Feature Inference Based on Label Propagation on Wikidata Graph for DST

Yukitoshi Murase, Koichiro Yoshino, Masahiro Mizukami, Satoshi Nakamura

Abstract One of the major problems in Dialog State Tracking (DST) is the large size of user intention space, and thus data preparation for statistical models is hard. In this paper, we propose a method to estimate reliable features of utterances based on creation of a knowledge graph and inference of features on the graph. For the knowledge graph, Wikidata, a large knowledge base consists of concepts and their relations on web, applied to construct a task-domain dependent graph structure. On the created graph, the label propagation algorithm is used to infer features. This inference algorithm propagates of unobserved word nodes from observed words in user utterances. In addition, dialog history words in previous turns is considered as inputs of label propagation. A large vector is created with the inference algorithm, and it is used as a feature of machine learning model. Multi-layer perceptron is adopted as machine learning model of the dialog state tracker, which predicts user intentions. Experimental results show that the proposed method obtain various and stable features, and the results achieve higher scores than the prepared baseline tracker, which does not use the inferred features from the knowledge graph.

1 Introduction

Dialog state tracking (DST) is an important task of spoken language understanding. DST is a task to trace user intentions (dialog states) of input utterances and several dialog histories [12, 5]. One major problem of DST is the large size of user intention space, which consists of every possible intention on tasks or domains. Collecting enough training data, which covers every user intention in test data, is hard due to the size of the space [10]. User intentions are mostly depended on tasks and domains, and this is one of the major reasons why preparing large scale data is hard.

Yukitoshi Murase, Koichiro Yoshino, Masahiro Mizukami, Satoshi Nakamura
Graduate School of Information, Nara Institute Science and Technology (NAIST), Takayama-cho, Ikoma, Nara, 6300192, Japane-mail: y-murase@is.naist.jp

In this paper, we extend and relax words in an user utterance into the generalized class word by the inference on knowledge graph. Knowledge graph has been widely used as resources for spoken dialog systems [9, 1], especially on Bayesian update of dialog state [4, 2, 7]. These works construct graphs by hands or unsupervised manners from Web search queries. Yi et al. [15] constructed the graph from database of web search queries and inferred the queries on the graph such as dialog state tracking. This work inspires our proposed method where the inference on knowledge graph can be used for feature, which improves the machine learning-based dialog state tracker.

Our proposed method adopted Wikidata and label propagation, which are respectively used as the graph and the inference algorithm. Wikidata is a free and open knowledge base [8], which contains numerous items and their properties. The items expresses names of concepts, and their properties expresses relationships between items. According to the characteristic of the data structure, undirected graph is created with items as nodes and properties as links.

Bag-of-words (BoW) is a basic feature and commonly used as the baseline feature of statistical dialog state tracker. However, the vector space of the word feature will be sparse because training data is not often enough for statistical learning. One solution for this problem is employing embedded expressions of words [13, 6] that is trained from large-scale data. This method compresses a vector of each word into fixed-length dimensions by using distribution of surrounding words in contexts to represent the meaning of the word implicitly. In contrast, subgraph is constructed to capture meanings of utterances with inferring neighbors in the our proposed method. Once some words are observed in an utterance, label propagation infers the features on the subgraph by using node discrimination [14, 11]. This method realize given values to some diversions of sparse word vectors.

Label propagation algorithm is one of the node discrimination methods and propagates labels from the observed node (seen words in the user utterance) to neighbors. Combinations of estimated labels can be used as a feature for the utterance. The algorithm has several procedures for feature extraction. In other words, the labels of seen nodes are labeled as known class (=1), and other nodes are labeled as unknown class (=0) at the first step. The label of each node is propagated to neighbors once the label propagation is executed on the graph. The advantages of the proposed method are that subgraph of Wikidata is easily created, and label propagation is applicable on any graph.

2 Dialog State Inference on Knowledge Graph

Our work is inspired by the previous work which tried transforming knowledge base to inference knowledge graphs (IKG) as graphical models [15]. In this work, IKG predicts dialog states by inferring confidence of each node. Markov Random Field (MRF) is used to find the most appropriate node as dialog state. The knowledge base contains data types of entities and attributes where they represent items and

relationships. The graphical model contains both entities and attributes as nodes. In other words, attribute nodes are always exist between entity nodes, and the attribute nodes can be factored nodes when a inference method executed on MRF. Their approach can utilize any inference method, and the method extracts factors from an utterance to infer some unknown entity class. A node, that have the highest confidence, is selected as the current dialog states. In contrast, we constructed undirected graph with only entities (items in Wikidata), where its edges represent relationships, and our proposed method inferred particular node with label propagation algorithm.

3 Feature Inference on Wikidata Graph

3.1 Subgraph Creation from Wikidata

Wikidata consists of items and properties for creating a graph, and items and properties respectively become nodes and edges on the graph. On the graph, label propagation algorithm extracts features for input of machine learning based tracker. However, Wikidata has numerous items, and thus label propagation costs a large calculation time during the inference process on the Wikidata graph. Due to the calculation time, a subgraph is created with words in utterances of data sets, and the words are matched with a name of items in Wikidata.

For the graph creation process, each utterance is tokenized by NLTK[1] tokenizer, and words that matched with NLTK stopwords, “!”, “?”, and etc are cleared. Uncleared words are added as initial nodes on the subgraph, and all related items of initial nodes are added as neighbor nodes on the subgraph. Finally, all nodes are given unique ids since there exists name duplications between some items.

An example of a subgraph is shown in **Fig. 1** and consists corresponding nodes of “Singapore”. The “Singapore” node is added on the subgraph with its neighboring nodes (“Asia”, “City”, “Island nation”, “Country” and “Malaysia”). Nodes on 1-hop relation are also added (“Area” and “Continent”). In addition, we assume that Malaysia is also observed in utterances, and related nodes in Wikidata are connected to the nodes of “Malaysia” (“Country” and “Asia”).

3.2 Label Propagation on Subgraph

Label propagation predicts class labels of unobserved nodes when labels of some observed nodes are given. The algorithm assumes that neighboring nodes in the graph network may have the same class label. Our proposed method defines observed class nodes and unobserved class nodes from utterances and infers class labels of unobserved nodes to extract features for machine learning models.

[1] NLTK: http://www.nltk.org/

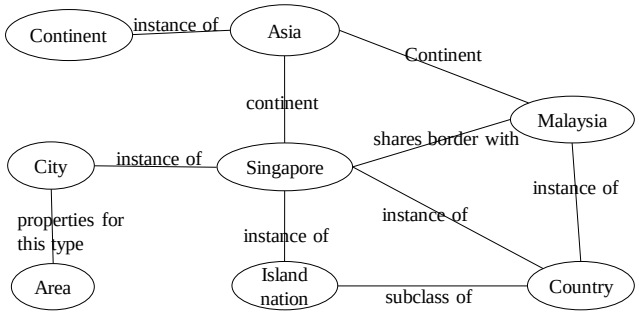

Fig. 1 An example of Wikidata graph with the appeared words "Singapore" and "Malaysia". There are titles of relations on edges, and neighbors. Some neighbors have shared relations of other nodes and an independent relation itself.

In label propagation algorithm, node links are represented as $\mathbf{W}$. $\mathbf{W}$ is an $N \times N$ matrix where N is the number of nodes in the graph. Each element in $\mathbf{W}$ represents the link existence. $\mathbf{y}$ is a vector and contains class labels for each node. In our case, $\mathbf{y}$ expresses the observation of the word in the current utterance. In other words, label 1 means that the node is observed, and 0 means that the node is not observed in the utterance. $\mathbf{f}$ is a vector of predicted class label of each node. The objective function of label propagation to be minimized is defined as,

$$J(f) = \sum_{i=1}^{n} (y_i - f_i)^2 + \lambda \sum_{i<j} w_{i,j} (f_i + f_j)^2. \tag{1}$$

The first term in Equation (1) approximates the predicted vector f to be close to the input vector y. The second term approximates the predicted values of neighboring nodes. λ is a constant value to keep balance between the first and the second terms.

Formula deformation of Equation (1) with Laplacian matrix is,

$$J(f) = \|\mathbf{y} - \mathbf{f}\|_2^2 + \lambda \mathbf{f}^{\mathbf{T}} \mathbf{L} \mathbf{f}. \tag{2}$$

$\mathbf{L} \equiv \mathbf{D} - \mathbf{W}$ is Laplacian matrix, and $\mathbf{D}$ is the summation of each row into diagonal components. This minimization problem is solved with,

$$(\mathbf{I} + \lambda \mathbf{L})\mathbf{f} = \mathbf{y}, \tag{3}$$

as defined in [14].

We implemented equation (3), where observed words in utterances are vectorized as $\mathbf{y}$, and $\mathbf{f}$ is a vector of predicted values of relaxed class nodes inferred by the Wikidata graph. $\mathbf{y}$'s elements are initially ones if words are observed in the input utterance. Then, we calculate $\mathbf{f}$ by,

$$\mathbf{f} = \mathbf{y}(\mathbf{I} + \lambda \mathbf{L})^{-\mathbf{1}}. \tag{4}$$

The previous values of **y** are also added with a discount value γ, which is a value between $0 \leq d \leq 1$ to consider the sequence of dialog. Once the discount value is factored on previous values, **y** is replaced with factored values and added labels at the current utterance. At the end, **f** is calculated by equation (4) and returned as the feature vector of the current utterance. Label propagation with γ is shown in **Algorithm1**.

Algorithm 1 Label Propagation with Discount Factor

Require: $\lambda > 0$, $0 \leq d \leq 1$, $i = index$ and $t = time$
  if Initial Utterance in Sessions **then**
    for $y_{i,t}$ in the word list **do**
      $y_{i,t} = 1$
    end for
  else
    From second utterance do:
    for $y_{i,t}$ **do**
      if $y_{i,t}$ in the word list **then**
        $y_{i,t} = 1 + \gamma y_{i,t-1}$
      else
        $y_{i,t} = \gamma y_{i,t-1}$
      end if
    end for
  end if
  $\mathbf{f} = \mathbf{y}(\mathbf{I} + \lambda \mathbf{L})^{-\mathbf{1}}$
  return f

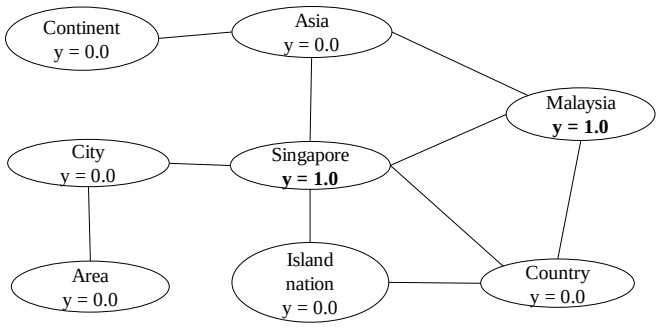

Fig. 2 An example of Wikidata graph with the appeared words "Singapore" and "Malaysia", and each node has a y value to be 1 for appeared nodes and 0 for others.

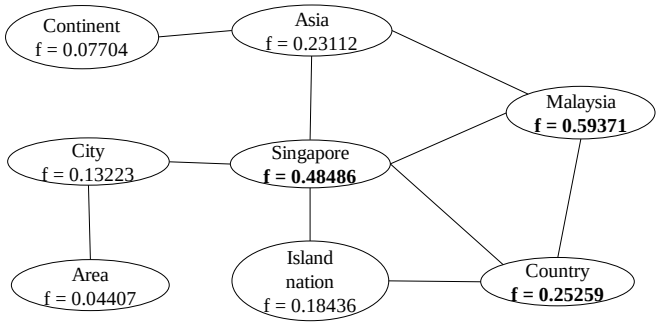

Fig. 3 An example of Wikidata graph after running label propagation on **Fig. 2**, and nodes have f values. Of course, observed nodes have higher values, and closer nodes also have higher values. For example, "Country " and "Asia" are linked to both observed nodes, and "Country" is also linked to "Singapore" 1-hop away by "Island nation". Thus, "Country" has a little higher predicted value than the value in "Asia".

4 Experimental Evaluation

4.1 Data Set and Task Description

We evaluated the proposed method on DSTC4 main task, which tracks dialog states at each utterance level. The corpus consists of conversations on touristic information for Singapore and contains 35 dialogs by 3 tour guides and 35 tourists. The corpus also contains 31,034 utterances and 273,580 words, which are manually transcribed and annotated. The corpus is respectively divided into training, development and test set, and each data set has 14, 6, and 9 dialogs. Each dialog is divided into sub-dialogs with annotations of begin/inside/others (BIO) tagging. 'B' annotation represents the beginning of the sub-dialog session, and 'I' annotation represents inside of the sub-dialog session. Otherwise, 'O' is annotated to the utterance. Each sub-dialog session is annotated with topics of five categories, and dialog states, which specify the contents of sub-dialog. The dialog states belong to one of topics for whole sub-dialog session. The number of possible dialog state is about 5,000, and each utterance has multiple states. We define that the main task's problem is solved as multi-label classification of machine learning method.

For experimental comparison, a baseline method, which is fuzzy string matching with ontology, is provided. We also examined BoW, BoW with Word2Vec (W2V), and our proposed method for the fair comparison. The baseline method matches some part of utterance and ontology's entries. Ontology is constructed as tree structure and has all possible dialog states at its leaves. The best score's method at DSTC4 are provided as different methods.

Accuracy and F-measure scores are used for evaluation metrics. The accuracy is harmonic mean of precision and recall of utterances that the tracker successfully

recognized the all of slots. The f-measure score is the ratio of slots that the tracker successfully recognized. There are two categories of scores: schedule1 calculates score at each utterance; and schedule2 calculates score at each end of sub-dialog.

4.2 Evaluation Settings

Feed forward neural network (FF-NN) model is adopted as machine-learning based classifier. Inputs of the classifier are defined three types of features: BoW; BoW with W2V; and our proposed method. BoW is a sparse vector of observed word in user utterance. W2V is summation of word vectors, which is calculated by W2V for all observed words in the user utterance. All Wikipedia articles are used to train the W2V with the default setting of gensim[2] library. Dialog histories are also considered during sub-dialog session by just adding previous feature vectors.

For FF-NN model, sigmoid function is used as the activation function of output layer. The following parameters are used to train the FF-NN: learning rate=0.000025; optimization method=Adam; dropout=0.2.

4.3 Parameters of the Proposed Method

We describe several parameters introduced by label propagation and the threshold for output of the NN model before showing the results. The input of label propagation includes dialog histories with discounted value γ. This discount value decides the degree to consider dialog histories. Note that γ=0 means the system does not consider any history, and γ=1 means the system never forget what users previously said. We assume that the smaller γ is more efficient for prediction. λ, which balances between two terms in label propagation, is needed to be decided. However, we do not have intuition for λ, so we set the balanced value between 0.5 to 8. The last parameter is threshold τ to decide the output of NN. 0.5 is generally used, and the smaller τs allow the NN to output candidates that NN has smaller belief. In other words, setting the τ smaller will cause the increasing of recall and the decreasing of precision. We simply set the stride by 0.1 between 0.1 and 0.9. **Table 1** shows the parameter candidates that are used in experiments. We tried grid search to find the best combination that achieves higher accuracy.

[2] https://radimrehurek.com/gensim/

Table 1 All three parameters lists.

λ	Discounts (γ)	Threshold (τ)
0.5	0	0.1
1	0.125	0.2
1.5	0.25	0.3
2	0.5	0.4
3	0.7	0.5
8	0.8	0.6
	0.9	0.7
	1	0.8
		0.9

Total number of combination of parameters is 432.

4.4 Experimental Results

Table. 2 and **3** show the results of using BoW as the feature, and the results are obtained by changing threshold τ, which makes decision to output candidates. All scores are still lower than those of the baseline methods as the results on the tables. This is probably because using the BoW is too sparse for NN of multi-label prediction, which has high dimensional output layer.

Table 2 Top 3 Accuracies of Bag of Words

Threshold	schedule1	schedule2
0.9	0.0061	0.0095
0.8	0.0054	0.0085
0.7	0.0037	0.0066

Table 3 Top 3 F-measures of Bag of Words

Threshold	schedule1	schedule2
0.7	0.0161	0.0159
0.6	0.0153	0.0154
0.5	0.0136	0.0132

Table 4 Top 5 Accuracies for schedule1 with Proposal Features

λ	Discount:γ	Threshold:τ	Accuracy
0.5	1.0	0.3	0.0490
0.5	1.0	0.2	0.0481
0.5	1.0	0.4	0.0456
1	1.0	0.3	0.0452
3	1.0	0.3	0.0427
baseline			0.0374

Table 5 Top 5 Accuracies for schedule2 with Proposal Features

λ	Discount:γ	Threshold:τ	Accuracy
0.5	1.0	0.3	0.0559
0.5	1.0	0.2	0.0559
0.5	1.0	0.4	0.0549
0.5	1.0	0.5	0.0521
3	1.0	0.3	0.0502
baseline			0.0488

Table 6 Top 5 F-measure for schedule1 with Proposal Features

λ	Discount:γ	Threshold:τ	F-measure
0.5	1.0	0.2	0.3444
3	1.0	0.2	0.3397
1	1.0	0.2	0.3391
8	1.0	0.2	0.3381
2	1.0	0.2	0.3371
baseline			0.2506

Table 7 Top 5 F-measure for schedule2 with Proposal Features

λ	Discount:γ	Threshold:τ	F-measure
1	1.0	0.2	0.3759
3	1.0	0.2	0.3763
8	1.0	0.2	0.3754
0.5	1.0	0.2	0.3750
2	1.0	0.2	0.3727
baseline			0.3014

Table 4 - **7** shows the results of our proposed method's feature. Specifically, **Table 4** and **5** show accuracies and **Table 6** and **7** show f-measure for each schedule with 5-best parameter conditions. Our proposed method outperformed the baseline method on each metrics according the highest results on tables.

According to the result tables, discount factor (γ)=1 achieves higher results, and thus we could conclude that all histories without discounting contributes the better result. Lower λ and thresholds (τ) call higher scores for accuracies. All top 5 results of f-measures have threshold τ=0.2, and non-stable value for λ for F-measure.

Fig. 4 and **Fig. 5** respectively show changes of f-measures on schedule 1 and 2 according to the changes of discount factor γ. λ is fixed as 0.5 where the value achieved the best results on **Table 4** - **6**. **Fig. 6** and **Fig. 7** respectively show relations between threshold τ and scores (precision, recall and F-measure) on both schedules. The highest F-measure is achieved at discount γ=1.0 and threshold τ=0.2 according to curves on **Fig. 6** and **Fig. 7**. Lower threshold allows the FF-NN classifier to output much more candidates of the output, which increase the recall and decrease the precision.

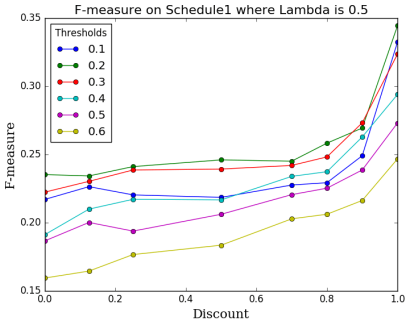

Fig. 4 Lines represent f-measure versus discount values along different thresholds thresholds on schedule1.

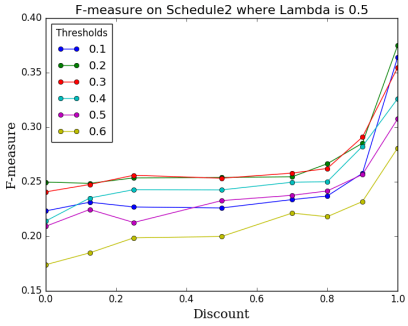

Fig. 5 Lines represent f-measure versus discount values along different thresholds thresholds on schedule2.

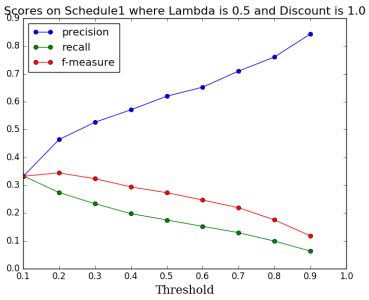

Fig. 6 Precision and recall. X-axis represents changes of thresholds, and y-axis represents f-measure on schedule1.

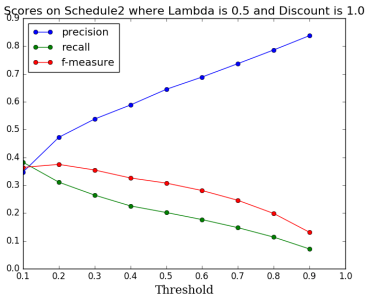

Fig. 7 Precision and recall. X-axis represents changes of thresholds, and y-axis represents f-measure on schedule2.

An example of the differences between baseline, proposed method, and gold standard labels are shown on **Table 8**. Compared to the baseline, the proposed method predicted value 'Fee' for 'INFO' where gold standard label also has. The word 'Fee' is not observed in the utterance, however, the proposed method may successfully predicted the label by features, which is probably inferred from 'free entry' in the user utterance.

Table 8 Frame Labels of Baseline, Proposed Method, and Gold Standard for an Utterance

Utterance	*Baseline*	*ProposedMethod*	*GoldStandard*
Uh National Museum, you may even get free entry because it's a- if it's a public holiday.	{}	{'INFO': ['Fee']}	{'PLACE': ['National Museum of Singapore'], 'INFO': ['Fee']}

All scores of 5 methods on schedule1 and schedule2 are showed on **Table 9** and **10** for the last comparison. The new results are BoW with W2V as input for NN model and the best results at DSTC4. Our proposed method outperforms baseline, BoW, BoW with W2V, however, the best result at DSTC4 has over 0.2 higher for f-measure [3]. One of the reasons why the best results outperforms all the other results is that the method used multiple kinds of features with elaborate hand-crafted rule-based features. The method requires hard work to imitate. The biggest difference with our proposed method is that our method used fully automated and unsupervised feature creation.

Table 9 Scores on schedule1

	Baseline	BoW	BoW w/ W2V	Proposed Method	The Best Score at DSTC4
Accuracy	0.0374	0.0036	0.0097	0.0389	0.1183
Precision	0.3589	0.0099	0.2850	0.4445	0.5780
Recall	0.1925	0.0440	0.0659	0.2749	0.4904
F-measure	0.2506	0.0163	0.1070	0.3397	0.5306

Table 10 Scores on schedule2

	Baseline	BoW	BoW w/ W2V	Proposed Method	The Best Score at DSTC4
Accuracy	0.0488	0.0066	0.0132	0.0502	0.1473
Precision	0.3750	0.0093	0.2563	0.4596	0.5898
Recall	0.2519	0.5335	0.0736	0.3186	0.5678
F-measure	0.3014	0.0159	0.1143	0.3763	0.5786

5 Conclusion

Label propagation on Wikidata graph ideally inferred features for Neural Network model, and the trained model outperformed other models, which is trained by other features. In other words, our proposed method automatically created features for large user intention space and had improved accuracy and f-measure. The experimental results showed some better combinations of parameters which had higher scores on the test set. Specially, discount and threshold values were static for f-measure's top 5 results. Inferred feature on subgraph created with nodes of words in the data set, although subgraph was not considered multi-word expressions. Moreover, the graph is created with nodes of 1-hop away from nodes, and this limit expresses less relations (properties) of nodes. We will also consider multi-word expressions and more distant nodes from the observed words in user utterances, and thus these improvement will bring a variety of properties to existing nodes for future work. Consequently, we will focus on improving the graph creation.

6 Acknowledgments

This research and development work was supported by the MIC/SCOPE #152307004.

References

1. Ali E.K., Xiaohu L., Ruhi S., Gokhan T., Dilek H.T., Larry H.: Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs. In: Proc. ICASSP, 4067-4071, (2014)
2. Antoine R., Yi M.: Efficient probabilistic tracking of user goal and dialog history for spoken dialog systems. In: Proc. INTERSPEECH, 801-804, (2011)
3. Franck D., Ji Y.L., Trung H.B., Hung H.B.: Robust Dialog State Tracking for Large Ontologies. Dialogues with Social Robots, Springer Singapore, 475-485, (2017)
4. Fabio C., Puay L. Lee.: Searching the web by constrained spreading activation. In: Proc. Information Processing and Management 36.4, 585-605, (2000)
5. Jason W., Antoine R., Deepak R., Alan B.: The dialog state tracking challenge. In: Proc. SIGDIAL, 404-413, (2013)
6. Jeffrey P., Richard S., Christopher D.M.: Distributed representations of words and phrases and theircompositionality. In: Proc. EMNLP, 1532-1543, (2014)
7. Judea P.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann (2014).
8. Kurt B., Colin E., Praveen P., Tim S., Jamie T.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. ACM SIGMOD, 1247-1250, (2008)
9. Lu W., Larry H., and Dilek H.T.: Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems. In: Proc. ICASSP, 4082-4086, (2014)
10. Matthew H., Blaise T., and Jason W.: Dialog state tracking challenge 2 and 3. (2014)
11. Oliver Chapelle B.S., and Alexander Z.: Semi-supervised Learning MIT Press, (2006)
12. Seokhwan K., Luis F.D., Rafael E.B., Jason W., Matthew H., Koichiro Y.: The fifth dialog state tracking challenge. In: Proc. IEEE Workshop on SLT (2016)
13. Tomas M., Ilya S., Kai C., Greg S.C., Jeff D.: Distributed representations of words and phrases and theircompositionality. In: Proc. NIPS, 3111-3119, (2013)
14. Tsuyoshi K., Hisahi K., Masashi S.: Robust label propagation on multiple networks. In: Proc. IEEE Transactions on Neural Networks, 35-44, (2009)
15. Yi M., Paul A.C., Ruhi S., and Eric F.: Knowledge graph inference for spoken dialog systems. In: Proc. ICASSP, 5346-5305, (2015)