International Journal of Computational Intelligence Systems
Vol. 13(1), 2020, pp. 275–290
DOI: https://doi.org/10.2991/ijcis.d.200310.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Information Retrieval Based on Knowledge-Enhanced Word Embedding Through Dialog: A Case Study

Jin Ren[1,*], Hengsheng Wang[1,2], Tong Liu[1]

[1] *College of Mechanical & Electrical Engineering, Central South University, Changsha 410083, China*
[2] *State Key Laboratory for High Performance Complex Manufacturing, Central South University, Changsha 410083, China*

ARTICLE INFO

Article History
Received 15 Apr 2019
Accepted 03 Mar 2020

Keywords
Information retrieval
Domain knowledge
Enhanced word embedding
Semantic understanding

ABSTRACT

The aim of this paper is to provide a systematic route of information retrieval from a knowledge-based database (or domain knowledge) through a dialog system of natural language interaction. The application is about a comprehensive building at a university, with classrooms, laboratory rooms, meeting rooms, research rooms and offices, and is to present related information the user asks for. First, the domain knowledge is expressed with predicate expressions based on the ontology structure; then the vocabulary is presented distributedly with word embedding enhanced with the domain knowledge; queries from the user are then converted into the intent (general) and slot elements (specific) with the help of trained recurrent neural network (RNN). The system works smoothly. The key point is integrating the two methods of knowledge-based and data-driven natural language processing into one system, and the domain knowledge is in the central part which is incorporated into the word embedding to make it specifically fit the natural language in this application.

1. INTRODUCTION

Questions often emerge when one wants to find something in a comprehensive building at a university, such as: *how can I get to the office of Professor Zhang, which room is the talk about intelligent robots to be held this evening, where is the laboratory of dynamic systems*, and so on. Of course, other similar situations may occur in tourist sites, shopping centers, public libraries, just to mention a few. It is possible to make an automatic information service available in these cases, and a dialog system with natural language interaction is getting near for this with the development of natural language processing (NLP) techniques. This paper is to provide a systematic way to this end.

To understand the question the user puts into the dialog system, we get the general intent of the question and fill the concrete details of the question in slots, which forms the understanding part of the system. To respond the question, the intent and filled-slots of the question are managed into queries in the RDF-triple (Resource Description Framework) manner which consults the knowledge database to get the answer.

Throughout the history of NLP research, the knowledge-guided and data-driven methods take turns playing the central role and there has been the debate on which is more important. Knowledge-guided method is based on symbolic processing, and data-driven method is based on statistical processing. The knowledge-guided method utilizes hand-crafted rules based on the language features of syntax and semantics; however, it doesn't work well on numerous authentic corpora on account of the complexity of linguistic phenomena due to the long natural evolvement of human languages. Although the data-driven method has made impressive progress to achieve the state-of-art performance in many NLP tasks, it is still insufficient for practical uses, and the theory inside of it cannot guarantee the completeness of its usage. In order to break through these bottlenecks, efforts should be made to combine these two ways, also called relational and distributional way, although they are uneasy collaborators [1]. In other words, it is expected to explore more deep learning models whose intern memory (bottom-up knowledge implicitly learned from the data) is enriched with an external memory (top-down knowledge explicitly inherited from a knowledge base) [2].

The main contribution of the paper is its attempt to integrate the two methods of knowledge-based and data-driven NLP into one system, where word embeddings are enhanced with the domain knowledge to specifically fit the natural language in this application. Specifically, an ontology is created in this paper for the specific application of information retrieval, as a basis not only for reasoning in RDF-triple manner, but also for forming the enhanced constraints for the training of word embedding which is then used for the retrieval of user's intents (intent detection) and sentence elements (slot-filling) from the user's questions in natural language. In this way, the knowledge is the root, and the statistic algorithms (skip-gram and recurrent neural network [RNN]) are the branches and leaves, and the combination of these two constitutes the information retrieval system in this paper.

[*] *Corresponding author. Email: renjin@csu.edu.cn*

The rest of this paper is organized as follows. After discussing related works in Section 2, the whole architecture of the information retrieval system is depicted in Section 3. As the basis of this work, the domain knowledge based on ontology is elaborated, from which multi-granularity knowledge constraints are constructed for the training of word embedding in Section 4. With the skip-gram model as baseline, two approaches of improving word embeddings are introduced in Section 5 which integrates multi-granularity semantic knowledge as the additional constraints in the process of training. Section 6 evaluates the resultant word embeddings in both qualitative and quantitative way, which shows the better results than the word embeddings from traditional training by examples. Finally, Section 7 concludes the work.

2. RELATED WORKS

2.1. Enhanced Word Embedding Model

"Word vector" and "word embedding" have the same meaning which is a model of digital representation of words in a natural language, and originates from one-hot expression by reducing the dimension of vectors tremendously and mapping words to a continuous vector space of real numbers. As a distributed representation, word embedding (through training) can extract both the semantic and syntactic information of words or phrases from a large unlabeled corpus [3], whose basic assumption is that words with similar context should have close semantics [4].

Although the co-occurrence information contained in word context of corpus has a wide coverage, its syntactic and semantic relationship involved is implicit and noisy, which makes the trained word embedding not good enough for some NLP tasks, especially the semantic part. To address this limitation, in the NLP community there is an increasing trend to make use of some mature structured domain knowledge such as WordNet [5] and Freebase [6], which conveys precise, explicit and compact semantics, such as synonym or hypernym relationship, which can bring new life to the original word embeddings. By incorporating the domain ontology as superior structured semantic knowledge into the traditional context constraints, it should make the representation coherent with domain knowledge, and hence produce a better word representation.

In general, existing knowledge-enhanced methods vary in three major aspects: (i) incorporating knowledge during the training procedure of word embeddings; (ii) employing knowledge to fine-tune the pre-trained word embeddings; (iii) utilizing knowledge graph (KG) embeddings directly trained from a structured KG as the knowledge information part in the joint model.

In terms of aspect (i), various types of external knowledge, such as morphological [7], syntactic [7,8], and semantic knowledge [5,7,9], are introduced into the training procedure of word embeddings. Hu *et al.* [8] propose a novel model named Continuous Dissociation between Nouns and Verbs Model (CDNV), which introduces certain syntactic knowledge, resembling part of speech (POS) information, to produce more reasonable word representations by constructing the Huffman tree on three groups of words (nouns, verbs, others) in the output layer. Liu *et al.* [9] incorporate hand-built domain-specific semantic relations (such as hyponymy and synonymy) into the training procedure, where the semantic constraint is considered more accurate than the context constraint, therefore more priority is put in the weight of the semantic part. Liu *et al.* [5] represent semantic knowledge as a set of ordinal ranking inequalities and formulate the training of semantic word embeddings as a constrained optimization problem, where the data-derived objective function is optimized subject to all ordinal knowledge inequality constraints extracted from available knowledge resources such as Thesaurus and WordNet. Bian *et al.* [7] explore the power of morphological knowledge (root, affix, syllable), syntactic and semantic knowledge by defining new basis for word representation, providing additional input information, and serving as auxiliary supervision.

In terms of aspect (ii), efforts are devoted to fine-tune the pre-trained word embeddings. Faruqui *et al.* [10] propose a method for retrofitting vector space representations by encouraging linked words (from semantic lexicons) to have similar vector representations. In another paper [11] they propose methods that transform word vectors into sparse and optionally binary vectors. In this way, the resulting overcomplete representations are sparse and categorial, which are more similar to the interpretable features typically used in NLP.

In terms of aspect (iii), as the knowledge information part in the joint model, KG embeddings are directly learned from structured knowledge bases. Bordes *et al.* [12] make their efforts to embed symbolic representations including entities and relations of a KG into continuous vector spaces, so as to simplify the manipulation while preserving the inherent structure of the KG [13]. Wang *et al.* [14] attempt to jointly embed entities and texts into the same continuous vector space, which is expected to preserve the relations between entities in the KG and the concurrences of words in the text corpus. In contrast to [14], Toutanova *et al.* [15] additionally model the textual co-occurrences of entities pairs and the expressed textual relations, which allows for deeper interaction between the sources of information. Zhang *et al.* [16] utilize both large-scale corpora and KGs to train an enhanced language representation model called Enhanced Language Representation with Informative Entities (ERNIE), which takes full advantage of lexical, syntactic, and knowledge information simultaneously, however, due to its large total parameters (about 114M), the model is not appropriate for the applications in specific domain with only small-scale corpora available, as is the case in this paper.

Along the line of the above aspect (i), this paper explores a way of combining explicit knowledge into word embeddings. As a main aspect of this paper, we manually incorporate the category (the ontology of our specific dialog system) as extra labels to the training objective (or multi-objective training) besides the context labels of traditional skip-gram model; moreover, we set a group of semantic inequalities of category items from the ontology according to the intuitive degree of closeness among these items, which is used in the training process to urge the final trained vectors showing this relationship. As for the use in the application of dialog system for information retrieval of a comprehensive building at Central South University, the word vectors trained this way behave very well in the experiments given below.

2.2. Natural Language Understanding Combining Knowledge-Based and Data-Driven Methods

As a fundamental component of dialog system for information retrieval, natural language understanding (NLU) has made considerable advances in the past several decades since Turing Test was proposed in 1950s. It is crucial and necessary to understand the intent conveyed in the user's question clearly and effectively before considerate responses are made. In this field there are two typical ways, namely knowledge-guided way and statistical way respectively.

Up until the 1980s, NLP was dominated by the knowledge-guided (or symbolic) paradigm in which grammars were built manually and language meanings were represented with logic-based formalisms, which proved successful for very limited domains using a finite set of commands and deeper semantics [17]. In traditional semantic parsing, First-Order Predicate Calculus (FOPC) has been used widely to represent literal content because of its advantage of inference and expressiveness.

In the 1990s, a paradigm shift occurred where probabilistic and statistical methods start playing a more important role in NLP. Besides the advances in speech recognition and NLP algorithms, the availability of large corpora of spoken language leads to an increasing use of machine learning techniques so that the previously manually crafted rules of the knowledge-based paradigm could be learned automatically from labeled training dataset in a supervised way. As a whole, statistical approaches gradually take over the research literature, but there are still many supporters of FOPC approaches, especially in industrial community where designers make their best efforts to have greater control over the output of their systems [18].

As a representative of statistical methods, neural networks recently tend to produce expected answers without providing reasonable explanation due to lack of interpretability, which become a bottleneck for further advances. Under the circumstance, explainable artificial intelligence (EAI) [19] is proposed to integrate symbolic knowledge into the data-driven statistical leaning models deeply, which will help models reduce the dependence on training samples and has great potential as the next key point to make a difference in breaking the bottleneck.

As a summary, in recent years there has been an increasing trend to combine knowledge-guided and data-driven way in NLP community, where some typical research works seem remarkable.

To answer questions in any domain, the paper [6] exploits the semantic embedding space in which the embeddings jointly encode the semantics of words and logical properties. In this way, the semantic associations between existing features can be utilized based on their embeddings instead of a manually produced lexicon and rules, which are supposed to contribute to the mapping of questions posed in a natural language onto logical representations of the correct answers guided by the knowledge base. In the paper [20], a mobile spoken dialog system for room booking is proposed with a new NLU module combining an ontology and a dependency graph to do semantic analysis, where the domain ontology is handcrafted and Stanford parser [21], a statistical parser, is used to obtain the dependency. The paper [22] deeply integrates the known grammar structure of Structural Query Language (SQL) into a new encoder–decoder neural network framework for translation from natural language to SQL, which focus on understanding semantics of the operations in natural language and save the efforts on SQL grammar learning. To map natural language to its symbolic representations of meanings, by learning low-dimensional embeddings that simulate the behavior of first-order logic, the paper [23] demonstrates that reasoning with embeddings could support the full power of symbolic representations such as first-order logic, which could capture the richness of natural language and support complex reasoning.

3. SYSTEM ARCHITECTURE

Compared with the above research works, in terms of combining knowledge-guided way and data-driven method for NLU, our work has different approaches: multi-granularity knowledge from ontology is integrated into word embeddings, which helps detect the intent of question and fill the relevant slots of structured language elements by using joint RNN model taking advantages of these knowledge-armed word embeddings. Semantic interpretations in the form of compound RDF triples for the extracted intents and language slots are then created, and the responses to the questions posed by the user are inferred with the help of the compounded RDF triples which provide more considerate and relevant answers in our dialog system.

From the design perspective, knowledge plays a central role throughout the flow of information during the semantic understanding and grounding, as shown in Figure 1. The field-specific ontology (the base part in Figure 1) is abstracted and organized manually as the domain knowledge. In the word embeddings block, three factors influence the final trained word vectors which cover all the words in the ontology: the conventional co-occurrence constraints in word context in corpora and two multi-granularity knowledge-enhanced constraints in form of general knowledge labels and precise knowledge inequalities. In the top row (Figure 1), information blocks are connected with corresponding actions: "lookup" transfers the original user question into digitalized one by looking up the trained word embeddings for words in the question; "load joint RNN model" converts the digitalized queries into general intent and filled-slots, which were labeled elaborately in accordance with the domain knowledge, and the result intent is corresponding to a relational predicate in ontology; "auto-generate" transforms the returned-intent and filled-slots into Prolog queries in compound RDF-triple manner with the help of a designed auto-generation algorithm; "consult and reason" helps make responses from Prolog queries by consulting the knowledge database and reasoning in more detailed level including the rich semantic relationships between entities as well as classes. On the whole, domain knowledge takes part in almost every system component during semantic understanding and grounding for user's question in our dialog system.

In the view of system implementation, Python and Prolog modules both make contributions to our system. Prolog module is to consult the database of domain knowledge and to provide services for semantic query and reasoning. Python module is to read and tokenize the sentence, to look up the trained word embeddings to form its consistent digitalized queries, to predict the user's intent and to fill constructed sentence slots by loading the trained RNN

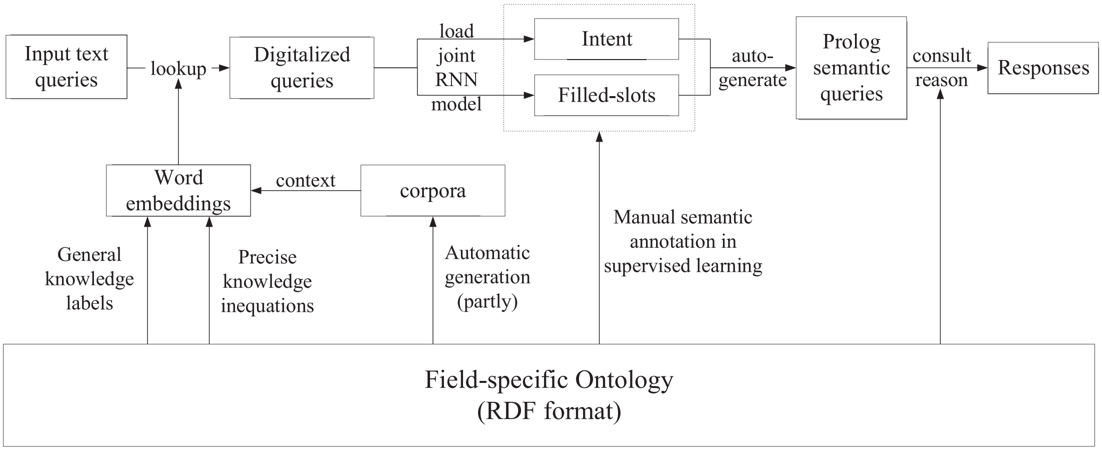


Figure 1 The architecture of knowledge-based dialog system.

model, to complete the dependency parsing, to auto-generate the compound semantic query and finally to send it to Prolog reasoner to get the response through the open software toolkit *PySwip*, the interface between Python and Prolog, and a bridge for querying SWI-Prolog in Python program. The training procedures of word embeddings and the joint RNN model are both implemented in the framework of TensorFlow with Python platform. iFLYTEK is employed to recognize Chinese speech, and LTP-Cloud platform is utilized to complete Chinese word segmentation and dependency parsing, where POS tagging is accomplished with the help of user dictionary extracted from domain knowledge.

4. DOMAIN KNOWLEDGE REPRESENTATION AND PREPROCESSING

4.1. Construction of the Domain Knowledge Ontology

An ontology is structured knowledge about the categories and properties of objects, and relations between them in a specific domain [24], which can be shared, reused, and merged across many domains. As the foundation of training word embeddings and the later reasoning, knowledge ontology can explicitly represent hierarchical structured classes, entities and the semantic relations between them. Ontology can be stored in the RDF manner, which can be parsed and consulted into database of reasoner in SWI-Prolog as facts for query and reasoning.

To provide a good knowledge retrieval service, it is of great priority to construct an ontology within a specific domain. At university campuses, it is normally needed to get some information about teachers and students, routes, places, on-going events, and so on. Questions about these needs and responses for those questions are the focus in this paper. We restricted our domain to a building complex at Central South University, which houses the College of Mechanical and Electrical Engineering (CMEE, detailed in [25]) and the information to be retrieved is all about the teachers, students, rooms, navigation routes, events, and so on, related to CMEE. The system can be expanded to the domain of the whole university and the scenario is transplantable to other similar situations.

For this specific domain, the overall hierarchy of the ontology is shown in Figure 2. Directly under the root class "Thing" are the POS level (sub)classes, consistent with the language grammar, including classes of noun, verb, preposition, pronoun, adjective, adverb, and so on. The "noun" class is most important for our application, which is divided into (sub)classes of "location," "person," "activity," "name," "gender," "major," "title," "research interest," "publication," "job," "institute," and so on, in which the hierarchical details of "location," "person," and "activity" are shown in Figure 3. Figure 2 also shows some relations between classes (besides hierarchical relation), for example, a "Person" may be "located_in" a "Location"; a "Person" may be "engaged_in" an "Activity"; an "Activity" may "occur_in" a "Location." At the bottom level of the classes hangs individual instances of each class which are consistent with the real data from CMEE.

As for the information retrieval of CMEE, "Location," "Person," and "Activity" are most frequently mentioned, such as questions: "*Where can I go to find Professor Wang's class?*" "*How can I get to Mr. Zhang's office?*" "*Which room is the job fair organized?*" where the knowledge about locations, persons, or activities is mainly concerned. Ontologies about indoor spaces, people, activities can be found in literatures [26–29], and Protégé ontology library[1] can also be used as a good reference. We chose to design the ontology ourselves because of the specificity of the application.

The "location" ontology of the building CMEE is shown in Figure 3(a), of which more details were provided in our previous work [25]. The "person" ontology for CMEE is shown in Figure 3(b), which basically reflect the roles of a person in activities and social relations between persons. The "activity" ontology for CMEE is shown in Figure 3(c).

Our knowledge base built upon the ontology above is expressed in the form of Web Ontology Language (OWL) with Protégé ontology editor. We collected real data from our college about students, teachers, building structures, offices, labs, lectures, and so on, as possible as we could, and added to the knowledge base. Figures 2 and 3 only show the hierarchical structure part of the ontology, while the rich relationships between classes and entities are not shown except for three relationship arrows in Figure 2 which

[1] https://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

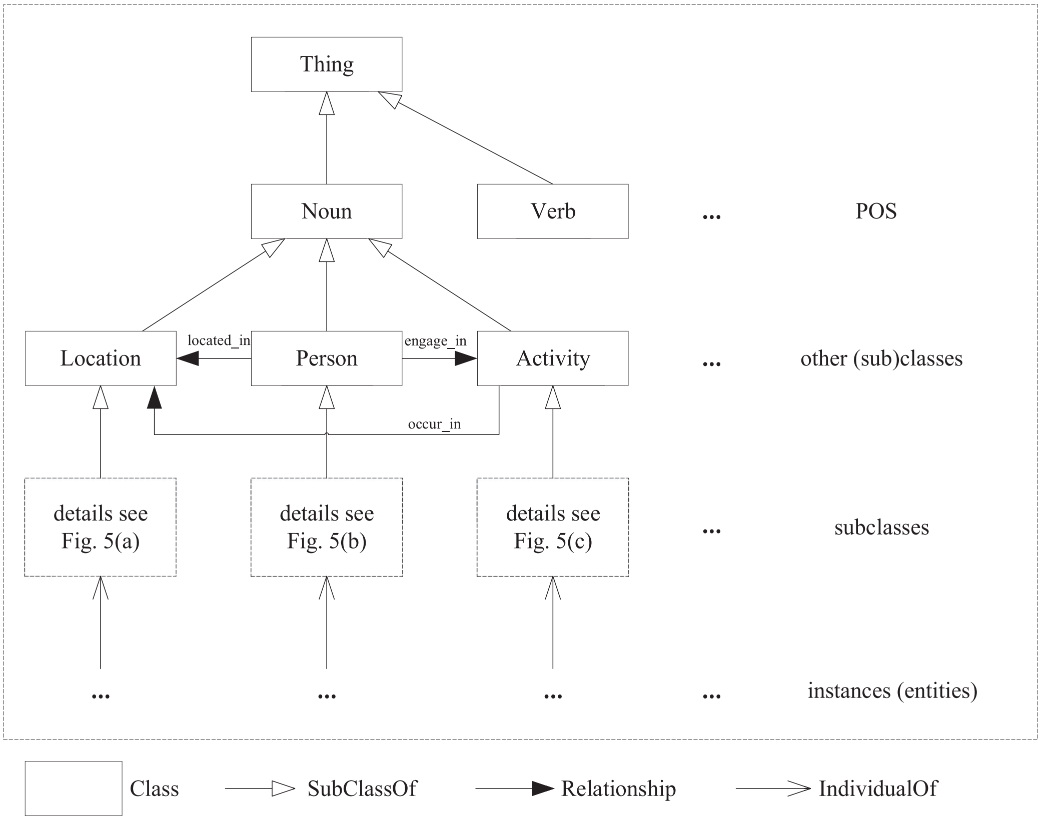


Figure 2 Hierarchy view of ontology.

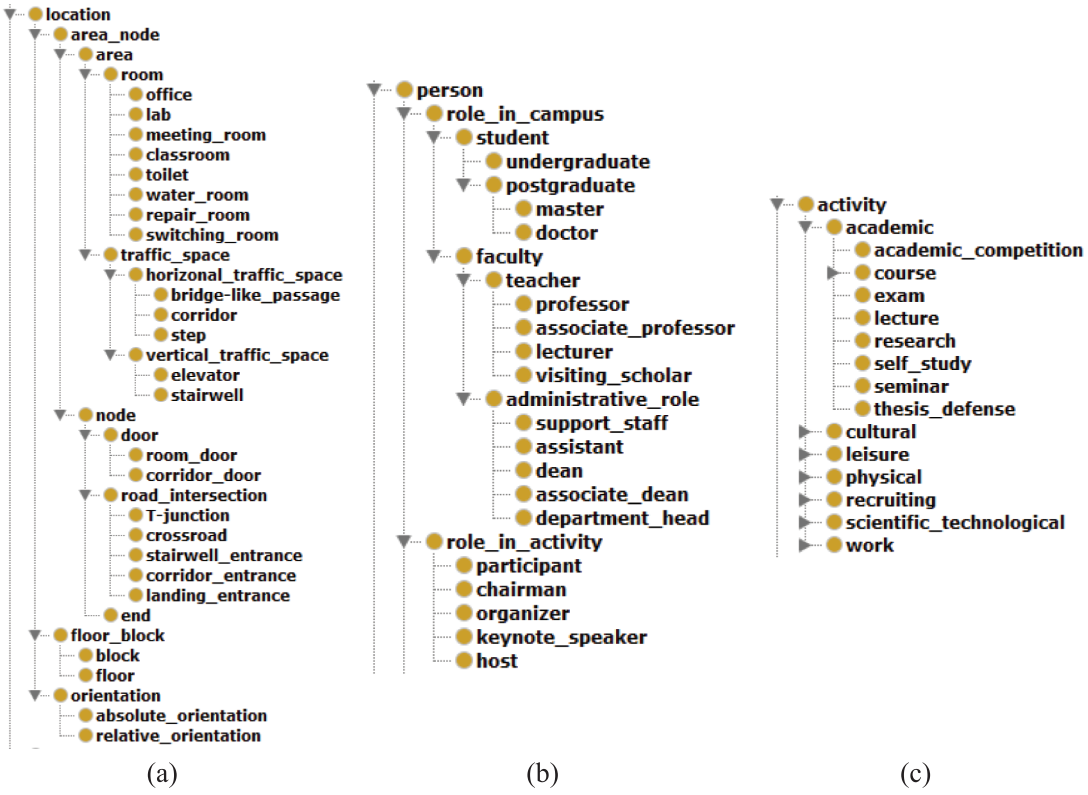


Figure 3 The ontology for locations, persons, and activities for CMEE.

show three relations between classes via three predicates (namely, occur_in, located_in, engaged_in). Sample lines of the knowledge base are look like these:

1. *rdf(indoorHCI:office, rdf:type, owl: 'Class').*
2. *rdf(indoorHCI:office, rdfs:subClassOf, indoorHCI:room).*
3. *rdf(indoorHCI: 'A520', rdf:type, owl: 'NamedIndividual').*
4. *rdf(indoorHCI: 'A520', rdf:type, indoorHCI:office).*
5. *rdf(indoorHCI: '5F', rdf:type, indoorHCI:floor).*
6. *rdf(indoorHCI: 'A520', indoorHCI:hasFloor, indoorHCI: '5F').*
7. *rdf(indoorHCI:hasFloor, rdf:type, owl:ObjectProperty).*
8. *rdf(indoorHCI:hasFloor, rdfs:domain, indoorHCI:area_node).*
9. *rdf(indoorHCI:hasFloor, rdfs:range, indoorHCI:floor).*
10. *rdf(indoorHCI: lecture_1016, rdf:type, owl: 'NamedIndividual').*
11. *rdf(indoorHCI:lecture_1016, rdf:type, indoorHCI:lecture).*
12. *rdf(indoorHCI:lecture_1016, indoorHCI:occur_in, indoorHCI: 'B214').*
13. *rdf(indoorHCI:lecture_1016, indoorHCI:hasTopic, indoorHCI: intelligent_robot).*
14. *rdf(indoorHCI:'Zhang', rdf:type, indoorHCI:professor).*
15. *rdf(indoorHCI:'Zhang', indoorHCI:is_a_keynote_speaker_of, indoorHCI:lecture_1016).*
16. *rdf(indoorHCI:is_a_keynote_speaker_of, rdfs:subPropertyOf, indoorHCI:engaged_in).*
17. *rdf(indoorHCI: 'Liu', indoorHCI:engaged_in, indoorHCI: lecture_1016).*
18. *rdf(indoorHCI: 'Liu', indoorHCI:is_a_student_of, indoorHCI: 'Zhang').*

Line 1–4 declare: (1) "office" is a class; (2) "office" is a subclass of class "room"; (3) 'A520' is an instance of some class; (4) 'A520' is an instance of class "office." And other lines express corresponding knowledge about CMEE in the similar fashion of triples.

For a practical message: "*The meeting room B214 is having an academic lecture about intelligent robot hosted by Professor Zhang, whose students Liu is attending the lecture,*" the knowledge triples are like *(lecture, occur_in, B214), (lecture, has_topic, intelligent_robot), (Zhang, is_a_keynote_speaker_of, lecture), (Zhang, is_a, professor), (Liu, engaged_in, lecture), (Liu, is_a_student_of, Zhang)*, corresponding to above lines of 10–18.

For a user's query like this: "*Where can I go to attend Professor Zhang's lecture about intelligent robots?*" if we could extract: (1) the intent of the question (asking "where," corresponding to the predicate "occur_in" in the knowledge base), (2) the corresponding entities (filled into corresponding slots) about the question, we would consult the knowledge base and reason through chains of predicates to get the answer of "B214."

This knowledge base of triples implicates semantics of words and relations among them. If only we could make constraints according to the knowledge base properly in the training of word embedding, the resulting word vectors would reflect the semantic relations someway like a vector space to its basis. One effort of this paper is to this end to construct semantic constraints from the knowledge base to train more expressible word embeddings, which is then used to digitalize user's query and to extract the intent of the query and to fill functional slots with corresponding entities. In this way, the query is understood, and the reasoning process is then applied for responding to the query which needs to generate chains of predicates according to triples in the knowledge base to get the final answer.

4.2. Construction of Multi-Granularity Semantic Constraints

All words are labeled in our system for the training of word embedding. The labels reflect the semantic meaning of words in the knowledge base with coarse granularity. We label all words in the POS level of the ontology (Figure 2) except for the "noun" type. For noun words, we label them with the subclass names immediately under class "noun." Table 1 gives some examples of word label which have a "verb" label and three subclasses of "noun" ("person," "location," and "activity"). By "coarse" we mean the label is not concrete down to the bottom level class of words. For example, the label of "A513" is "location" (a subclass immediately under class "noun") rather than "office" (bottom level class of the word "A513").

As a complement to "coarse" level of semantics, inequalities of meaning similarities among words according to the hierarchical structure of the ontology are added to the constraints for the training of word embedding, which shows somewhat "precise" relations among words in the sense of closeness in the hierarchy of ontology. As in [5] the ordinal similarity inequalities are based on intuitive degree of closeness among items of classes or instances of classes with the basic rules (Semantic Category Rule and Semantic Hierarchy Rule), for the handcrafted domain ontology in this paper, we use Prolog to help acquire the ordinal similarity inequalities.

As a logic programming language, Prolog [30] has its roots in the first-order predicate logic, where logic is represented as facts and rules. With the help of mechanism of unification and backtracking to satisfy goals and recursive tricks to realize loops, Prolog is suitable for symbolic reasoning to match answers satisfying the constraints on the base of facts and rules of knowledge. The similarity inequalities are formed automatically from the knowledge base using SWI-Prolog programing.

Similarity inequalities basically show the relationship of semantic closeness between three items (or words). For example, $sim(A, C) - sim(A, B) < 0$ indicates A is closer to B than to C in the vector space,

Table 1 Example of extracted general category labels.

Word	Label
Teacher	Person
Student	Person
Postgraduate	Person
Go	Verb
Room	Location
Office	Location
A513	Location
Lecture	Activity

or the similarity of meanings of A and C ($sim(A, C)$) is less than that of A and B ($sim(A, B)$), and this inequality is expressed with a rule in predicate logic: *semantic_hierarchy_rule(A,B,C)* or *semantic_category_rule(A,B,C)* (the definition of rule is given below). The inequalities will be used later in Section 5.3 as additional constraints for the training of the word embeddings which further elaborate the distribution of the trained word vectors.

Not all three items (or words) form an inequality, and actually we do not compare the similarity of A, B, and C in the cases of Figure 4(c) and (d), the relations between which are not obvious or instinctively comparable in the sense of the closeness of semantic similarity.

4.2.1. Semantic hierarchy rule

The similarity inequality for the case of Figure 4(a) can be formed by checking the satisfaction of following rule (called Semantic Hierarchy Rule) through the items (or words) in the knowledge base:

```
semantic_hierarchy_rule(A,B,C):-    % Semantic Hierarchy Rule
   rdfs_subclass_of(A,B), % A is a subclass of B
   rdfs_subclass_of(B,C).
semantic_hierarchy_rule(A,B,C):-
   rdfs_individual_of(A,B), % A is an individual of B
   rdfs_subclass_of(B,C).
```

which simply means "if A is a subclass (or entity) of B, and B is a subclass of C, then A and B is closer than A and C." And this is what the predicate *semantic_hierarchy_rule*/3 is used for.

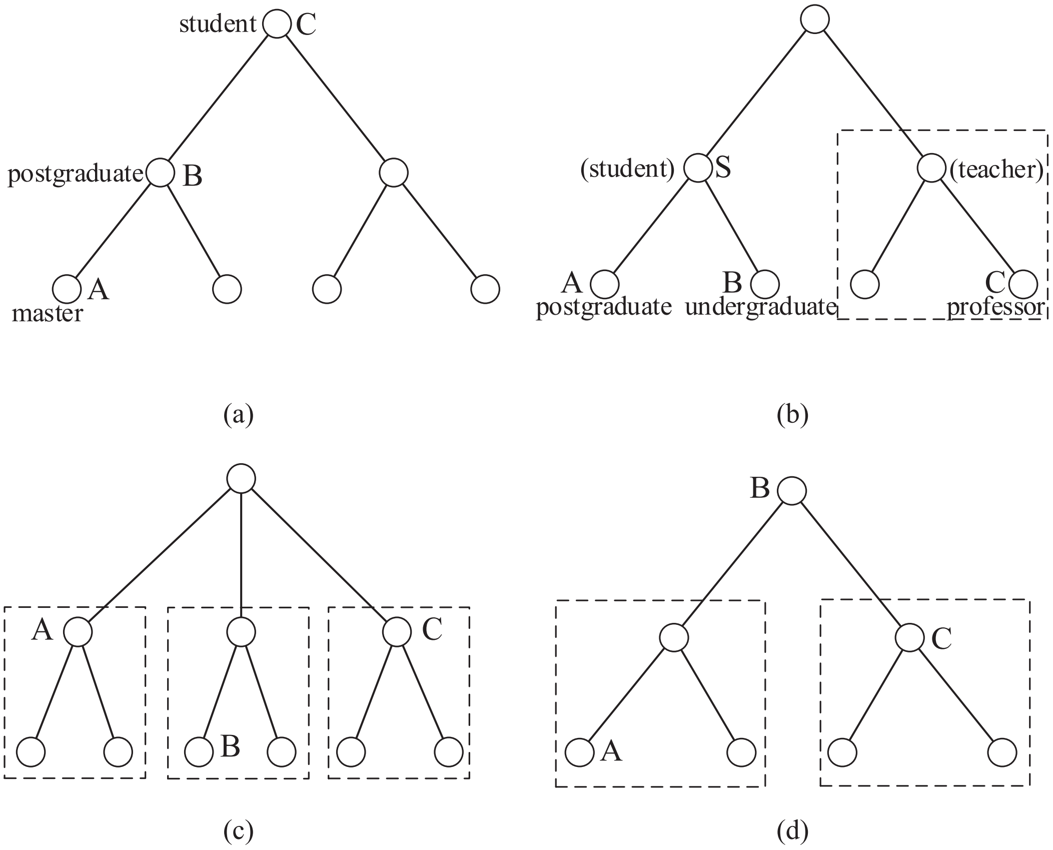
For example, *semantic_hierarchy_rule (master, postgraduate, student)* is true according to Figure 3(b) because *"master"* is closer to *"postgraduate"* than to *"student"* in the vector space, or the similarity of *"master"* and *"postgraduate"* $\left(sim(master, postgraduate)\right)$ is more than that of *"master"* and *"student"* ($sim(master, student)$), as the case in Figure 4(a).

4.2.2. Semantic category rule

The predicate *semantic_category_rule*/3 is used for the cases of Figure 4(b), which means "if both A and B belong to a class (S in Figure 4(b)), but C does not belong to that class, then A and B is closer than A and C." The definition of the predicate is as follows:

```
semantic_category_rule(A,B,C):-    % Semantic Category Rule
   (rdfs_subclass_of(A,S); rdfs_individual_of(A,S)), % A is a subclass or individual of B
   (rdfs_subclass_of(B,S); rdfs_individual_of(B,S)),
    \+(rdfs_subclass_of(C,S)),    % C is not a subclass of S
    \+(rdfs_individual_of(C,S)).    % C is not an individual of S
```

For example, *semantic_category_rule(postgraduate, undergraduate, professor)* is true according to Figure 3(b) because *"postgraduate"* and *"undergraduate"* are both under the class *"student,"* but *"professor"* is not, as the case in Figure 4(b). In other words, the similarity of *"postgraduate"* and *"undergraduate"* $\left(sim(postgraduate, undergraduate)\right)$ is more than that of *"postgraduate"* and *"professor"* $\left(sim(postgraduate, professor)\right)$.



(a) (b) (c) (d)

Figure 4 Typical cases of relative relation among arbitrary three items (or words) (the word labels in (a) and (b) are the instances of item from Figure 3(b)).

5. THE TRAINING OF WORD EMBEDDINGS WITH KNOWLEDGE-ENHANCED CONSTRAINTS

As is the basic idea of training model of word embeddings, constrained by the semantic relationship such as co-occurrence information between words and its contexts in corpora, the parameters of neural network model get optimized by back-propagating steps. Some parameters from the trained model constitute the digital expressions of words in the form of vectors, which are vector representations of these words.

Impressively, literatures using this simple model of word vector brought the world a surprisingly well-behaved vector space, like relatively concentrated distribution of words with close semantics, explicitly encoding some linguistic regularities and patterns with potential of linear translations [31], such as “King - Man” results in a vector very close to “Queen - Woman” [32]. Word vector expression shows a great potential for vector-oriented computing NLP.

Mathematically, basis is the key to vector space. One-hot expression simply makes every word an axis of the basis, and word embedding dramatically deduces the dimension of the space built on one-hot expression. POS could be considered as a potential and natural set of bases for the word embedding vector space, but this seems not so critically solvable to the problems related with NLP due to its complexity as to linear algebra. However, the idea of basis gives us inspiration for the training of word embeddings. The coarse level of word labels (Section 4.2 above) works as a basis-like function, trying to distribute word vectors in the directions of corresponding labels in the space through the training of the model; the similarity inequalities help the training by giving more information about the closeness among words.

The basic training model is continuous skip-gram architecture (Figure 5); our contribution to elaborate the trained word embeddings is incorporating the above idea into the training process, and the result word vectors improved the dialog system of this paper by predicting more precise intent of users' queries and filling the slots of structured queries more accurately.

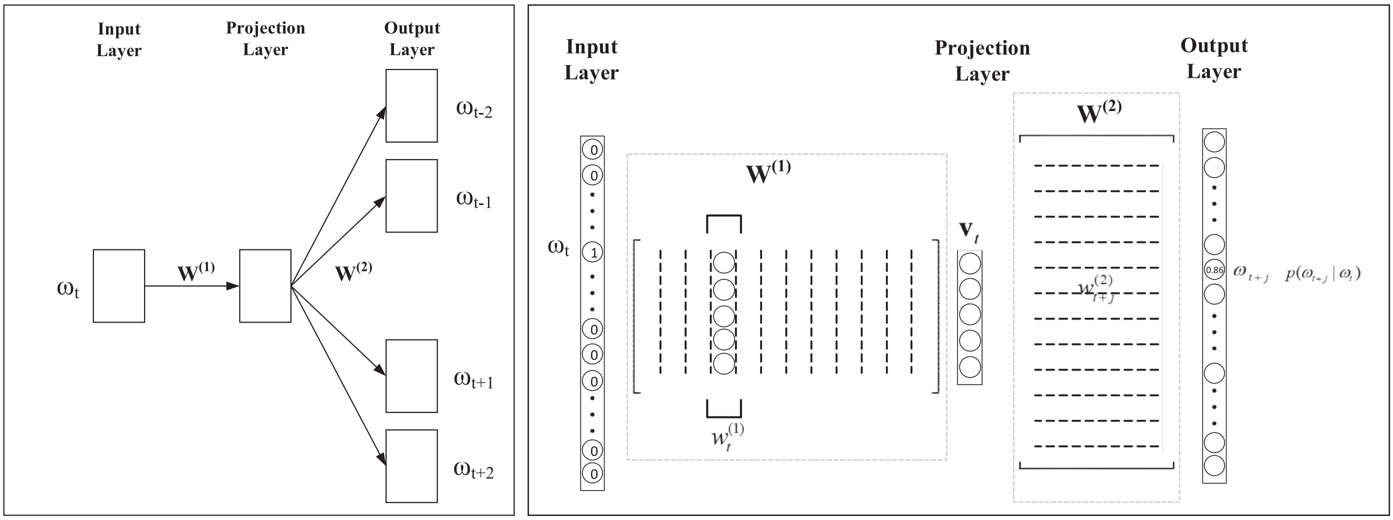
5.1. The Skip-Gram Model

The continuous skip-gram model proposed by Mikolov *et al.* [31,33] has a good balance between the quality of the resulting word vectors and the computational complexity of model. It is a simple three-layer feedforward neural network (Figure 5(a)), and the trained parameters are easily accessible.

Suppose V is the total number of the vocabulary, and k is the dimension of word vector ($k << V$). Then in Figure 5(a), (1) ω_t is a column vector of V dimension representing the current input word in one-hot expression; (2) $\omega_{t\text{-}1}$ and ω_{t+1} are output column vectors of V dimension representing the context words before and after ω_t of a sample sentence in the corpus, and the number of context vectors depends on the window size c of the context given in the training (we call c-window context and c = 2 in Figure 5(a)); (3) $\mathrm{W}^{(1)} \in \mathbb{R}^{k\times V}$ is the word embedding matrix, $\mathrm{W}^{(2)} \in \mathbb{R}^{V\times k}$ is the auxiliary matrix, and both of them are initialized randomly and updated in each training iteration; each column of final $\mathrm{W}^{(1)}$ is the continuous vector expression of corresponding word in the vocabulary (such as the column $W_t^{(1)}$ corresponding to the word ω_t in Figure 5(b)); $\mathrm{W}^{(2)}$ is used to form the predictive vectors of the context words in the output layer, the largest vector element (predictive probability) of which has the position corresponding to the one-hot expression of context words (such as the predictive probability 0.86 corresponding to the context word ω_{t+j} in Figure 5(b)); $\mathrm{W}^{(1)}$ and $\mathrm{W}^{(2)}$ can be expressed as:

$$\mathrm{W}^{(1)} = \left(w_1^{(1)}\ w_2^{(1)}\ \dots\ w_t^{(1)}\ \dots\ w_V^{(1)} \right) \tag{1}$$

$$\mathrm{W}^{(2)} = \begin{pmatrix} w_1^{(2)} \\ w_2^{(2)} \\ \dots \\ w_t^{(2)} \\ \dots \\ w_V^{(2)} \end{pmatrix} \tag{2}$$



(a) schematic diagram

(b) detailed version of the diagram

Figure 5 The continuous skip-gram model.

where as shown in Figure 5(b), the column vector $w_t^{(1)}$ of word embedding matrix $\mathrm{W}^{(1)}$ is corresponding to the target word ω_t of index t. Similarly, the row vector $w_{t+j}^{(2)}$ of another auxiliary word embedding matrix $\mathrm{W}^{(2)}$ is corresponding to one of its predictive context words ω_{t+j}.

Given a sample sequence $\omega_1, \omega_2, \omega_3, ..., \omega_T$ in the training set, where each ω_t represents the word of index t, suppose $\log p\left(\omega_{t+j} \mid \omega_t\right)$ denotes the log conditional probability of a word (or phrase) ω_{t+j} being within the c-window of context given token word (or phrase) ω_t, then $\sum_{-c \leq j \leq c, j \neq 0} \log p\left(\omega_{t+j} \mid \omega_t\right)$ is the union log conditional probability of all context words within c-window.

$$\mathrm{Q} = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p\left(\omega_{t+j} \mid \omega_t\right) \tag{3}$$

In the objective functional Equation (3), the goal of the continuous skip-gram model to maximize the average union log conditional probability of all context words within c-window for all words in the sample sequence, where the term $p\left(\omega_{t+j} | \omega_t\right)$ is defined by the softmax function (4).

$$p\left(\omega_{t+j} \mid \omega_t\right) = \frac{\exp\left(w_{\mathrm{t+j}}^{(2)} \cdot w_t^{(1)}\right)}{\sum_{n=1}^{\mathrm{V}} \exp\left(w_n^{(2)} \cdot w_t^{(1)}\right)} \tag{4}$$

In essence, the whole training process of word embeddings can be modeled as an optimization problem. With the help of optimization algorithms such as the stochastic gradient descent (SGD), the weights of matrix $\mathrm{W}^{(1)}$ and $\mathrm{W}^{(2)}$ are updated by the back-propagation, and the trained matrix $\mathrm{W}^{(1)}$ is adopted as the final word embedding matrix, in which the resulting word embeddings could be the feature input to follow-up joint RNN model in Section 6.3.

5.2. General-Label-Constrained Skip-Gram Model (LC-Skip-Gram)

Based on the continuous skip-gram model mentioned in Section 5.1, with general knowledge category labels as extra constraints, our neural network model is proposed for multi-objective training. As shown in Figure 6, the bottom block of "label$_t$" is added, which constitutes the second part of objective of the neural network, and which will constrain the parameters in matrix $\mathrm{W}^{(1)}$ and $\mathrm{W}^{(3)}$ in the training iterations, where similar to the matrix $\mathrm{W}^{(2)}$, $\mathrm{W}^{(3)} \in \mathbb{R}^{\mathrm{L} \times k}$ is another auxiliary matrix (L is the total number of categories of labels), which is used to form the predictive vectors of the general category labels in the output layer, the largest vector element (predictive probability) of which has the position corresponding to the one-hot expression of labels of the token words. $\mathrm{W}^{(3)}$ can be expressed as:

$$\mathrm{W}^{(3)} = \begin{pmatrix} label_1^{(3)} \\ label_2^{(3)} \\ \cdots \\ label_t^{(3)} \\ \cdots \\ label_L^{(3)} \end{pmatrix} \tag{5}$$

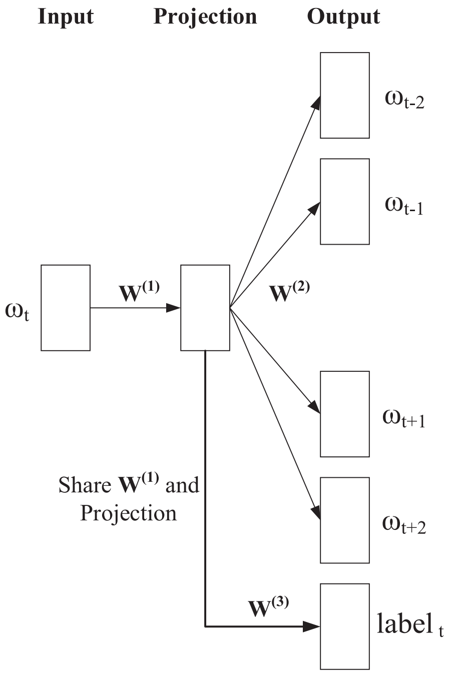


Figure 6 General-label-constrained skip-gram model.

The "label$_t$" for word ω_t is the coarse knowledge category label given in Section 4.2, and the compound objective function is a weighted sum shown below,

$$\mathrm{Q}_{LC} = (1 - \alpha) \cdot Q + \alpha \cdot \mathrm{Q}_{label} \tag{6}$$

where α is the weight of general category label part of constraints (here $\alpha = 0.5$ is taken); Q is the objective in Equation (3), and $\mathrm{Q}_{\mathrm{label}}$ is the additional part to predict the coarse knowledge category to which the token word ω_t belongs. $\mathrm{Q}_{\mathrm{label}}$ is formulated as,

$$\mathrm{Q}_{\mathrm{label}} = \frac{1}{\mathrm{T}} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p\left(\mathrm{label}_t \mid \omega_t\right) \tag{7}$$

where $\log p\left(\mathrm{label}_t | \omega_t\right)$ denotes the log conditional probability of label$_t$ given token word ω_t. $\mathrm{Q}_{\mathrm{label}}$ part is to maximize the average union log conditional probability of the general category labels of all token words in the sample sequence during the predication of their context words within c-window.

The joint objective training model shares the common embedding matrix $\mathrm{W}^{(1)}$ as well as the projection layer. The trained neural network would predict both the context of the token word and its general category label in the ontology. Consequently, the resulting word embeddings in $\mathrm{W}^{(1)}$ would incorporate semantics about general category of words and make them with the same label distributed more closely.

5.3. Label-and-Inequality-Constrained Skip-Gram Model (LIC-Skip-Gram)

The general-label-constrained skip-gram model moves one step forward in absorbing the knowledge of class hierarchy of "words (or phrases)" in a coarse affiliated level. If we labeled the "words (or phrases)" precisely to the bottom level as shown in Figure 3, each word would be urged to distribute in its own direction of corresponding label in the vector space without reflecting real hierarchical relationships in detail. As a matter of fact, the idea of basis for the word vector space mentioned above should result in less dimension than to label the words to bottom levels of class in the ontology hierarchy. To catch the different closeness between words in the hierarchy in the training process, we use ordinal inequalities of similarity between words as detailed in Section 4.2.

As a complement to the constraints of general category labels, ordinal semantic inequalities are incorporated as another set of more accurate and detailed constraints for the training of word vectors. As shown in Figure 7, besides the original prediction of the context word and general category label of token word, it is also constrained by those ordinal similarity relations between the token word and another two words with different semantic closeness respectively, which could be generated by the rules mentioned in Section 4.2, and make the resulting embeddings keep their real semantic hierarchy to reflect the different semantic closeness between them in a more precise way.

The compound objective function is formulated with another similar weighted sum shown as follows,

$$\mathrm{Q_{LIC}} = (1-\alpha-\beta)\cdot \mathrm{Q} + \alpha\cdot \mathrm{Q}_{label} + \beta\cdot \mathrm{Q}_{inequality} \tag{8}$$

$$\mathrm{Q}_{inequality} = \frac{1}{\mathrm{T}}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} \mathrm{H}_t \tag{9}$$

$$\mathrm{H}_t = \sum_{\{i,m,k\}\in S, t\in\{i,m,k\}} \max(\delta, sim(\omega_i,\omega_k) - sim(\omega_i,\omega_m)) \tag{10}$$

where β is the weight of inequality part of constraints (here $\beta = 0.1$ is taken); $Q_{inequality}$ is new additional part, whose constraints replace $\log p\left(\omega_{t+j}|\omega_t\right)$ with a new compound term H, namely the sum of the larger ones between the factor for control of similarity diversity δ (here $\delta = 0$ is taken) and the penalty terms consisting of all the inequalities related to the token word ω_t, as shown in the formula (10), where S is the universal set of indexes of vocabulary in corpora, and one of i, m, k is equal to t, the index of the word ω_t.

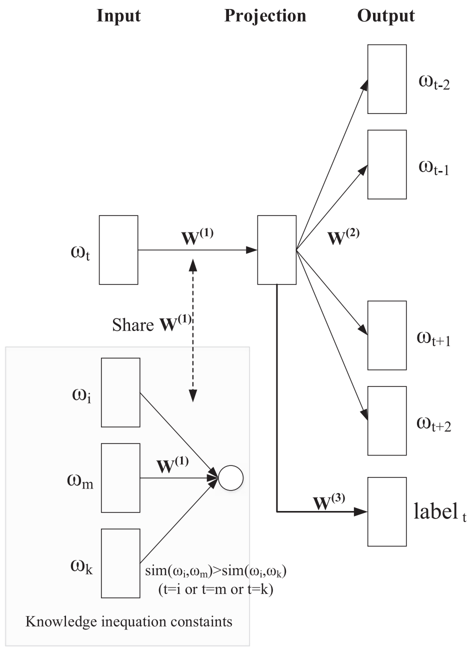


Figure 7 Label-and-inequality-constrained skip-gram model.

Note that the ordinal similarity inequalities mentioned above can be extracted in the form of triple $(\omega_i, \omega_m, \omega_k)$ with the method in Section 4.2, which are subject to the inequality $sim(\omega_i, \omega_k) - sim(\omega_i, \omega_m) < 0$. As a matter of fact, with the help of max function in the formula (10), it is semantically constrained only for those knowledge-inconsistent pairs with all the related inequalities as their penalty terms.

In this way, the idea turns into another optimization problem containing penalty terms, where the related words in inequalities set share the common embedding matrix $\mathrm{W}^{(1)}$ with each token word in the progress of training. Considering the more precise semantic relationships in ontology as inequalities constraints, the new model incorporates much more real, consistent and accurate semantics to produce word embedding with the best quality in these three methods.

6. EXPERIMENT

To compare and verify the quality of the word embeddings trained by the three models mentioned above, qualitative and quantitative experiments for evaluation are designed and carried out. In qualitative way, by comparing the cosine similarity of word embeddings trained by different models, the five nearest candidates of some typical words are collected to inspect how reasonable they are in the view of domain experts, also known as **intrinsic evaluations**. Despite qualitative analysis, the corresponding cosine similarity values as quantitative data are listed additionally to help validate the conclusion. In quantitative way, word embeddings are evaluated by using the specific task of intent detection and slot-filling in this case, also known as **extrinsic evaluations**.

6.1. Dataset

The corpora in this specific domain are elaborately prepared in Mandarin Chinese. To provide a natural and practical service of information retrieval, the related corpora was collected from plenty of volunteers' real inquiries about the information such as locations, persons, and activities, when they explore inside the building and interact with the mobile terminal. 1400 of the collected corpora was chosen and taken in this paper, of which 10 examples and their corresponding predicates are listed in the following Table 2. The corpora include 20 general intent (or predicates) and 26 slots elements, and one example is illustrated in Figure 8.

For instance, for the inquiry "*Which floor is Professor Wang's office on?*" as depicted in Figure 8, the segmented text sequence inputted into dialog system is firstly mapped to its consistent digitalized word vector representation by looking up the trained word embedding matrix in Section 5. With the help of the trained joint RNN in the later Section 6.3, the digitalized sequence is then converted into its general intent "*hasFloor*" and related slots sequence "*o* | *location* | *o* | *person* | *family_name* | *location* | *o*."

6.2. Qualitative Experiment (Intrinsic Evaluation)

The words of this specific domain, especially those concerning locations, persons, and activities, are randomly picked up as reference word representatives, as presented in the first left column of Table 3. In order to check whether it is more reasonable for the word embeddings trained by the knowledge-powered models than the baseline model, the five nearest candidates of reference word representatives are evaluated by comparing the cosine similarity of their trained word embeddings.

Suppose ω_1 and ω_2 are two words. The cosine similarity between ω_1 and ω_2 can be defined in the formula (11), where v_1 and v_2 are the vector representations for ω_1 and ω_2 respectively in the word embedding space.

$$similarity(\omega_1, \omega_2) = \frac{v_1 \cdot v_2}{||v_1|| \cdot ||v_2||} \tag{11}$$

Table 2 Examples in domain corpora and their corresponding general intents (or predicates).

Domain Corpora Example	Predicate
How can I get to the nearest toilet?	indoorHCI:hasRoute
How can I get to the Professor Wang's office?	indoorHCI:hasRoute
Which recruitment fairs can we masters attend this week?	indoorHCI:engaged_in
Where does Teacher Li work on the 5th floor and block A?	indoorHCI:located_in
Which room will the meeting be held this afternoon?	indoorHCI:occur_in
What is the topic of this lecture?	indoorHCI:hasTopic
Who are the postgraduates of Associate Professor Zhang?	indoorHCI:hasStudent
Which students will attend the math class today?	indoorHCI:hasPaticipant
What time does the dissertation defense begin this morning?	time:hasTime
Which floor is Professor Wang's office on?	indoorHCI:hasFloor

Table 3 shows the five nearest candidates of words from word embeddings trained by three different models, namely skip-gram, LC-skip-gram and LIC-skip-gram model, where the candidates are sorted in descending order by the cosine similarity and bold words are highly related to the corresponding reference words in terms of the semantic closeness. Figure 9 shows the resulting spatial distribution of two-dimensional t-SNE [34] word embeddings trained by these three models, where the legends are consistent with the general category labels in Section 4.2.

As indicated in Table 3 and Figure 9, compared with baseline skip-gram model, LC-skip-gram and LIC-skip-gram model both bring more accurate word embeddings enhanced with knowledge, where the semantically related words, such as synonyms, hypernyms, and hyponyms, have a more and more concentrated spatial distribution, especially for the latter one exhibiting categorical difference in an obvious way and having the best expressiveness, which meets our expectations.

More specifically, as shown in the bold parts in Table 3, the word embeddings by the conventional skip-gram model cannot make a clear distinction between words of different categories, even some irrelevant words play a leading role, such as "person in charge." For word embeddings from LC-skip-gram model, there are more semantically related words appearing in the five nearest candidates list with higher similarity than those from the former baseline model, however, there are still some noise words in it, such as the candidate word "lab" with regard to the reference word "meeting." With both category labels and inequalities as extra constraints, LIC-skip-gram model reaches the best performance out of the three models. The word candidates are all highly semantically related and consistent with the domain ontology.

Most of the natural spoken corpora are irregular and flexible, where there are not the fixed relative positions between context words as is the often case. The baseline skip-gram model exploits the co-occurrence information in contexts as constraints to train the word embeddings, which have limited expressiveness due to plenty of implicit noise in the training procedure. Furthermore, based on the original context constraints, the two novel knowledge-enhanced models LC-skip-gram and LIC-skip-gram make use of new semantic knowledge as additional constraints, including general category

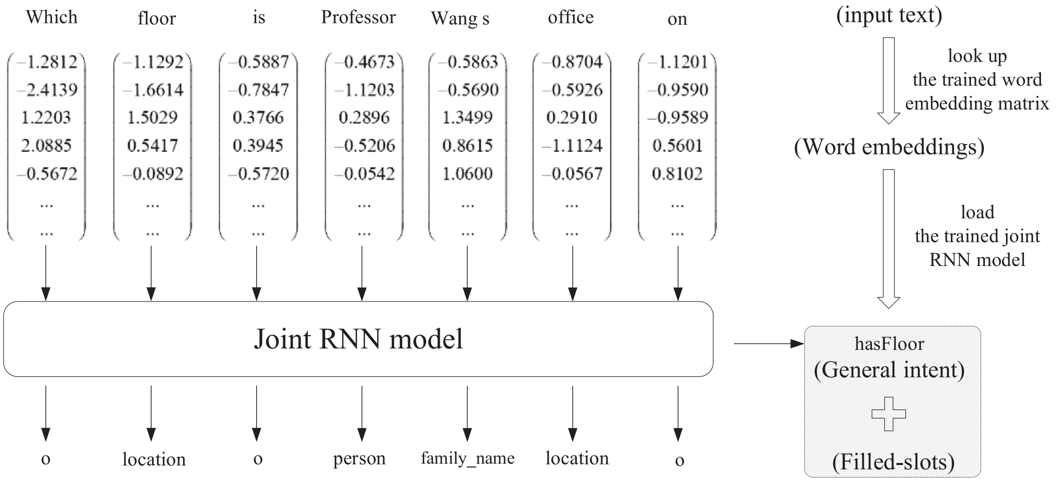
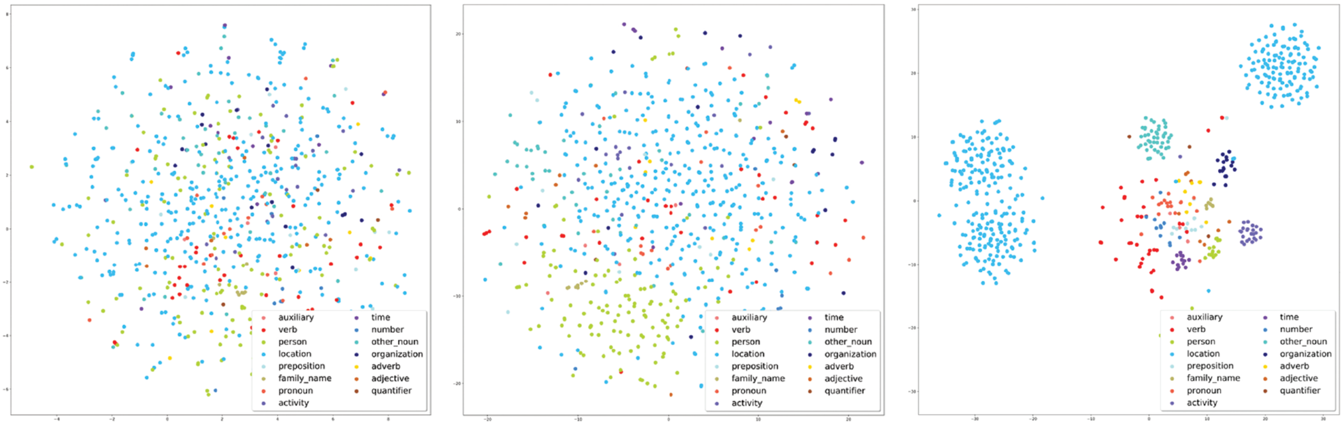


Figure 8 An example of semantic understanding from input text to its general intent and filled-slots.

Table 3 The five nearest candidates of words from word embeddings learnt by three models.

Reference Word	Skip-gram		LC-skip-gram		LIC-skip-gram	
Doctor	**Master**	0.934	**Master**	0.981	**PhD**	0.997
	Major	0.892	**PhD**	0.958	**Master**	0.993
	PhD	0.860	Guest	0.923	**Graduate student**	0.992
	Graduate student	0.773	**Graduate student**	0.901	**Postgraduate**	0.990
	Name	0.725	Major	0.819	**Student**	0.987
Meeting	**Lecture**	0.794	**Lecture**	0.906	**Forum**	0.851
	Fire plug	0.715	Location	0.803	**Conference**	0.847
	Stairwell	0.711	**Job fair**	0.778	**Activity**	0.805
	Lab	0.694	Lab	0.771	**Course**	0.773
	Watering place	0.672	Watering place	0.758	**Exam**	0.766
Office	**Lab**	0.795	**Lab**	0.814	**Lab**	0.971
	Stairwell	0.762	**Stairwell**	0.720	**Meeting room**	0.954
	Watering place	0.737	**Elevator gate**	0.696	**Watering place**	0.950
	Competition	0.685	**Watering place**	0.672	**Toilet**	0.918
	Location	0.666	**Location**	0.672	**Room**	0.906
Wang (one of family names)	**Zhang**	0.640	**Liu**	0.790	**Liu**	0.980
	Li	0.537	**Li**	0.775	**Zhang**	0.979
	Dean	0.528	**Zhang**	0.762	**Li**	0.976
	Liu	0.517	Instructor	0.611	**Kong**	0.972
	Instructor	0.499	Math course	0.577	**Zhao**	0.966
Teacher	**Professor**	0.645	**Associate professor**	0.756	**Instructor**	0.980
	Associate professor	0.572	**Professor**	0.683	**Associate professor**	0.980
	Stairwell	0.538	**Host**	0.562	**Professor**	0.978
	Fire plug	0.534	**Keynote speaker**	0.557	**Lecturer**	0.974
	Host	0.516	**Lecturer**	0.524	**Master**	0.971
Person in charge	Topic	0.937	**Host**	0.975	**Host**	0.992
	Requirement	0.922	**Keynote speaker**	0.955	**Chairman**	0.990
	Topic	0.912	Gender	0.877	**Participant**	0.988
	Keynote speaker	0.907	Route	0.874	**Keynote speaker**	0.987
	Host	0.906	Classroom	0.871	**Audience**	0.980



(a) word embeddings trained by skip-gram model

(b) word embeddings trained by LC-skip-gram model

(c) word embeddings trained by LIC-skip-gram model

Figure 9 The resulting spatial distribution of two-dimensional t-SNE word embeddings trained by three models.

labels and inequalities extracted from domain ontology, so as to reduce the influence of noise and make the semantically related words closer and the unrelated words more dispersed.

6.3. Joint RNN Model for Intent Detection and Slot-Filling

As crucial steps of NLU, the problems of intent detection and slot-filling have attracted plenty of attentions in the past few years. The semantic contents of a sentence inputted into dialog systems consist of key words and the relations between them, which can be represented as RDF triples in the form of <*subject, predicate, object*>, where the center *predicate* indicates the general intent, and the *subject* and *object* are related arguments. In essence, the problems of intent detection and slot-filling can be treated as a classification problem and a sequential labeling problem respectively, which could be addressed by a joint model to simplify this NLU module.

The paper [35] creatively trained and fine-tuned only one model for these two tasks simultaneously in the ATIS (Airline Travel Information Systems) data set. Inspired by this architecture, an attention-based [36] encoder–decoder neural network is used for both intent detection and slot-filling, as illustrated in Figure 10. A bidirectional RNN is employed on the encoder side, where the Long Short-Term Memory (LSTM) [37] is taken as its basic unit to promote the ability of modeling the long-term dependencies.

In the process of slot-filling, a word sequence $x = (x_1, x_2, ..., x_T)$ (T = 7 in Figure 10) is intended to map to its corresponding label sequence $y = (y_1, y_2, ..., y_T)$. The encoder part of RNN reads the source of word sequence forward in the original order and backward in the reverse order respectively, and eventually generate their corresponding sequences of hidden states $(fh_1, fh_2, ..., fh_T)$ and $(bh_T, bh_{T-1}, ..., bh_1)$. At each time step i, the final resulting hidden state h_i is a concatenation of the forward hidden state fh_i and back hidden state bh_i, namely $h_i = [fh_i, bh_i]$. In this way, the final hidden state h_T conveys the information of the whole sequence, therefore taken as input to the later classifier to predict the general intent of word sequence.

The decoder is a single forward direction RNN model with the LSTM as its basic unit. At each time step i, the hidden state s_i is a compound function affected by four prats: the aligned encoder hidden state h_i, the previous hidden state s_{i-1}, the previous predicted label y_{i-1}, and the context factor c_i,

$$s_i = f(h_i, s_{i-1}, y_{i-1}, c_i) \tag{12}$$

where c_i represents the attention mechanism factor and is a weighted sum of the encoder states $h = (h_1, h_2, ..., h_T)$ as follows,

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \tag{13}$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})} \tag{14}$$

$$e_{i,k} = g(s_{i-1}, h_k) \tag{15}$$

For this joint model, the same encoder is shared by intent detection decoder and slot-filling decoder, the costs from which are both back-propagated to this common encoder.

6.4. Quantitative Experiment (Extrinsic Evaluation)

As the feature inputs, the trained word embeddings are exploited by the joint RNN model in Section 6.3, which facilitates the tasks of intent detection and slot-filling. As a result, the quality of word embeddings can be measured by the accuracy (Acc) rate of intent detection and the F1 score of slot-filling as well as its precision (P) and recall (R), calculated with the Equations (16–19) and shown in Table 4 and Figure 11 as follows. The corpora are divided into the train set and test set at the rate of 8:2.

$$P = \frac{TP}{TP + FP} \tag{16}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{18}$$

$$Acc = \frac{correct\ number\ of\ predictions}{total\ number\ of\ predictions} \tag{19}$$

where TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative.

Table 4 Performance comparison of intent detection and slot-filling from word embeddings trained by different models.

	Intent Detection	Slot-filling		
The Training Model of Word Embeddings	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**
Random	87.50	90.02	90.22	90.12
Skip-gram	88.64	92.67	92.79	92.73
LC-skip-gram	90.91	93.06	93.34	93.20
LIC-skip-gram	**95.45**	**97.12**	**97.38**	**97.25**

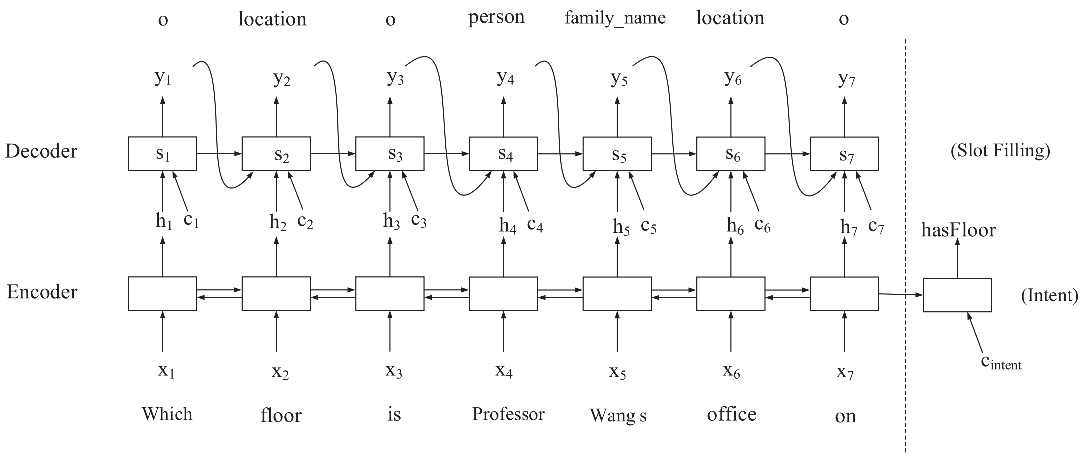
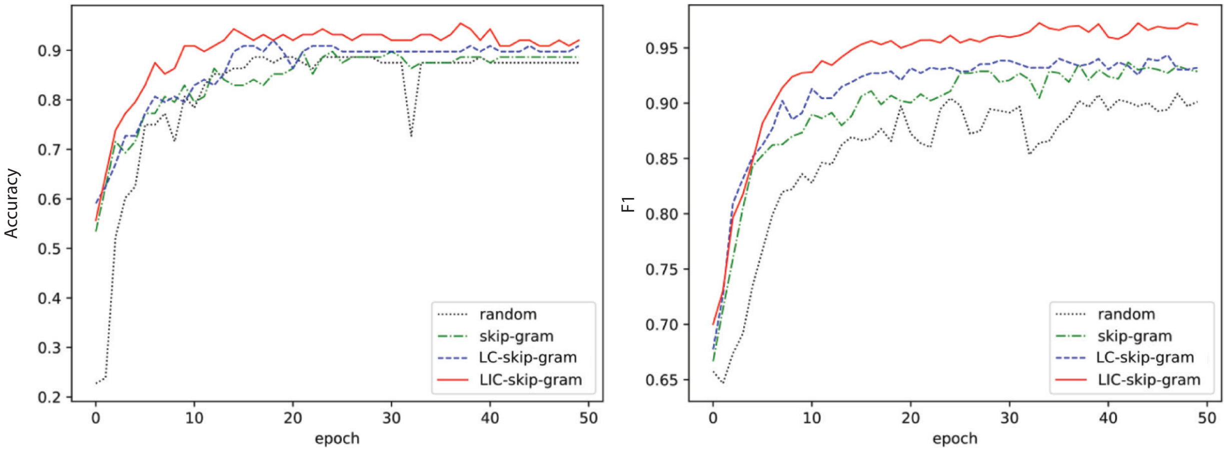

o
location
o
person
family_name
location
o
y_1 y_2 y_3 y_4 y_5 y_6 y_7
Decoder
s_1 s_2 s_3 s_4 s_5 s_6 s_7
(Slot Filling)
h_1 c_1 h_2 c_2 h_3 c_3 h_4 c_4 h_5 c_5 h_6 c_6 h_7 c_7
hasFloor
Encoder
(Intent)
c_{intent}
x_1 x_2 x_3 x_4 x_5 x_6 x_7
Which
floor
is
Professor
Wang s
office
on

Figure 10 The joint encoder–decoder neural network for intent detection and slot-filling.



(a) Intent detection accuracy

(b) F1-values of slot-filling

Figure 11 Performance comparison of intent detection and slot-filling from word embeddings trained by different models.

It can be seen that as feature inputs, word embeddings trained by the three models all make contributions to the task of intent detection and slot-filling, compared to random word embedding. Although random word embeddings almost don't convey any semantic information, its accuracy of intent detection and the F1 value of slot-filling still reach 87.5% and 90.02% respectively, which are high enough as the baseline. The reason is that under the supervision of label sequence, RNN model does well in processing time-varying sequence and extracting features, such as syntactic information, where semantic information is not the only and main contributor. Random word embeddings convey no semantics, which may decrease the convergence speed of model training (see the black dot line in Figure 11), however, with supervisory label sequence as its powerful assistance, the trained RNN model still could reach a good performance.

From another perspective, compared with the syntactic part, it is semantic one that has more potential to increase the convergence speed of model training and further enhance the performance of RNN model. As indicated in Table 4 and Figure 11, comparing with the baseline, skip-gram, LC-skip-gram and LIC-skip-gram model all further promote the task and achieve a better performance step by step. Among the three models, LIC-skip-gram model achieve the best performance with 95.45% and 97.25% respectively in the two tasks, which meets our expectation. With conveying certain syntax and semantics as well as implicit noise, the word embeddings trained by skip-gram model only improve the performance a little. The two novel knowledge-enhanced models allow the word embeddings to convey more accurate semantics. Especially in the LIC-skip-gram model, the involved semantic noises are discarded partly with both general category labels and semantic inequalities containing penalty factors as additional constraints. With more precise semantics, the RNN model naturally further enhance performance based on the same sequence information.

In a word, knowledge-enhanced word embeddings are more suitable for applications in this specific field, and the more accurate semantic constraints the word embeddings are enhanced with, the better the users' queries are understood in the dialog system.

7. CONCLUSION

This paper provides a systematic route of information retrieval from an ontology-based domain knowledge database through a dialog system of natural language interaction in a comprehensive building at a university. In this case, the knowledge-based and data-driven approaches are integrated into one system, and the domain knowledge is in the central part which is incorporated into the word embedding to make it specifically fit the natural language in this application.

Based on the domain knowledge, two semantic knowledge constraints are constructed and integrated in the training process to improve the semantic information conveyed in the word embeddings. On the one hand, general knowledge category labels are exploited as extra constraints in the multi-objective training procedure to urge word vectors distributed in the directions of corresponding labels in the space; on the other hand, as a complement to coarse category labels, more detailed semantic closeness among terms are employed in form of ordinal inequality constraints, in which penalty terms make a difference in separating semantically irrelevant words and gathering the related ones together. In this way, word embeddings are enhanced with the domain knowledge to specifically fit the natural language in this application. As feature inputs, these trained word embeddings naturally enhance the performance of predicting general intent and concrete filled-slots of user's question with the help of a trained joint RNN model.

Moreover, experiments are conducted to measure the quality of these trained word embeddings by different models in both qualitative and quantitative way. It demonstrates that the knowledge-enhanced word embeddings are more accurate and expressive in conveying semantic information, especially those trained by LIC-skip-gram model, which perform the best for the semantic understanding in the dialog system.

Word embeddings enhanced with multi-granularity domain knowledge better facilitate the semantic understanding and make the dialog process more fluent. The systematic route in this case provides reference for oral guidance system in many occasions such as tourist sites, shopping centers, public libraries, hospitals, museums, residence communities, and so on.

CONFLICT OF INTEREST

The authors declare they have no conflicts of interests.

AUTHORS' CONTRIBUTIONS

Jin Ren contributed the central idea, designed and conducted most of the experiments, analyzed and interpreted the results and wrote the initial draft of the paper. Hengsheng Wang made significant contributions to refining the ideas and revising the manuscript. Tong Liu contributed to collecting the data and conducting the quantitative experiment. All authors discussed the results.

ACKNOWLEDGEMENTS

This work has been supported by the National Basic Research Program of China (973 Program, 2013CB035504) and Hunan Provincial Innovation Foundation for Postgraduate (CX2017B049). We thank the editor and two anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Eisenstein, Natural Language Processing, MIT Press, Georgia Tech, USA, 2018, pp. 20–22.

[2] T. Young, *et al.*, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (2018), 55–75.

[3] S. Lai, *et al.*, How to generate a good word embedding, IEEE Intell. Syst. 31 (2016), 5–14.

[4] Z.S. Harris, Distributional structure, Word. 10 (1954), 146–162.

[5] Q. Liu, *et al.*, Learning semantic word embeddings based on ordinal knowledge constraints, in Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics, Beijing, China, 2015.

[6] M.C. Yang, *et al.*, Knowledge-based question answering using the semantic embedding space, Expert Syst. Appl. 42 (2015), 9086–9104.

[7] J. Bian, B. Gao, T. Liu, Knowledge-powered deep learning for word embedding, in: T. Calders, F. Esposito, E. Hüllermeier, R. Meo (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer, Berlin, Germany, 2014, pp. 132–148.

[8] B. Hu, *et al.*, A novel word embedding learning model using the dissociation between nouns and verbs, Neurocomputing. 171 (2016), 1108–1117.

[9] M. Liu, *et al.*, Measuring similarity of academic articles with semantic profile and joint word embedding, Tsinghua Sci. Technol. 22 (2017), 619–632.

[10] M. Faruqui, *et al.*, Retrofitting word vectors to semantic lexicons, in Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 2015.

[11] M. Faruqui, *et al.*, Sparse overcomplete word vector representations, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015.

[12] A. Bordes, *et al.*, Learning structured embeddings of knowledge bases, in Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[13] Q. Wang, *et al.*, Knowledge graph embedding: a survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (2017), 2724–2743.

[14] Z. Wang, *et al.*, Knowledge graph and text jointly embedding, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014, pp. 1591–1601.

[15] K. Toutanova, *et al.*, Representing text for joint embedding of text and knowledge bases, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015, pp. 1499–1509.

[16] Z. Zhang, *et al.*, ERNIE: enhanced language representation with informative entities, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019.

[17] G. Tur, R. De Mori, Spoken Language Understanding: Systems for Extracting Semantic Information from Speech, John Wiley & Sons, Hoboken, NJ, USA, 2011.

[18] M. McTear, Z. Callejas, D. Griol, Spoken language understanding, in: M. McTear, Z. Callejas, D. Griol (Eds.), The Conversational Interface, Springer, Berlin, Germany, 2016, pp. 161–185.

[19] D. Gunning, S. Mark, C. Jaesik, *et al.*, Explainable Artificial Intelligence (XAI), Science Robotics, 4 (2019), y7120.

[20] M.S. Yakoub, S. Selouani, R. Nkambou, Mobile spoken dialogue system using parser dependencies and ontology, Int. J. Speech Technol. 18 (2015), 449–457.

[21] D. Chen, C.D. Manning, A fast and accurate dependency parser using neural networks, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014, pp. 740–750.

[22] R. Cai, *et al.*, An encoder-decoder framework translating natural language to database queries, in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2017.

[23] T. Rocktäschel, *et al.*, Low-dimensional embeddings of logic, in Proceedings of the ACL 2014 Workshop on Semantic Parsing, Baltimore, MD, USA, 2014, pp. 45–49.

[24] B. Chandrasekaran, J.R. Josephson, V.R. Benjamins, What are ontologies, and why do we need them? IEEE Intell. Syst. Appl. 14 (1999), 20–26.

[25] H. Wang, J. Ren, A semantic map for indoor robot navigation based on predicate logic, Int. J. Knowl. Syst. Sci. 11 (2020), 1–21.

[26] G. Sriharee, Indoor navigation using semantic symbolic information, in Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, Australia, 2013, pp. 1167–1173.

[27] K. Lee, J. Lee, M.P. Kwan, Location-based service using ontology-based semantic queries: a study with a focus on indoor activities in a university context, Comput. Environ. Urban Syst. 62 (2017), 41–52.

[28] A. Ameen, K.U.R. Khan, B.P. Rani, Construction of university ontology,in 2012 World Congress on Information and Communication Technologies, Trivandrum, India, 2013, pp. 39–44.

[29] J. Scholz, S. Schabus, An indoor navigation ontology for production assets in a production environment, in International Conference on Geographic Information Science, Vienna, Austria, 2014, pp. 204–220.

[30] W. Ertel, Logic programming with PROLOG, in: W. Ertel (Ed.), Introduction to Artificial Intelligence, Springer, Berlin, Germany, 2017, pp. 75–90.

[31] T. Mikolov, *et al.*, Distributed representations of words and phrases and their compositionality, in Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, pp. 3111–3119.

[32] T. Mikolov, W.T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 2013.

[33] T. Mikolov, *et al.*, Efficient estimation of word representations in vector space, in Proceedings of the International Conference on Learning Representations, 2013.

[34] L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008), 2579–2605.

[35] B. Liu, I. Lane, Attention-based recurrent neural network models for joint intent detection and slot filling, in Proceeding of Interspeech 2016, 2016.

[36] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2014.

[37] F.A. Gers, J.U.R. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, in Ninth International Conference on Artificial Neural Networks (ICANN), Edinburgh, UK, 1999.