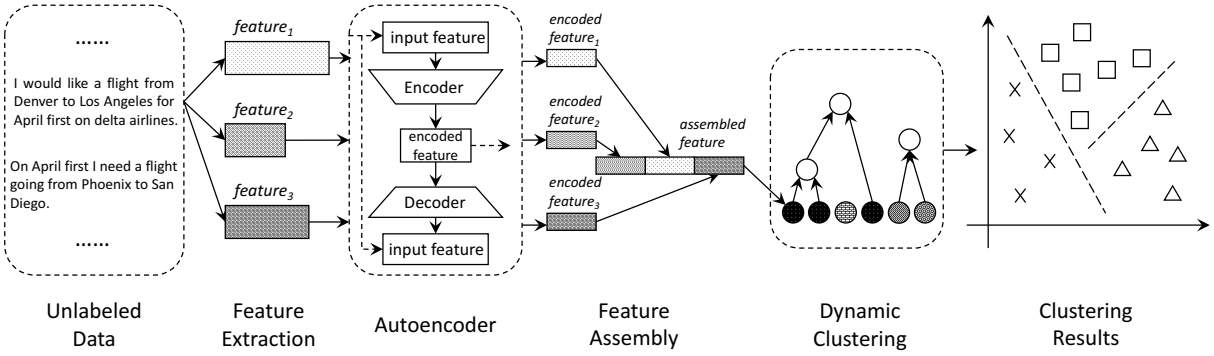

Figure 2: An illustration of our proposed auto-dialabel framework for intent labeling.

texts, while conventional topic models such as LDA and PLSA depends on document-level word co-occurrence patterns to detect topics, which may suffer from data sparsity, thus directly applying those models may not work well. In this paper, we leverage the biterm topic model (BTM) proposed by Yan et al. (2013) for better performance.

The assembled feature embedding $\mathcal{E}$ is the combination of each $\mathcal{F}_i$. Since each $\mathcal{F}_i$ has different dimensions, which may unequally affect the clustering results, we use an autoencoder to encode all $\mathcal{F}_i$ into same dimensions as $\mathcal{E}_i$, then we concatenate all the $\mathcal{E}_i$ as $\mathcal{E}$, which will be used in the clustering procedure, as shown in Figure 2.

2.2 Intent Clustering

Since given a new set of dialogue data, we do not know the number of intents it contains, thus we adapt the hierarchical clustering method to a *dynamic* version which can automatically decide the end of the clustering process by the cohesion of different classes. At the beginning of clustering, each query utterance is considered as a different class. At each steps, the cluster model chooses two classes which are the closest in distance, and cluster them into the same class. We use radial basis function (rbf) as the clustering distance. This process ends when all the distances exceed the threshold value, which is tuned on a labeled dataset, and fixed for future use.

2.3 Slot Clustering

Since slots usually correspond to intents, we do slot labeling based on both the query utterance and the labeled intents. Considering that most slots are composed of noun words, we extract all the noun words in a dialogue, and leverage the same clustering methods as in the intent clustering part to cluster them into different slot classes. Note that the slot clustering are word-level. So in *feature extraction*, it did not extract the topic-level features.

3 Experiments

3.1 Dataset and Baseline Systems

We conduct experiments on the widely used ATIS dataset (Hemphill et al., 1990). The clustering parameters are tuned on the training set of the ATIS dataset (Hemphill et al., 1990), while the experiments are conducted on its test set. During clustering, we tune the clustering distance limit since it may be more general than the number of classes, and can be transferred to other datasets. The radial basis function (rbf) is used with sklearn default settings.

Since there is no existing systems specially designed for unsupervised dialogue labeling, we choose three well-known and widely used sentence representation methods, and leverage the results vector for clustering as our baseline systems. The first one is the BTM topic model (Yan et al., 2013). We use the topic distribution for clustering. The second one is the CDSSM model proposed by Shen et al. (2014). We use the clickthrough data for pre-training and get encoded sentence vector for clustering. The third one is the sentence embedding calculated by the average of word embedding in query utterance. For each baseline, we leverage the k-means for clustering and use the gold intent number as the cluster number of these models.

3.2 Intent Labeling

We leverage glove.6B.300d (Pennington et al., 2014) as the pre-trained word embedding in all the baseline and proposed systems. The POS tag is labeled through NLTK toolkit (Bird et al., 2009).

Models	Intent Labeling Acc (%)
topic model	25.4
CDSSM vector	20.7
glove embedding	25.6
auto-dialabel	**84.1**

Table 1: Intent labeling accuracy.

The topic number of BTM is set to 20 as default. Each encoded feature dimension is set to 30, then concatenated to a 120 dimension assembled vector. We have 17 kinds of gold intents, and our system predicted 18 kinds of intents. Since the clustering results have no label information, we sort the predicted intent classes and gold intent classes by size, and manually map them. We leverage intent labeling accuracy as the evaluation metrics.

3.2.1 Overall Performance

Our auto-dialabel can reduce the tedious work of understanding the intent of each dialogue utterance and finding slot words in it. It clusters all utterances into several intent classes, and words into slot classes, which leaves human labelers only small labors to set label names for each class. Experimental results show that with auto-dialabel, we can label the whole ATIS dataset in less than 1 hour from end to end, compared to days we spend by human labeling. Table 1 shows the performance comparison of our model with other baseline systems. From Table 1, we find that our proposed auto-dialabel achieves high intent labeling accuracy (84.1%) and outperforms other baseline systems by a large margin. This may be because that the baseline systems are not specifically designed for intent scenario so that they can not handle the intent and slot clustering well, or are not capable of capturing complex intent relevant information.

3.2.2 Ablation Tests

Feature Extraction The feature extraction part extracts 4 kinds of features. We conduct ablation tests including and excluding each kind of features to see the performance. The results are shown in Table 2. From Table 2, we find the frequent key information as the most important feature. Since the intent is usually influenced by key words occurring in utterance, these kind of features can better capture key information which contributes most to the intent detection. On the contrary, the topic feature is less useful. Since it is not designed for this task, the information it represents may not directly assist the detection of intents.

Model	Acc	Model	Acc	Model	Acc
$\mathcal{F}_{\mathcal{T}}$	50.8	$-\mathcal{F}_{\mathcal{T}}$	78.9	*no_encode*	32.5
$\mathcal{F}_{\mathcal{X}}$	78.0	$-\mathcal{F}_{\mathcal{X}}$	77.5	*K-means*	28.5
$\mathcal{F}_{\mathcal{W}}$	75.9	$-\mathcal{F}_{\mathcal{W}}$	80.7	*Spectral*	40.7
$\mathcal{F}_{\mathcal{P}}$	70.2	$-\mathcal{F}_{\mathcal{P}}$	78.3	***auto-dialabel***	**84.1**

Table 2: Intent labeling ablation tests. $\mathcal{F}_*$ shows the performance of the system with only feature $\mathcal{F}_*$. $-\mathcal{F}_*$ shows the performance of the system excluded feature $\mathcal{F}_*$. *no_encode* shows the performance of the system excluded the autoencoder parts.

Intent	Slot
flight	*period_of_day* : noon, evening *month_name* : november, april *day_name* : monday, sunday *city_name* : cleveland, houston
ground_service	*period_of_day* : night, morning; *day_name* : monday; *city_name* : denver, washington

Table 3: Slot clustering cases

Feature Assembly We concatenate the originally extracted features for clustering. The results are shown in Table 2 as *no_encode*, which suggests that the encoding part is essential in feature assembly. Since each feature has a different dimension, if we alternatively concatenate them directly, the "longer" feature may get more "attention", which may distract the clustering results. Generally, encoding all the features into the same dimension is an efficient way to balance the information.

Dynamic Clustering To test the performance of our modified dynamic clustering method, we leverage two most common used clustering methods for ablation test, which are k-means and spectral clustering. Both clustering numbers are set to gold intent number. The results are shown in Table 2, and we can find both methods perform worse than our methods, which shows that our methods can handle this task well. Besides, both baseline methods need prior intent number which is unavailable in advance in most cases. Compared with these baseline clustering methods, our method can dynamically determine the intent number and is more practical.

3.3 Slot Labeling

Due to the limitation of space, we just show some slot clustering result cases in Table 3. After manually assigning names, we find that auto-dialabel can extract about 70% of the slots with accuracy, including *city_name*, *period_of_day*, *month_name*,

and *day_name*. The labeled slots above are the main slots for this scenario and could cover a large portion of airline ticket reservation demands. Generally, the slots clustered by auto-dialabel are reasonable and constructive.

4 Conclusion

In this paper, we formalize the auto-labeling task for dialogue data, and propose an unsupervised framework *auto-dialabel*. We design a set of linguistics and neural-network based features, leverage an autoencoder for feature assembly, and modify a hierarchical clustering method for dialogue intents and slots labeling. Experimental results show that our framework can achieve 84.1% intent clustering accuracy, and provide reasonable and instructive slot labeling results.

Acknowledgements

Our work is supported by National Natural Science Foundation of China under Grant No.61333018 and the National Key Research and Development Program of China under Grant No.2017YFB1002101. The corresponding author of this paper is Houfeng Wang.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

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

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1995. Active learning with statistical models. In *Advances in neural information processing systems*, pages 705–712.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.

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

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *Computer Science*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 438–449.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456. ACM.