Structured Knowledge Discovery from Massive Text Corpus

BY

CHENWEI ZHANG
B.E., Southwest University, 2014

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:
Professor Philip S. Yu, Chair and Advisor
Professor Bing Liu
Professor Piotr Gmytrasiewicz
Professor Caragea Cornelia
Professor Jiawei Zhang, Department of Computer Science, Florida State University

This dissertation is dedicated to my parents and grandparents,

for their unconditional love and support.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my Ph.D. advisor, Prof. Philip S. Yu, for his guidance and support throughout my Ph.D. study and research. It has been my privilege to work with you at different aspects of my Ph.D. journey. Your invaluable suggestions, guidance and your passion for research not only help me with my past academic achievements but also will influence my professional career in the future.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Bing Liu, Prof. Piotr Gmytrasiewicz, Prof. Cornelia Caragea, and Prof. Jiawei Zhang, for your valuable time serving as my dissertation committee members.

I am grateful to Prof. Yong Deng at Southwest University, for the mentorship during my early research career and enlightening me the first glance of research. My sincere thank goes to Dr. Wei Fan, who mentored me when I was a research intern at Baidu Research and Tencent America. Without your continuous support, this dissertation would not have been possible. I would like to thank Dr. Nan Du and Dr. Yaliang Li for invaluable suggestions and fruitful discussions during our collaborations in various research projects, which relate to this dissertation.

I would like to express gratitude to my fellow lab mates in the Big Data and Social Computing Lab at the University of Illinois at Chicago, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last five years. My warmest thanks extend to all the collaborators, colleagues and friends that I met at the University of Illinois at Chicago.

Last but not least, none of this could have happened without my family. I am grateful for my parents and grandparents, for their unconditional love and encouragement.

CZ

CONTRIBUTION OF AUTHORS

Chapter 1 is an introduction that outlines my dissertation research.

Chapter 2 presents published papers (Zhang et al., 2016; Zhang et al., 2017) for which I was the primary author. Dr. Nan Du, Dr. Wei Fan, Dr. Yaliang Li, Dr. Chun-Ta Lu, and Prof. Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

Chapter 3 presents a published paper (Zhang et al., 2019), for which I was the primary author. Dr. Yaliang Li, Dr. Nan Du, Dr. Wei Fan, and Prof. Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

Chapter 4 presents a published paper (Zhang et al., 2018a) for which I was the primary author. Dr. Yaliang Li, Dr. Nan Du, Dr. Wei Fan, and Prof. Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

Chapter 5 presents a published paper (Zhang et al., 2018b) for which I was the primary author. Dr. Yaliang Li, Dr. Nan Du, Dr. Wei Fan, and Prof. Philip S. Yu contributed to discussions with respect to the work and revising the manuscript.

Chapter 6 concludes this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Nowadays, with the booming development of the Internet, people benefit from its convenience due to its open and sharing nature. A large volume of natural language texts is being generated by users in various forms, such as search queries, documents, and social media posts. As the unstructured text corpus is usually noisy and messy, it becomes imperative to correctly identify and accurately annotate structured information in order to obtain meaningful insights or better understand unstructured texts. On the other hand, the existing structured information, which embodies our knowledge such as entity or concept relations, often suffers from incompleteness or quality-related issues. Given a gigantic collection of texts which offers rich semantic information, it is also important to harness the massiveness of the unannotated text corpus to expand and refine existing structured knowledge with fewer annotation efforts.

In this dissertation, I will introduce principles, models, and algorithms for effective structured knowledge discovery from the massive text corpus. We are generally interested in obtaining insights and better understanding unstructured texts with the help of structured annotations or by structure-aware modeling. Also, given the existing structured knowledge, we are interested in expanding its scale and improving its quality harnessing the massiveness of the text corpus. In particular, four problems are studied in this dissertation: Structured Intent Detection for Natural Language Understanding, Structure-aware Natural Language Modeling, Generative Structured Knowledge Expansion, and Synonym Refinement on Structured Knowledge.

CHAPTER 1

INTRODUCTION

1.1 Dissertation Outline

Nowadays, with the booming development of the Internet, people benefit from its convenience due to its open and sharing nature. A wide range of user goals is fulfilled on the Internet through various forms of interactions such as web search, web chats, social media postings and so on. The abundant text corpus that is available online embodies rich knowledge that is to be discovered. Due to the open, sharing nature of the Internet and different linguistic preferences of individuals, the gigantic collection of unstructured text corpus is usually noisy and messy. It is challenging yet rewarding to correctly identify and accurately annotate structured information in order to obtain meaningful insights or better understand the massive unstructured texts.

The structured information summarizes our existing knowledge in a structured manner, which is ubiquitously accessible for both machine and human beings. We may introduce triplets that contain factual relationships among entities as the structured information in knowledge graphs. For example, `Barack Obama` as an entity has a semantic relation `president of` with another entity `U.S.A`. The structured information could be also on the concept level, where the triplets introduce semantic relationships between concepts. For example, we may have `medicine` as a concept with a relation `cure` to another concept `disease`. Besides that, the

structured information can contain both entity and concept level information in a hierarchical structure, such as `Barack Obama` as an entity may connect to `Politician` as a concept. Many researchers in academia and industry are striving to obtain high-quality structured knowledge, such as WordNet (Miller, 1995), Yago (Fabian et al., 2007), Freebase (Bollacker et al., 2008), ConceptNet (Speer and Havasi, 2012), and SenticNet (Cambria et al., 2018).

However, the existing structured knowledge often suffers from incompleteness and quality-related issues. As obtaining high-quality structured information for knowledge discovery is usually time-consuming and labor-intensive, it is thus important to automatically expand and refine the structured information exploiting the massiveness of unannotated text corpus.

The contributions of this dissertation are made toward two strongly correlated, synergistic objectives:

- **Utilizing Structured Information for Natural Language Understanding and Modeling**: Given a massive unannotated text corpus, we are interested in obtaining insights, understanding and modeling the texts with the help of existing structured annotations or by structure-aware modeling.
- **Expanding and Refining Structured Knowledge Harnessing the Massiveness of the Text Corpus**: Given the existing structured knowledge, we are interested in expanding the scale and improving the quality of structured knowledge, where additional human annotation efforts are minimized via harnessing the massive collection of the unannotated text corpus.

In particular, four problems are studied in this dissertation: Structured Intent Detection for Natural Language Understanding, Structure-aware Natural Language Modeling, Generative Structured Knowledge Expansion, and Synonym Refinement on Structured Knowledge.

- To better understand complicated user intents from their diversely expressed natural language utterances, we utilize concept-level structured knowledge and treat intent detection on unannotated text utterances as a structured prediction problem.
- To extract both word-level and sentence-level semantics while preserving their structural relationships, we provide a structure-aware approach that jointly annotates word-level concept mentions and sentence-level intent labels for each utterance.
- To expand the scale of high-quality structured knowledge and reduce data preparation efforts, we introduce a generative modeling approach that harnesses word-level semantics learned from the massive text corpus for structured knowledge expansion.
- To improve the quality of the existing structured knowledge, we refine it by removing synonymous entities. We introduce a framework that detects entity synonyms by comparing among contexts in which entities are mentioned from a massive text corpus.

1.2 Structured Intent Detection for Natural Language Understanding

(Part of this chapter was previously published in (Zhang et al., 2016; Zhang et al., 2017).)

Unstructured texts generated by users in their web search or social media posts are naturally encoded with users' information-seeking intents. To better understand texts generated by users, an intent detection task aims to categorize the text corpus according to intents. Unlike

conventional topic classification tasks where the label of the text is highly correlated with some topic-specific concept words, words from different concept categories tend to co-occur in a single piece of information-seeking utterance. When the user tries to express more information in a single piece of utterance, the intent also becomes complicated: the users mention multiple concepts and semantic transitions emerge among multiple concepts.

In Chapter 2, first we formally define the user intent as a semantic transition between two concepts. For complicated utterances, we further utilize a concept-level intent graph and formulate intent detection as a structured prediction problem: a structured intent is defined as a sub-graph over the pre-defined concept-level intent graph where each node represents a concept mention and each directed edge indicates a semantic transition. A multi-task neural network model is proposed: one task extracts concept mentions, and the other task infers semantic transitions from the utterance. A customized graph-based mutual transfer loss function is designed to impose explicit constraints over two subtasks for collective inference.

1.3 Structure-aware Natural Language Modeling

(Part of this chapter was previously published in (Zhang et al., 2019))

Being able to recognize words as slots and detect the intent of an utterance has been a keen issue in natural language understanding. Existing works either treat word-level slot filling and utterance-level intent detection separately in a pipeline manner, or adopt joint models which sequentially label slots while summarizing the utterance-level intent without explicitly preserving the semantic hierarchy among words on the word level, slots on the concept level, and intents on the utterance level. In Chapter 3, to exploit the semantic hierarchy for effective

natural language modeling, we investigate a structure-aware approach that accomplishes slot filling and intent detection in a bottom-up fashion via a dynamic routing-by-agreement schema. The model does slot filling by learning to assign each word on the word-level to the most appropriate slot on the concept-level via dynamic routing. The dynamic routing also aggregates concept-level slot representations to predict the utterance-level intent. As the intent of the utterance may also help recognize words as different slots, a re-routing schema is proposed that further synergizes the word-level slot filling performance using the inferred utterance-level intent in a top-down fashion.

1.4 Generative Structured Knowledge Expansion

(Part of this chapter was previously published in (Zhang et al., 2018a).)

When knowledge graph is becoming an indispensable resource that offers rich structured information for numerous knowledge-intensive applications, it often suffers from incompleteness issues. Building a complete, high-quality knowledge graph is time-consuming and requires significant human annotations. Previously, most knowledge graph completion methods use discriminative classifiers that extract triplets directly from corpus where certain relations are expressed. When the knowledge graph is in its infancy, we lack sufficient and high-quality annotations on the text corpus for existing discriminative models to excel.

To reduce human annotation efforts for structured knowledge expansion, in Chapter 4 we introduce a generative perspective to increase the scale of high-quality structured knowledge and study the Structured Knowledge Expansion task. The proposed model explores the generative modeling capacity for entity pairs and harnesses word-level semantics learned from the massive

text corpus for structured knowledge expansion. It is able to generate meaningful entity pairs that are not yet observed and efficiently expand the scale of structured knowledge.

1.5 Synonym Refinement on Structured Knowledge

(Part of this chapter was previously published in (Zhang et al., 2018b).)

Currently, information extraction systems can automatically extract structured knowledge from a large collection of text corpus. However, the task to extract information is challenging and current systems make many mistakes: ambiguous, redundant or conflicting entity information are prevalently observed during the construction of structured knowledge. Given an existing knowledge graph, a lot of human annotation efforts are being made to improve the quality of the extracted knowledge.

To improve the quality of the existing structured knowledge, in Chapter 5 we propose to remove duplicated and redundant entity information in an existing knowledge graph. Previous works on detecting synonymous entities focus on learning the similarity between entities using character-level features. These methods work well for synonyms that share a lot of character-level features like `airplane/aeroplane`. However, a much larger number of synonym entities in the real-world do not share a lot of character-level features, such as `JD/law degree`. Instead of relying on excessive human annotations, we propose to leverage the free-text contexts in which entities are mentioned in a gigantic collection of text corpus for effective synonym detection. Instead of using entities features, a novel neural network model is proposed which makes use of multiple pieces of contexts in which the entity is mentioned, and compares the context-level similarity via a bilateral matching schema to determine synonymity.

CHAPTER 2

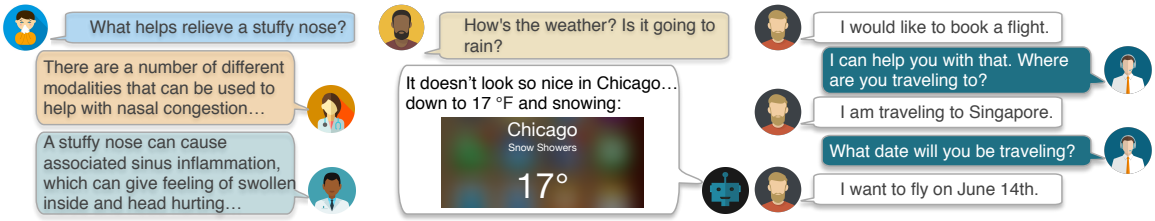
Structured Intent Detection for Natural Language Understanding

This chapter was previously published as "Mining User Intentions from Medical Queries: A Neural Network based Heterogeneous Jointly Modeling Approach" in WWW'16 (Zhang et al., 2016), DOI: https://doi.org/10.1145/2872427.2874810, and "Bringing Semantic Structures to User Intent Detection in Online Medical Queries" in BigData'17 (Zhang et al., 2017). DOI: https://doi.org/10.1109/BigData.2017.8258025.

2.1 Introduction

A wide range of user goals is fulfilled on the Internet through various forms of interactions such as web search, web chats and so on. For example, online question answering websites are able to offer globally accessible information via human-human interactions. As voice assistants and chat-bots become more and more popular, users may ask smart devices questions via voice commands. In service center question answering systems, customers express their requests and get their tasks resolved. For example, booking a flight with customer service representatives. Figure 1 illustrates three scenarios on community Q&A, voice assistant/chatbot, and service center Q&A.

With various forms of interactions, a huge amount of text corpus are generated by users. The text corpus generated by users, usually consists of declarative statements followed by questions, are naturally encoded with users' information-seeking intentions. An intent detection task tries



(a) Community Q&A (b) Voice AssistantChatbot (c) Service Center Q&A

Figure 1: Intent Detection from various types of user-generated utterances.

to model and discover intentions that a user encodes in the text corpus. Unlike conventional text classification tasks where the label of text is highly correlated with some topic-specific words, words from different topic categories tend to co-occur in questions generated by users for information-seeking purposes. Besides the existence of topic-specific words and word order, word correlations and the way words are organized in the corpus are crucial to the intent detection task. Due to different linguistic preferences of individuals, the intentions can be expressed partially, implicitly or diversely, which makes it challenging to accurately understand user intentions from the text corpus.

Text	Intent
I have (got) a fever, should I take the Tylenol?	*Symptom* → *Medicine*
Which medicine should I take if I'm running a fever?	*Symptom* → *Medicine*
I've come down with a fever, should I take Aspirin?	*Symptom* → *Medicine*
Is it okay to use ibuprofen when I'm running a temperature?	*Symptom* → *Medicine*
My temperature is 103, can I use Advil?	*Symptom* → *Medicine*

TABLE I: Utterances with the same intent but different expressions.

Specifically, the intention we studied in this work is characterized as a directed semantic transition between two concepts: from a concept that is mentioned in declarative statements (*e.g.* `Symptom`), to another concept that indicates the user's information need (*e.g.* `Medicine`). As shown in Table I, each sentence adopts a unique expression but they all share the same intention where users mention symptom concepts and look for related medications. Moreover, when users try to express more sophisticated information in a single piece of sentence, the semantic transitions also become complicated over multiple concepts, as shown in Table II.

Text	Structured Intent
My three-year-old child is sick with a temperature of 100 degrees she can't keep anything down including liquids. What kind of medicine should I give my child, and how much?	Symptom → *Medicine* → *Instruction*
Do I have insomnia if I have trouble staying asleep? Any medication is recommended to help me fall asleep easier?	*Disease* ← *Symptom* → *Medicine*

TABLE II: Complicated sentences with structured intents.

In this work, we introduce a novel neural network architecture that bring structures to detect complicated user intents in the user-generated text corpus. We observe an appealing property that information-seeking text corpus exhibits a strong coupling between concept mentions and semantic transitions between concepts. The proposed model is trained to automatically discover concept mentions and infer semantic transitions from the unstructured text corpus, in contrast

to relying on fixed dictionaries for word-concept mapping (Chiang et al., 2012; Godbole et al., 2010; Zhang et al., 2016) or using pre-defined parsing rules (De and Kopparapu, 2010) and templates (Spink et al., 2004) in prior works. A customized graph-based mutual transfer loss function is designed to impose explicit constraints to reduce the conflicts between extracting concept mentions and inferring semantic transitions. We show that by taking the correlations among concept mentions and semantic transitions into considerations, the proposed model is able to accurately detect complicated user intents from the text corpus.

Experiments are conducted on the text corpus collected from an online question-answering discussion forum. We contrast the performance of the proposed model with other alternatives by an 8% relative improvement in micro-AUC and an 23% relative reduction in coverage loss.

2.2 Preliminaries

We now formally define the terminologies and describe the structured intent detection problem for natural language understanding. Also, we provide observations to show appealing coupling properties of concept mentions and semantic transitions in the text corpus (Cai et al., 2017), which motivates a graph-based formulation for structured intent.

2.2.1 Terminologies

Definition 1 (**Concept**)**.** Let a concept c be a group or class of objects and/or abstract ideas that share similar fundamental characteristics in a certain domain. $C = \{c_1, c_2, ..., c_M\}$ is list of a full spectrum of M concepts in a specific domain. For example, the medical domain contains concepts of diseases, symptoms, medicine and so on. Users can mention concepts in a text corpus by specific object names as explicit mentions (“Tylenol”, “Ibuprofen” or “xxx

caplet/capsule/drop/syrup"), or as implicit mentions by abstract ideas ("remedy" or "which medication/medicine/drug").

Definition 2 (**Semantic Transition**)**.** Let a semantic transition $t_{i \to j}$ defines a transition of a user information-seeking intention from a concept c_i to a concept c_j. A semantic transition $t_{i \to j}$ exists in the text corpus when two concepts c_i, $c_j \in C$ are mentioned (either explicitly or implicitly) with a semantic transition between them. For example, a concept transition $t_{Symptom \to Medicine}$ in the healthcare domain usually starts with patients describing their symptoms and asking for related information about medications that help them alleviate their symptoms.

T contains the full spectrum of N semantic transitions in a certain domain, which can be indexed as flat labels $T = \{t_1, t_2, ..., t_N\}$ for simplicity instead of $\{t_{i \to j}\}$. Those two index notations are used interchangeably in this work. Multiple semantic transitions can co-exist in a single piece of text corpus and the direction of a semantic transition does not necessarily follow the order of concept occurrence in the text corpus. Multiple semantic transitions may follow certain structures such as a chain-like path, like $Symptom \to Medicine \to Instruction$.

An intent is defined as a semantic transition between two concepts. Formally, we have:

Definition 3 (**Intent**)**.** Considering a basic case, where each text corpus consists of some declarative sentences followed by questions. For each information-seeking text corpus Q, the resulting intent is denoted as a tuple pair $\langle s, n \rangle$: where s is the concept being mentioned in declarative sentences as the information known to the user, while n indicates a concept being mentioned in questions indicating user's information needs.

When a user tries to express complicated information needs, a single piece of text corpus may embody multiple intents. We observe that multiple semantic transitions in a single piece of text are often correlated with each other, coupled with some shared concept mentions.

To effectively model complicated semantic transitions among multiple concept mentions, we first define an intent graph that bring structures to concept mentions and semantic transitions.

Definition 4 (**Intent Graph**). Let $G = \langle C, T \rangle$ be an intent graph where each node represents a concept $c_m \in C$ and each directed edge $t_{i,j} \in T$ be a semantic transition from node c_i to c_j. An intent graph G is a graph representation that indicates all possible concept mentions and semantic transitions in a certain domain. Note that the domain-specific intent graph can be obtained from domain experts or constructed as a concept-level graph from large text corpora using existing techniques (Hasegawa et al., 2004; Yan et al., 2009; Zhang et al., 2016).

Definition 5 (**Structured Intent**). Let a Structured Intent $\hat{G}_Q = \langle \hat{C}_Q, \hat{T}_Q \rangle$ be a sub-graph of $G = \langle C, T \rangle$, indicating concepts $\hat{C}_Q \subseteq C$ mentioned by the text corpus Q and semantic transitions $\hat{T}_Q \subseteq T$ inferred from Q.

2.2.2 Problem Statement

Definition 6 (**Structured Intent Detection for Natural Language Understanding**). Given 1) an information-seeking text corpus Q which consists of K elements $\{q_1, q_2, ..., q_K\}$, where each element is a word or a phrase and 2) an intent graph $G = \langle C, T \rangle$, where C denotes all possible concept mentions and T indicates all possible semantic transitions, the Structured Intent Detection problem tries to effectively infer the Structured Intent $\hat{G}_Q = \langle \hat{C}_Q, \hat{T}_Q \rangle$ as a

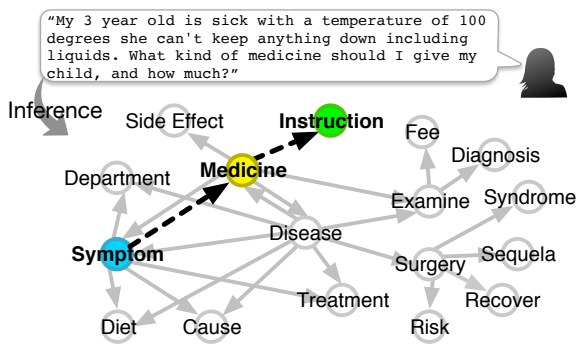


Figure 2: Structured Intent Detection for Natural Language Understanding.

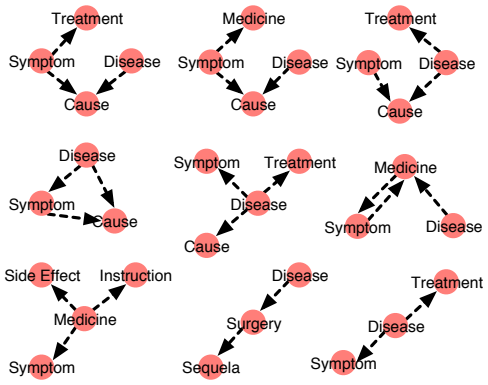


Figure 3: Frequent structured intents.

sub-graph of the intent graph G, where $\hat{C}_Q \subseteq C$ and $\hat{T}_Q \subseteq T$. Figure 2 illustrates this idea, where $\hat{C}_Q$ are shown as colored nodes and $\hat{T}_Q$ are shown as black arrows with dashed lines.

2.2.3 Observations

We sample 10,000 pieces of text corpus from an online medical question answering discussion forum and label them with structured intents. We end up having 17 unique types of concepts and 23 unique types of semantic transitions (Details in Section 2.4.1).

We show the top-9 frequent structured intents being annotated, as shown in Figure 3. By characterizing complicated user intentions with a graph-based formulation, the Structured Intent Detection task maps each text corpus with diverse expressions into a graph structure that indicates users' information needs in a clear and structural way.

More importantly, we found that the Structured Intents rarely have disconnected components, from a perspective of the graph theory. This not only shows that users tend to express multiple semantic transitions in a single piece of information-seeking text corpus but also in-

dicates that multiple semantic transitions in a single piece of text corpus are expressed and developed together, coupled with some shared concept mentions. In summary, the connectivity patterns of frequent Structured Intents further imply that by taking advantages of the semantic structure, the correlations between nodes and edges in the Structured Intent can be jointly inferred with a synergistic effect.

2.3 Proposed Approach

In this section, a neural network structure is introduced to provide an end-to-end solution to the Structured Intent Detection problem where the input is a text corpus and the output is a Structured Intent inferred from the corpus. The model utilizes word representations to deal with the lexical diversities. Also, part-of-speech embedding of each word is used to further capture the syntax information. Recurrent neural networks are adopted to model the sequential information from distributed representations of word and POS tag sequences in each query simultaneously. In the graph-based co-inference procedure, concept mentions and semantic transitions are inferred collectively. A Concept Extractor is proposed to utilize the joint outputs of two RNNs to encode each element into a concept vector. Especially, the Concept Extractor is able to learn an attention weight as a confidence score that indicates the contribution of each element to each concept. While for inferring semantic transitions, a transition encoder learns to summarize the semantics and construct a transition vector, from which we infer a probability distribution over all possible semantic transitions. The loss of the neural network structure not only incorporates prediction errors between the inferred semantic transitions and the true semantic transitions but also exploits a mutual transfer loss indicating the conflicts between

the extracted concepts and the semantic transitions. A Structured Intent is presented with the inferred concepts and semantic transitions, by collectively minimizing a graph-based mutual transfer loss based on the intent graph. Figure 4 gives an overview of the proposed method.

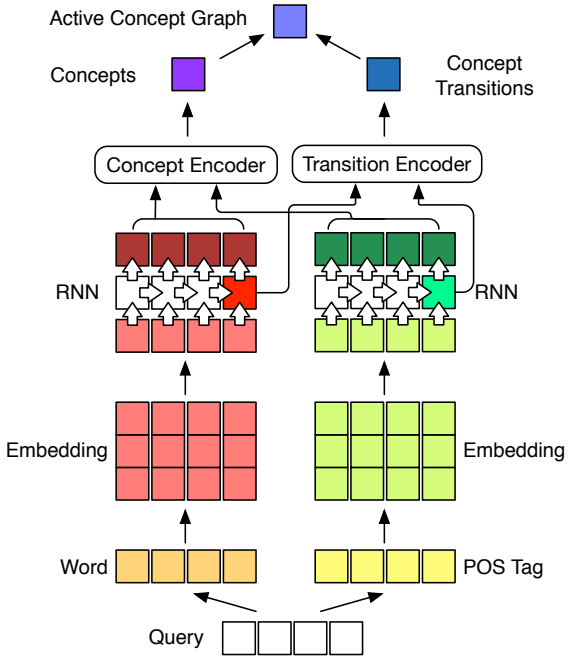


Figure 4: The proposed neural network architecture.

2.3.1 Lexical-Syntax Representations

Unlike traditional methods which ignore the sequential information of the input text corpus and treat it as a bag-of-words (BoW), in this work a text corpus Q is considered as a sequence of elements $\{q_1, q_2, ..., q_K\}$, where each element q_k can be a word or a phrase. K is the length of

the text corpus, which varies in different corpora. For each element q_k in a text corpus Q, we utilize both word representations indicating the lexical information, as well as its corresponding Part-of-Speech (POS) tag as the syntax information.

Part-of-speech (POS) tags bring useful syntax information about general word categories (such as noun, verb, adjective, etc.), which is helpful in dealing with ambiguous words and diversified expressions. For example, `fever` can be either a noun or a verb. The word `fever` with a POS tag "noun" is defined as a disease that causes an increase in body temperature and the fever with a POS tag "verb" can be considered as someone in a fever, as a symptom. In this work, an existing POS tagger[1] is utilized to give general POS tags to each element in the text corpus. The lexical-syntax joint representation consists of words along with POS tags are shown to be effective in modeling both lexical (words) and syntax (POS tags) from the natural language text corpus in various tasks (Legrand and Collobert, 2015; Zhang et al., 2016). In this work, each element q_k of a text corpus Q is represented by words and POS tags as a tuple:

$$q_k = (w_k, p_k) \;\; s.t. \; w_k \in \mathbb{R}^{V_{word}}, p_k \in \mathbb{R}^{V_{pos}}, \tag{2.1}$$

where w_k is the one-hot representation of the k-th word in the corpus Q and V_{word} is the number of unique words, namely the vocabulary size. Similarly, p_k is the one-hot representation of the k-th word's POS tag in the corpus. V_{POS} is the POS vocabulary size.

---

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

2.3.2 Word Embedding

The one-hot representation suffers from the curse of dimensionality since the representation becomes extremely sparse as the vocabulary becomes large. The word embedding is used to transfer one-hot representation of each word w_k and POS tag p_k into a dense representation: $w\_embed_k \in \mathbb{R}^{D_{word}}, p\_embed_k \in \mathbb{R}^{D_{pos}}$, where V_{word} usually can be large up to millions while D_{word} is reduced to several hundreds. Note that D_{word} and D_{pos} are usually set empirically. In this work, we set $D_{word} = 100$ and $D_{pos} = 20$. The embedded representation of each w_k and p_k are learned respectively by a linear mapping via a skip-gram model (Mikolov et al., 2013b):

$$embed\_w_k = \mathbf{W}_{word}\ w_k, \quad embed\_p_k = \mathbf{W}_{pos}\ p_k, \tag{2.2}$$

where $\mathbf{W}_{word} \in \mathbb{R}^{D_{word} \times V_{word}}$ and $\mathbf{W}_{pos} \in \mathbb{R}^{D_{pos} \times V_{pos}}$ are weights.

In this work, the embedding is initialized with word vectors pre-trained from 64 million text corpus and updated with the model during training. After the word embedding, the k-th element in the text corpus q_k has a lexical-syntax representation, represented by a tuple: $e_k = (embed\_w_k, embed\_p_k)$.

2.3.3 Recurrent Neural Network

Once we obtained a representation e_k for each element q_k in a text corpus Q, the $embed\_w_k$ sequence and the $embed\_p_k$ sequences are fed into two recurrent neural networks, namely $\mathrm{RNN_W}$ and $\mathrm{RNN_P}$, to capture the sequential semantics respectively.

In general, a recurrent neural network keeps hidden states over a sequence of elements and updates the hidden state h_k by the current input x_k as well as the previous hidden state h_{k-1} where $k > 1$ by a recurrent function: $h_k = \text{RNN}(x_k, h_{k-1})$. The Gated Recurrent Unit (GRU) (Cho et al., 2014) is proposed to address the gradients decay or exploding problem (Bengio et al., 1994; Hochreiter, 1998) over long sequences in the vanilla RNN. The GRU has been attracting great attention since it overcomes the vanishing gradient in traditional RNNs and is more efficient than LSTM (Hochreiter and Schmidhuber, 1997) on certain tasks (Chung et al., 2014). The GRU is designed to learn from previous time stamps with long time lags of unknown size between important time stamps.

In this work, two separate RNN with GRU cells, namely RNN_W and RNN_P, are adopted to model the sequential information in the sequence of embedded words $embed\_w_k$ and the sequence of embedded POS tags $embed\_p_k$:

$$h\_w_k, o\_w_k = \text{RNN}_\text{W}(embed\_w_k, h\_w_{k-1}), \quad h\_p_k, o\_p_k = \text{RNN}_\text{P}(embed\_p_k, h\_p_{k-1}), \tag{2.3}$$

2.3.4 Graph-based Co-inference

In order to fully exploit the correlations of concept mentions and semantic transitions, instead of inferring concepts and semantic transitions separately, a collective inference schema is adopted. The Concept Extractor aims to select a subset of concepts $\hat{C}_Q \subseteq C$ that are mentioned in a the corpus Q. A transition encoder is introduced to infer semantic transitions $\hat{T}_Q \subseteq T$ over the Intent Graph G. The concepts $\hat{C}_Q$ and transitions $\hat{T}_Q$ are inferred collectively,

by minimizing a mutual transfer loss which indicates the conflicts within the inferred Structured Intent $\hat{G}_Q =< \hat{C}_Q, \hat{T}_Q >$.

Concept Extractor The Concept Extractor encodes all concept mentions from a sequence of output states of an RNN to a single concept vector. Since some words in the text corpus may contribute more to a concept, the Concept Extractor itself learns to assign a confidence score to each output state. Let o_k be the k-th output vector of an RNN, while in this work we concatenate the output vectors of RNN_W and RNN_P:

$$o_k = [o\_w_k, o\_p_k], o\_w_k \in \mathbb{R}^{1\times D_{o_w}}, o\_p_k \in \mathbb{R}^{1\times D_{o_p}}, \tag{2.4}$$

where D_{o_w} and D_{o_p} are the output dimensions of output vectors in RNN_W and RNN_P. The Concept Extractor assigns a score s_k for each o_k indicating the degree of confidence, parameterized by θ:

$$s_k = \frac{CE\left(o_k;\theta\right)}{\sum\limits_{k'\in K} CE\left(o_{k'};\theta\right)} \quad \text{s.t. } \sum\nolimits_k s_k = 1, \forall s_k \in [0,1]. \tag{2.5}$$

The s_k scores for all elements in a text corpus are normalized to sum up to one. We implement the $CE(\cdot)$ function as a single layer neural network with a non-linear activation function ReLU. Thus θ consists of $\{\mathbf{W}_\theta \in \mathbb{R}^{(D_{o_w}+D_{o_p})\times 1}, b_\theta \in \mathbb{R}\}$. Note that although weights and biases are applied on each of the o_k, they are shared among all $o_1, o_2, ..., o_K$. Figure 5 shows the architecture of the Concept Extractor, which is used to determine confidence scores for each joint output state. This figure shows an example of a score s_1 learned from the Concept Extractor for o_1. The $o_{CE} \in \mathbb{R}^{(D_{o_w}+D_{o_p})\times 1}$ is a representation of encoded concepts from the

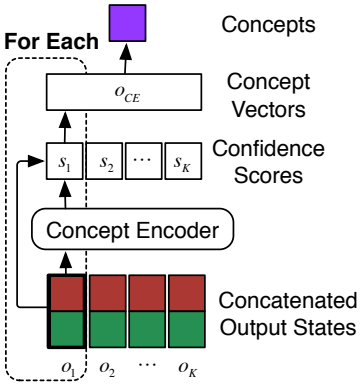


Figure 5: The Concept Extractor.

text corpus, which is calculated as the weighted sum on the output vectors as $O_{CE} = \sum_k s_k o_k$.

The probability of a concept $c_i \in C$ being mentioned in any part of the text corpus Q is defined by a softmax function over logits on all concepts, where the logit for each concept is learned by a logistic function:

$$(\hat{C}_Q)_m = P(c_m | c_m \in C; \theta) = \frac{1}{1 + e^{-W_{CE} O_{CE} + b_{CE}}}, \tag{2.6}$$

where $\mathbf{W}_{CE} \in \mathbb{R}^{1 \times (D_{o_w} + D_{o_p})}$, $b_{CE} \in \mathbb{R}$ are weights and biases for each type of concept $m \in M$. We feed logits $\hat{C}_Q \in \mathbb{R}^{1 \times M}$ to the softmax layer and get the probability distribution over all M types of concepts being mentioned in the given text corpus Q.

Transition Encoder In the field of machine translation, a novel recurrent neural network encoder-decoder has gained attention (Sutskever et al., 2014), where the encoder recurrent neural network encodes the global information spanning over the whole input sentence in its last

hidden state. Inspired by the effectiveness of the last hidden states in modeling natural language sequences in applications like dialog systems (Serban et al., 2016), we propose a transition encoder which leverages the last hidden state of the neural network for both RNN_W, RNN_P to make inferences on semantic transitions, where the transition vector o_{TE} is constructed by $O_{TE} = [h\_w_K, h\_p_K]$, where K is the length of the query. The logit of a transition $t_n \in T$ is quantified as:

$$(\hat{T}_Q)_n = P(t_n | t_n \in T; \phi) = \frac{1}{1 + e^{-W_{TE} O_{TE} + b_{TE}}}, \tag{2.7}$$

where $\phi = \{\mathbf{W}_{TE} \in \mathbb{R}^{1 \times (D_{ow} + D_{op})}, b_{TE} \in \mathbb{R}\}$ parameterizes weights and biases for each type of transition. Similarly, $\hat{T}_Q \in \mathbb{R}^{1 \times N}$ is fed to the softmax layer and we get the inferred probability distribution over all N semantic transitions.

2.3.5 Mutual Transfer Loss

The idea of mutual transfer loss is to characterize the loss caused by transferring the inferred semantic transitions to their corresponding concept mentions, and the other way around. Since for each semantic transition $t_{i \to j} \in T$, two concepts c_i and c_j are involved. If a semantic transition $t_{i \to j}$ is inferred with a high probability while its corresponding concepts c_i, c_j have low probabilities, then that indicates conflicts in the final Structured Intent. The mutual transfer loss is proposed in the co-inference procedure to minimize the conflicts between the inferred concepts and semantic transitions so that the resulting Structured Intent can be more reasonable.

The graph-based formulation for the Structured Intent gives an appealing property that transitions and their proximate concepts can be clearly characterized by a transfer matrix

$A \in \mathbb{R}^{M \times N}$ over the Intent Graph $G = \langle C, T \rangle$. Each entry $a_{mn} = 1$ if and only if the concept c_m involves in at least one end of a semantic transition $t_{m\rightarrow\cdot}$ or $t_{\cdot\rightarrow m}$. The mutual transfer loss is defined on $\hat{C}_Q, \hat{T}_Q, \mathcal{T}_Q$ as:

$$\mathcal{L}_{MTL}(\hat{C}_Q, \hat{T}_Q, \mathcal{T}_Q) = H(\mathcal{T}_Q, \hat{T}_Q) + E(\hat{C}_Q, \hat{T}_Q), \tag{2.8}$$

where $\mathcal{T}_Q$ is a ground truth one-hot indicator for semantic transitions given the corpus Q. $\hat{C}_Q$ and $\hat{T}_Q$ are extracted concepts and inferred semantic transitions with the proposed method. $H(\cdot, \cdot)$ calculates the cross entropy (Tsoumakas et al., 2009). $E(\hat{C}_Q, \hat{T}_Q)$ is an energy-based function on inferred transitions $\hat{T}_Q$ and extracted concepts $\hat{C}_Q$. Each combination of $\hat{C}_Q$ and $\hat{T}_Q$ corresponds with an energy value, the lower energy level a combination of $\hat{C}_Q$ and $\hat{T}_Q$ has indicates less conflicts among the inferred concepts and transitions. In this work, an energy-based function for $E(\hat{C}_Q, \hat{T}_Q)$ is proposed as:

$$E(\hat{C}_Q, \hat{T}_Q) = \mathcal{L}_R(\hat{C}_Q, \hat{T}_Q A^T) + \mathcal{L}_R(\hat{T}_Q, \hat{C}_Q A), \tag{2.9}$$

where $\mathcal{L}_R$ is implemented by a ranking loss function (Murphy, 2012) that penalizes cases where the inferred concepts/transitions after transformation by matrix A have high probabilities but order below the ranking of the originally inferred concepts/transitions in the same corpus. $\mathcal{L}_R$ has a general form:

$$\mathcal{L}_R(\hat{X}, \hat{Y}) = \frac{1}{\left|\hat{X}\right| (L - \left|\hat{X}\right|)} \left| \{(p, q) : \hat{Y}_p < \hat{Y}_q, \hat{X}_p \geq \hat{X}_q\} \right|, \tag{2.10}$$

where $\hat{X} \in \mathbb{R}^{1\times L}$ is the originally inferred labels and $\hat{Y} \in \mathbb{R}^{1\times L}$ is the inferred labels from the transformation with A. $|\cdot|$ denotes the number of ground truth labels being assigned. L is the label size, where we have M for concepts and N for semantic transitions.

2.4 Evaluation

2.4.1 Dataset

We collect text corpora from an online medical question answering discussion forum[1], on which user posted their healthcare related questions and medical professionals give online suggestions or advice. The obtained corpora are in Chinese. Due to the fact that sentences in Chinese are not naturally split by spaces, word segmentation is performed using a Chinese word segmentation package[2].

After preprocessing and annotation, we obtain 10,000 pieces of text corpora. We end up having 17 unique types of concepts and 23 unique types of semantic transitions, among which 11,531 unique words and 60 unique POS tags are observed. The POS tagging uses ICTCLAS annotation (Zhang et al., 2003). The average length of the text corpus is 13.8, with a standard variation of ±6.1. The average number of concepts in the labeled corpus is 3.6020±0.8. The average number of semantic transitions is 2.4723±0.7.

Word embeddings are pre-trained using a skip-gram model (Mikolov et al., 2013b) on 64 million unlabeled text corpus separately. Context window size is set to 8 and we specify a

---

[1]http://club.xywy.com

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

minimum occurrence count of 5. The vocabulary contains 100-dimension vectors on 382,216 words. Words not presented in the set of pre-trained words are initialized as random vectors. All word vectors will be updated during training.

2.4.2 Experiment Settings

To show the advantages of the proposed method in addressing the concept transition inference problem, we compare it with the following baseline models.

- **LR**: a logistic regression model applied with POS tagging features and word representations.
- **NNID-JM** (Zhang et al., 2016): the neural network intent detection model with joint modeling. Both words and POS tags are used to characterize the words in the corpus. Domain-specific POS tags, such as "noun_medicine", are used in NNID-JM instead of "noun" for word "Tylenol". The NNID-JM doesn't explicitly exploit label correlations on the output level.
- **CI**: the Concept Inference model which only infers mention of concepts from the corpus with the Concept Extractor. $H(\mathcal{C}_Q, \hat{C}_Q)$ is used as the loss function for the CI task.
- **CTI**: the Concept Transition Inference model without co-inference. Only semantic transitions are inferred from the corpus. The last output states of two RNNs are concatenated to predict the semantic transitions. $H(\mathcal{T}_Q, \hat{T}_Q)$ is used as the loss function.
- **coCTI**: the Concept Transition Inference model with co-inference. $H(\mathcal{T}_Q, \hat{T}_Q)+H(\mathcal{C}_Q, \hat{C}_Q)$ is used as the loss function. This variation can be seen as a multi-task learning model for

extracting concepts and inferring semantic transitions, where two tasks share the lower-level neural network structure for word representation.

- **coCTI-MTL**: the proposed model with co-inference and a mutual transfer loss $\mathcal{L}_{MTL}$, where the CI task and CTI task not only share the neural network structure, but also adopt the mutual transfer loss.

Evaluation Metrics: Each directed edge in the Intent Graph is considered as an individual label and we evaluate inferred Structured Intent as a multi-class, multi-label classification problem. *Receiver operating characteristic* (ROC), the *micro/macro-average area under the curve* (micro-AUC, macro-AUC), *coverage error* and *label ranking average precision* (LRAP) are used to evaluate the effectiveness of the proposed model in inferring Structured Intents from the text corpus. The ROC and AUCs focus on the quality of prediction, while the coverage error and LRAP are introduced to evaluate the completeness/ranking of the prediction. ROC is the curve created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Micro-AUC computes the averaged area under the ROC curve over all the labels. Coverage error computes the average number of labels that we need to have in the final prediction in order to predict all true labels. LRAP score favors better rank to labels that are associated to each sample and is usually used in multi-label ranking problems.

Experiment Settings: The embeddings for word and POS tagging have a dimension of 100 and 20, respectively. The hidden layer and the output layer of the GRU unit have a dimension of 100. For training the proposed neural network structure, 70% of the labeled data are used for training and 10% samples are served as the validation set to tune for the best parameter set.

The remaining data are used for testing. Cross-validation is used and we combine test data in each fold to report the test performance. The optimization is performed in a mini-batch fashion with a batch size of 32. The Adam Optimizer (Kingma and Ba, 2014) is applied to train the neural network and the initial learning rate is set to 10^{-4}. Weight variables are initialized with the Xavier initializer (Glorot and Bengio, 2010) and bias variables are initialized as zeros.

2.4.3 Experiment Results

Figure 6 shows the effectiveness of the proposed model by micro-AUC and ROC curves. Generally, neural network based models (NNID-JM, CTI, coCTI, coCTI-MTL) outperform traditional logistic regression model (LR) consistently. For NNID-JM, in order to make a fair

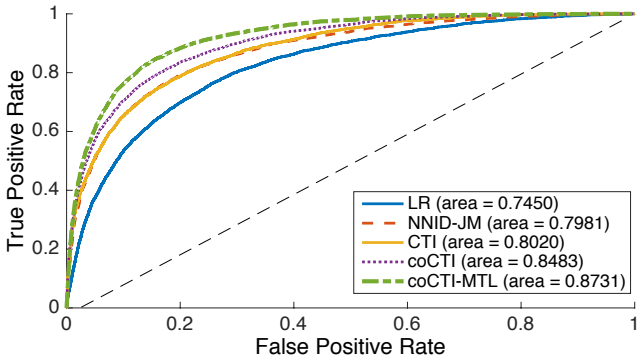


Figure 6: Micro-AUC scores and ROC curves.

comparison, domain specific POS tags (such as noun_disease, noun_medicine, noun_symptom) are maintained as an external knowledge base. Those POS tags are used by the POS tag-

ger in NNID-JM as its default setting. When compared with NNID-JM, the proposed CTI model achieves similar performance on micro-AUC, while it doesn't rely on any other external knowledge like domain-specific POS tags in NNID-JM.

From Figure 6 we can further observe that CTI-MTL achieves the best performance (0.8731 in micro-AUC) among all the comparison methods in correctly inferring semantic transitions from the text corpus. The CTI-MTL model has a nearly 2.5% improvement on micro-AUC when compared with coCTI and a nearly 7.5% improvement with CTI. This demonstrates that the mutual transfer loss which penalizes conflicts between the extracted concept mentions and inferred semantic transitions can indeed improve the structured intent detection performance.

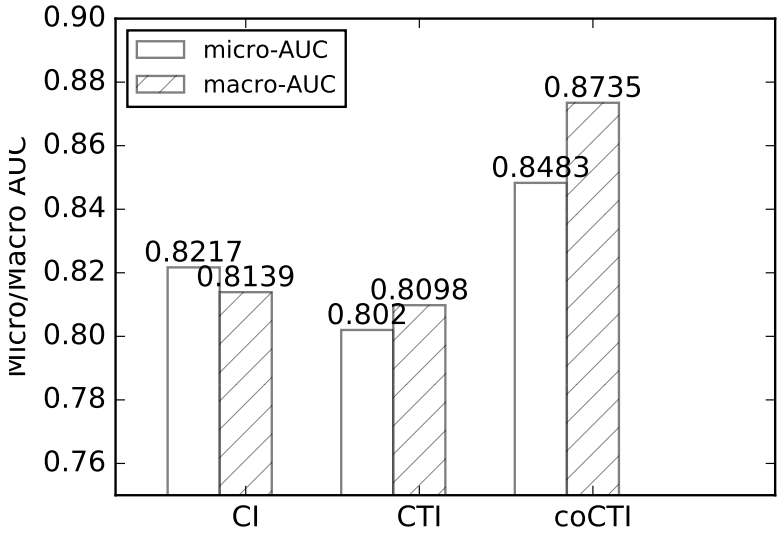


Figure 7: Micro/Macro-AUC scores on collective inference and separate inference.

Concept Transition	LR	NNID-JM	CTI	coCTI	coCTI-MTL
Symptom→Diet	0.6544 (5)	0.7755 (4)	0.7669 (3)	0.7959 (2)	0.8495 (1)
Symptom→Medicine	0.7022 (5)	0.7893 (4)	0.8242 (3)	0.8571 (2)	0.8624 (1)
Symptom→Cause	0.7600 (5)	0.8549 (4)	0.8786 (3)	0.8911 (1)	0.8880 (2)
Disease→Diet	0.7818 (5)	0.8670 (4)	0.8681 (3)	0.9059 (2)	0.9458 (1)
Disease→Treatment	0.7181 (5)	0.7787 (3)	0.7482 (4)	0.8456 (2)	0.8836 (1)
Disease→Examine	0.6397 (5)	0.6707 (4)	0.7838 (3)	0.8221 (2)	0.8480 (1)
Disease→Medicine	0.7623 (5)	0.8726 (4)	0.8749 (3)	0.8873 (2)	0.9015 (1)
Surgery→Recover	0.8117 (5)	0.9126 (3)	0.9012 (4)	0.9239 (2)	0.9396 (1)
Surgery→Sequela	0.7385 (5)	0.8031 (4)	0.8214 (3)	0.8417 (2)	0.8972 (1)
Surgery→Syndrome	0.7896 (5)	0.7994 (4)	0.8634 (2)	0.8619 (3)	0.9172 (1)
Surgery→Risk	0.6613 (5)	0.8063 (4)	0.8688 (3)	0.8715 (2)	0.9099 (1)
Medicine→Symptom	0.6861 (5)	0.8275 (3)	0.7553 (4)	0.8294 (2)	0.8598 (1)
Medicine→Side Effect	0.6652 (5)	0.8162 (3)	0.7771 (4)	0.8135 (2)	0.8814 (1)
Medicine→Disease	0.6806 (4)	0.6514 (5)	0.8081 (3)	0.8126 (2)	0.8678 (1)
Medicine→Instruction	0.7090 (5)	0.7761 (3)	0.7603 (4)	0.8170 (2)	0.8820 (1)
Examine→Fee	0.7576 (5)	0.9049 (3)	0.8981 (4)	0.9425 (2)	0.9482 (1)
Examine→Diagnosis	0.6832 (5)	0.7956 (3)	0.7445 (4)	0.8383 (2)	0.8822 (1)
Symptom→Treatment	0.6817 (5)	0.7640 (3)	0.7313 (4)	0.8130 (2)	0.8531 (1)
Symptom→Department	0.5978 (5)	0.6460 (3)	0.6013 (4)	0.6738 (2)	0.8080 (1)
Disease→Cause	0.7306 (5)	0.8206 (4)	0.8515 (3)	0.8608 (2)	0.8634 (1)
Disease→Symptom	0.6936 (4)	0.7552 (3)	0.6845 (5)	0.7554 (2)	0.8372 (1)
Disease→Department	0.6931 (5)	0.7387 (4)	0.7431 (3)	0.7652 (2)	0.8290 (1)
Disease→Surgery	0.7801 (5)	0.8795 (4)	0.9029 (3)	0.9236 (2)	0.9380 (1)

TABLE III: Fine-grained AUC scores for all semantic transitions.

Figure 7 shows the effectiveness of the co-inference procedure by comparing the performance of CTI with coCTI. The CI infers concept mentions so we can't simply compare its performance with CTI/coCTI where semantic transitions are inferred. However, for CTI and coCTI, the improved performance on both micro-AUC and macro-AUC validates the effectiveness of inferring concepts and semantic transitions collectively than inferred separately. The coCTI model can be considered as a multi-task learning model where the lower-level text representations are learned jointly and shared between two sub-tasks.

Furthermore, the fine-grained AUC scores on all semantic transitions without micro/macro-averaging are shown in Table III. A general observation we can draw from the results is that

the coCTI-MTL model is able to outperform other baselines in almost all types of semantic transitions.

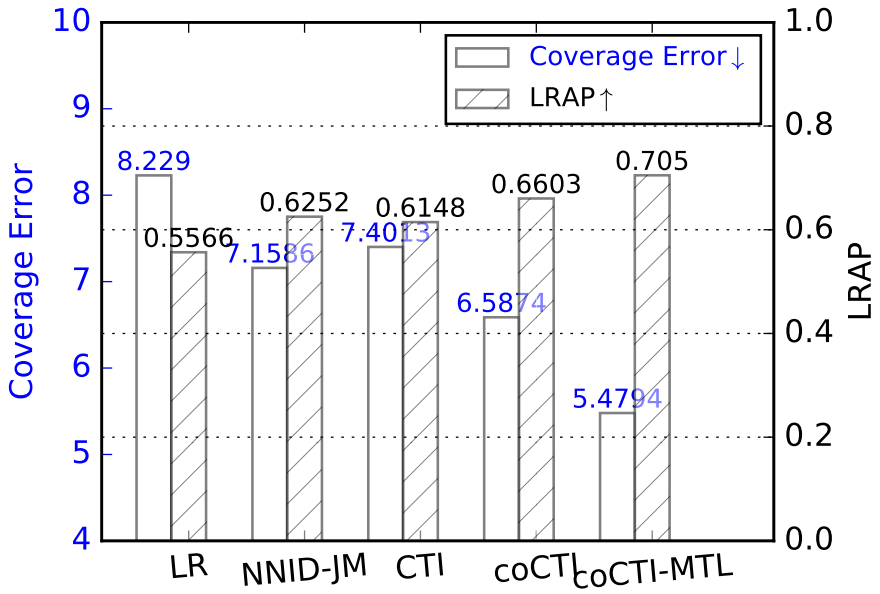


Figure 8: Coverage Loss and Label Ranking Average Precision (LRAP).

Figure 8 shows the coverage loss and LRAP over proposed methods and other baselines, where the coCTO-MTL model is able to achieve the lowest coverage error and the highest label ranking average precision score.

2.5 Related Works

Query Analysis As the number of people posting questions or searching for information online is growing rapidly, researchers have been focusing on new problems and applications based on

the user-generated text corpus, such as queries or search queries. (Limsopatham et al., 2013) analyzes the conceptual relationship in online web documents records for a better web search. (Stanton et al., 2014) focuses on the circumlocution problem in diagnostic questions in the healthcare domain, where users are not able to express their ideas effectively. (Zhang et al., 2016) tries to model user intentions as a classification task for text queries. (Liu et al., 2015) proposes a technique to detect whether users express their own experiences in the generated text corpus. (Li et al., 2016) introduces a knowledge discovery model for the online question-answering corpus. In (Liu et al., 2016), authors introduce a neural network model to understand users questions and try to generate answers appropriately. Being able to infer concept transitions from noisy, user-generated questions may further facilitate various applications in domains like healthcare, such as healthcare question-answering, medical dialog systems or recommendation. For example, once we extracted the concept transition $Symptom \rightarrow Medicine$ from a question `Any medication is recommended to help me fall asleep easier?`, we may follow up by recommending the user to the nearest pharmacy for further medical consultations on corresponding OTC medicines on Insomnia.

Text Classification Recently, lots of neural network models are developed for classifying natural language text corpus into different categories (Kalchbrenner et al., 2014; Lai et al., 2015). Those methods achieve decent performance on general text classification tasks. The proposed Structured Intent Detection task can be seen as a multi-class multi-label classification problem. Unlike traditional text classification tasks like news classification where the existence of some topic words may easily dominate the label for a news title, users tend to mention

multiple concepts in a single piece of text corpus. It is crucial to accurately infer semantic transitions among those concepts, besides extracting concept mentions only.

Also, the aforementioned methods consider the textual information only. With a graph-based formation in this work, our model seamlessly incorporates an existing Intent Graph for effective intent detection on complicated information-seeking text corpora. More specifically, we propose to predict concept mentions as nodes and semantic transitions as links collectively, while most existing works have been focusing on predicting links among concrete entities, e.g. among users in social networks (Liben-Nowell and Kleinberg, 2007), or predicting links among entities on a knowledge graph (Nickel et al., 2016; Bordes et al., 2013).

CHAPTER 3

Structure-aware Natural Language Modeling

Part of this chapter was published as "Joint Slot Filling and Intent Detection via Capsule Neural Networks", in ACL'19 (Zhang et al., 2019): `https://arxiv.org/abs/1812.09471`.

3.1 Introduction

With the ever-increasing accuracy in speech recognition and complexity in user-generated utterances, it becomes a critical issue for mobile phones or smart speaker devices to understand the natural language in order to give informative responses. Slot filling and intent detection play important roles in Natural Language Understanding systems. For example, given an utterance from the user, the slot filling annotates the utterance on a word-level, indicating the slot type mentioned by a certain word such as the slot `artist` mentioned by the word `Sungmin`, while the intent detection works on the utterance-level to give categorical intent label(s) to the whole utterance. Figure 9 illustrates this idea.

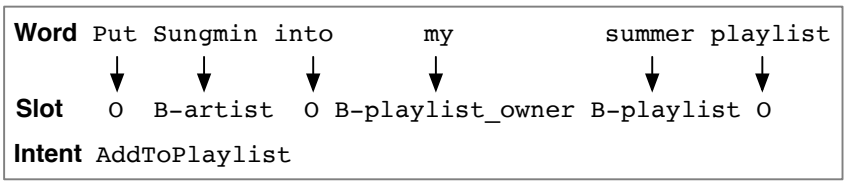


Figure 9: An example of an utterance with BIO format annotation.

To deal with diversely expressed utterances without additional feature engineering, deep neural network based user intent detection models (Hu et al., 2009; Xu and Sarikaya, 2013; Zhang et al., 2016; Liu and Lane, 2016; Zhang et al., 2017; Chen et al., 2016; Xia et al., 2018) are proposed to classify user intents given their utterances in the natural language.

Currently, the slot filling is usually treated as a sequential labeling task. A neural network such as a recurrent neural network (RNN) or a convolution neural network (CNN) is used to learn context-aware word representations, along with sequence tagging methods such as conditional random field (CRF) (Lafferty et al., 2001) that infer the slot type for each word in the utterance.

Word-level slot filling and utterance-level intent detection can be conducted simultaneously to achieve a synergistic effect. The recognized slots, which possess word-level signals, may give clues to the utterance-level intent of an utterance. For example, with a word `Sungmin` being recognized as a slot `artist`, the utterance is more likely to have an intent of `AddToPlayList` than other intents such as `GetWeather` or `BookRestaurant`.

Some existing works learn to fill slots while detecting the intent of the utterance (Xu and Sarikaya, 2013; Hakkani-Tür et al., 2016; Liu and Lane, 2016; Goo et al., 2018): a convolution layer or a recurrent layer is adopted to sequentially label word with their slot types: the last hidden state of the recurrent neural network, or an attention-weighted sum of all convolution outputs are used to train an utterance-level classification module for intent detection. Such approaches achieve decent performances but do not explicitly consider the task taxonomy on two tasks, nor the hierarchical relationship between words, slots, and intents: intents are sequentially

summarized from the word sequence. As the sequence becomes longer, it is risky to simply rely on the gate function of RNN to compress all contexts in a single vector (Cheng et al., 2016).

In this work, we make the very first attempt to bridge the gap between word-level slot modeling and the utterance-level intent modeling via a hierarchical capsule neural network structure (Hinton et al., 2011; Sabour et al., 2017) that is aware of the task taxonomy. A capsule houses a vector representation of a group of neurons. The capsule model learns a hierarchy of feature detectors via a routing-by-agreement mechanism: capsules for detecting low-level features send their outputs to high-level capsules only when there is a strong agreement of their predictions to high-level capsules.

The aforementioned properties of capsule models are appealing for natural language understanding from a hierarchical perspective: words such as `Sungmin` are routed to concept-level slots such as `artist`, by learning how each word matches the slot representation. Concept-level slot features such as `artist`, `playlist owner`, and `playlist` collectively contribute to an utterance-level intent `AddToPlaylist`. The dynamic routing-by-agreement assigns a larger weight from a lower-level capsule to a higher-level when the low-level feature is more predictive to one high-level feature, than other high-level features. Figure 10 illustrates this idea. The model does slot filling by learning to assign each word in the WordCaps to the most appropriate slot in SlotCaps via dynamic routing. The weights learned via dynamic routing indicate how strong each word in WordCaps belongs to a certain slot type in SlotCaps. The dynamic routing also learns slot representations using WordCaps and the learned weight. The learned slot representations in SlotCaps are further aggregated to predict the utterance-level intent of the

utterance. Once the intent label of the utterance is determined, a novel re-routing process is proposed to help improve word-level slot filling by the inferred utterance-level intent label. The solid lines indicate the dynamic-routing process and dash lines indicate the re-routing process.

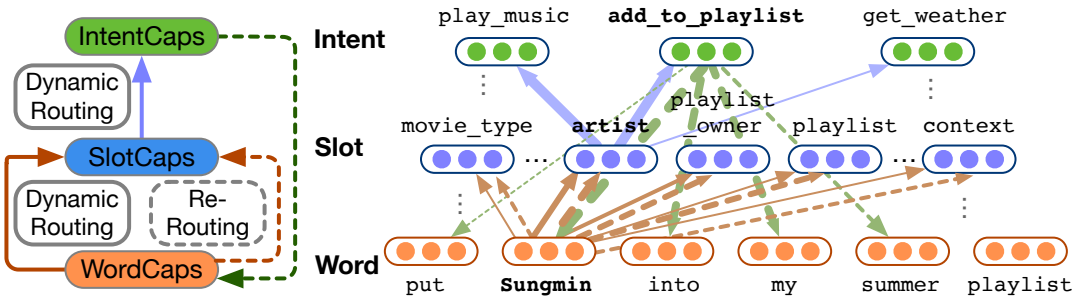


Figure 10: Illustration of the proposed CAPSULE-NLM model.

The inferred utterance-level intent is also helpful in refining the slot filling result. For example, once an `AddToPlaylist` intent representation is learned in IntentCaps, the slot filling may capitalize on the inferred intent representation and recognize slots that are otherwise neglected previously. To achieve this, we propose a re-routing schema for capsule neural networks, which allows high-level features to be actively engaged in the dynamic routing between WordCaps and SlotCaps, which improves the slot filling performance.

To summarize, the contributions of this work are as follows:

- Encapsulating the hierarchical relationship among word, slot, and intent in an utterance by a hierarchical capsule neural network structure.
- Proposing a dynamic routing schema with re-routing that achieves synergistic effects for joint slot filling and intent detection.
- Showing the effectiveness of our model on two real-world datasets, and comparing with existing models as well as commercial natural language understanding services.

3.2 Proposed Approach

We propose to model the hierarchical relationship among each word, the slot it belongs to, and the intent label of the whole utterance by a hierarchical capsule neural network structure called Capsule-NLM. The proposed architecture consists of three types of capsules: 1) WordCaps that learn context-aware word representations, 2) SlotCaps that categorize words by their slot types via dynamic routing, and construct a representation for each type of slot by aggregating words that belong to the slot, 3) IntentCaps determine the intent label of the utterance based on the slot representation as well as the utterance contexts. Once the intent label has been determined by IntentCaps, the inferred utterance-level intent helps re-recognizing slots from the utterance by a re-routing schema.

3.2.1 WordCaps

Given an input utterance $x = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T)$ of T words, where each word is initially represented by a vector of dimension D_W. Here we simply trained word represenations from scratch. Various neural network structures can be used to learn context-aware word represen-

tations. For example, a recurrent neural network such as a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) can be applied to learn representations of each word in the utterance:

$$\vec{\mathbf{h}}_t = \text{LSTM}_{fw}(\mathbf{w}_t, \vec{\mathbf{h}}_{t-1}), \quad \overleftarrow{\mathbf{h}}_t = \text{LSTM}_{bw}(\mathbf{w}_t, \overleftarrow{\mathbf{h}}_{t+1}). \tag{3.1}$$

For each word $\mathbf{w}_t$, we concatenate each forward hidden state $\vec{\mathbf{h}}_t$ obtained from the forward LSTM_{fw} with a backward hidden state $\overleftarrow{\mathbf{h}}_t$ from LSTM_{bw} to obtain a hidden state $\mathbf{h}_t$. The whole hidden state matrix can be defined as $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T) \in \mathbb{R}^{T \times 2D_H}$, where D_H is the number of hidden units in each LSTM. In this work, the parameters of WordCaps are trained with the whole model, while sophisticated pre-trained models such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) may also be integrated.

3.2.2 SlotCaps

Traditionally, the learned hidden state $\mathbf{h}_t$ for each word $\mathbf{w}_t$ is used as the logit to predict its slot tag. When $\mathbf{H}$ for all words in the utterance is learned, sequential tagging methods like the linear-chain CRF models the tag dependencies by assigning a transition score for each transition pattern between adjacent tags to ensure the best tag sequence of the utterance from all possible tag sequences.

Instead of doing slot filling via sequential labeling which does not directly consider the dependencies among words, the SlotCaps learn to recognize slots via dynamic routing. The routing-by-agreement explicitly models the hierarchical relationship between capsules to address the task taxonomy explicitly. For example, the routing-by-agreement mechanism send a low-

level feature, e.g. a word representation in WordCaps, to high-level capsules, e.g. SlotCaps, only when the word representation has a strong agreement with a slot representation.

The agreement value on a word may vary when being recognized as different slots. For example, the word `three` may be recognized as a `party_size_number` slot or a `time` slot. The SlotCaps first convert the word representation obtained in WordCaps with respect to each slot type. We denote $\mathbf{p}_{k|t}$ as the resulting prediction vector of the t-th word when being recognized as the k-th slot:

$$\mathbf{p}_{k|t} = \sigma(\mathbf{W}_k \mathbf{h}_t^T + \mathbf{b}_k), \tag{3.2}$$

where $k \in \{1, 2, ..., K\}$ denotes the slot type and $t \in \{1, 2, ..., T\}$. σ is the activation function such as $tanh$. $\mathbf{W}_k \in \mathbb{R}^{D_P \times 2D_H}$ and $\mathbf{b}_k \in \mathbb{R}^{D_P \times 1}$ are the weight and bias matrix for the k-th capsule in SlotCaps, and D_P is the dimension of the prediction vector.

Slot Filling by Dynamic Routing-by-agreement We propose to determine the slot type for each word by dynamically route prediction vectors of each word from WordCaps to SlotCaps. The dynamic routing-by-agreement learns an agreement value c_{kt} that determines how likely the t-th word agrees to be routed to the k-th slot capsule. c_{kt} is calculated by the dynamic routing-by-agreement algorithm (Sabour et al., 2017), which is briefly recalled in Algorithm 1.

The above algorithm determines the agreement value c_{kt} between WordCaps and SlotCaps while learning the slot representations $\mathbf{v}_k$ in an unsupervised, iterative fashion. $\mathbf{c}_t$ is a vector that consists of all c_{kt} where $k \in K$. b_{kt} is the logit (initialized as zero) representing the log prior probability that the t-th word in WordCaps agrees to be routed to the k-th slot capsule in SlotCaps (Line 2). During each iteration (Line 3), each slot representation $\mathbf{v}_k$ is calculated by

Algorithm 1 Dynamic routing-by-agreement

1: **procedure** DYNAMIC_ROUTING($\mathbf{p}_{k|t}$, *iter*)
2: for each WordCaps t and SlotCaps k: $b_{kt} \leftarrow 0$.
3: **for** *iter* iterations **do**
4: for all WordCaps t: $\mathbf{c}_t \leftarrow \text{softmax}(\mathbf{b}_t)$
5: for all SlotCaps k: $\mathbf{s}_k \leftarrow \Sigma_r c_{kt}\mathbf{p}_{k|t}$
6: for all SlotCaps k: $\mathbf{v}_k = \text{squash}(\mathbf{s}_k)$
7: for all WordCaps t and SlotCaps k: $b_{kt} \leftarrow \text{b}_{kt} + \mathbf{p}_{k|t} \cdot \mathbf{v}_k$
8: **end for**
9: Return $\mathbf{v}_k$
10: **end procedure**

aggregating all the prediction vectors for that slot type $\{\mathbf{p}_{k|t}|t{\in}T\}$, weighted by the agreement values c_{kt} obtained from b_{kt} (Line 5-6):

$$\mathbf{s}_k = \sum_{t}^{T} c_{kt}\mathbf{p}_{k|t}, \tag{3.3}$$

$$\mathbf{v}_k = \text{squash}(\mathbf{s}_k) = \frac{\|\mathbf{s}_k\|^2}{1 + \|\mathbf{s}_k\|^2}\frac{\mathbf{s}_k}{\|\mathbf{s}_k\|}, \tag{3.4}$$

where a squashing function squash$(\cdot)$ is applied on the weighted sum $\mathbf{s}_k$ to get $\mathbf{v}_k$ for each slot type. Once we updated the slot representation $\mathbf{v}_k$ in the current iteration, the logit b_{kt} becomes larger when the dot product $\mathbf{p}_{k|t} \cdot \mathbf{v}_k$ is large. That is, when a prediction vector $\mathbf{p}_{k|t}$ is more similar to a slot representation $\mathbf{v}_k$, the dot product is larger, indicating that it is more likely to route this word to the k-th slot type (Line 7). An updated, larger b_{kt} will lead to a larger agreement value c_{kt} between the t-th word and the k-th slot in the next iteration. On the other

hand, it assigns low c_{kt} when there is inconsistency between $p_{k|t}$ and $\mathbf{v}_k$. The agreement values learned via the unsupervised, iterative algorithm ensures the outputs of the WordCaps get sent to appropriate subsequent SlotCaps after $iter_{\text{slot}}$ iterations.

Cross Entropy Loss for Slot Filling

For the t-th word in an utterance, its slot type is determined as follows:

$$\hat{y}_t = \arg\max_{k \in K}(c_{kt}). \tag{3.5}$$

The slot filling loss is defined over the utterance as the following cross-entropy function:

$$\mathcal{L}_{slot} = -\sum_{t}\sum_{k} y_t^k \log(\hat{y}_t^k), \tag{3.6}$$

where y_t^k indicates the ground truth slot type for the t-th word. $y_t^k = 1$ when the t-th word belongs to the k-th slot type.

3.2.3 IntentCaps

The IntentCaps take the output $\mathbf{v}_k$ for each slot $k \in \{1, 2, ..., K\}$ in SlotCaps as the input, and determine the utterance-level intent of the whole utterance. The IntentCaps also convert each slot representation in SlotCaps with respect to the intent type:

$$\mathbf{q}_{l|k} = \sigma(\mathbf{W}_l \mathbf{v}_k^T + b_l), \tag{3.7}$$

where $l \in \{1, 2, ..., L\}$ and L is the number of intents. $\mathbf{W}_l \in \mathbb{R}^{D_L \times D_P}$ and $\mathbf{b}_l \in \mathbb{R}^{D_L \times 1}$ are the weight and bias matrix for the l-th capsule in IntentCaps.

IntentCaps adopt the same dynamic routing-by-agreement algorithm, where:

$$\mathbf{u}_l = \textsc{Dynamic\_Routing}(\mathbf{q}_{l|k}, iter_{\text{intent}}). \tag{3.8}$$

Max-margin Loss for Intent Detection

Based on the capsule theory, the orientation of the activation vector $\mathbf{u}_l$ represents intent properties while its length indicates the activation probability. The loss function considers a max-margin loss on each labeled utterance:

$$\mathcal{L}_{intent} = \sum_{l=1}^{L} \{[\![z = z_l]\!] \cdot \max(0, m^+ - \|\mathbf{u}_l\|)^2 + \lambda\, [\![z \neq z_l]\!] \cdot \max(0, \|\mathbf{u}_l\| - m^-)^2\}, \tag{3.9}$$

where $\|\mathbf{u}_l\|$ is the norm of $\mathbf{u}_l$ and $[\![]\!]$ is an indicator function, z is the ground truth intent label for the utterance x. λ is the weighting coefficient, and m^+ and m^- are margins.

The intent of the utterance can be easily determined by choosing the activation vector with the largest norm $\hat{z} = \underset{l \in \{1,2,...,L\}}{\arg\max} \|\mathbf{u}_l\|$.

3.2.4 Re-Routing

The IntentCaps not only determine the intent of the utterance by the length of the activation vector, but also learn discriminative intent representations of the utterance by the orientations of the activation vectors. Previously, the dynamic routing-by-agreement shows how low-level features such as slots help construct high-level ideas such as intents. While the high-level fea-

tures also work as a guide that helps learn low-level features. For example, the `AddToPlaylist` intent activation vector in IntentCaps also helps strength the existing slots such as `artist_name` during slot filling on the words `Sungmin` in SlotCaps.

Thus we propose a re-routing schema for SlotCaps where the dynamic routing-by-agreement is realized by the following equation that replaces the Line 7 in Algorithm 1:

$$\mathrm{b}_{kt} \leftarrow \mathrm{b}_{kt} + \mathbf{p}_{k|t} \cdot \mathbf{v}_k + \alpha \cdot \mathbf{p}_{k|t}^T \mathbf{W}_{RR} \hat{\mathbf{u}}_{\hat{z}}^T, \tag{3.10}$$

where $\hat{\mathbf{u}}_{\hat{z}}$ is the intent activation vector with the largest norm. $\mathbf{W}_{RR} \in \mathbb{R}^{D_P \times D_L}$ is a bi-linear weight matrix, and α as the coefficient. The routing information for each word is updated toward the direction where the prediction vector not only coincides with representative slots, but also towards the most-likely intent of the utterance. As a result, the re-routing makes SlotCaps obtain updated routing information as well as updated slot representations.

3.3 Evaluation

To demonstrate the effectiveness of our proposed models, we compare the proposed model Capsule-NLM with existing alternatives, as well as commercial natural language understanding services.

3.3.1 Datasets

For each task, we evaluate our proposed models by applying it on two real-word datasets: SNIPS Natural Language Understanding benchmark[1] (SNIPS-NLU) and the Airline Travel

[1] `https://github.com/snipsco/nlu-benchmark/`

Information Systems (ATIS) dataset (Tur et al., 2010). The statistical information on two datasets are shown in Table IV.

	SNIPS-NLU	**ATIS**
Vocab Size	11,241	722
Average Sentence Length	9.05	11.28
#Intents	7	21
#Slots	72	120
#Training Samples	13,084	4,478
#Validation Samples	700	500
#Test Samples	700	893

TABLE IV: Dataset statistics.

SNIPS-NLU contains natural language corpus collected in a crowdsourced fashion to benchmark the performance of voice assistants. ATIS is a widely used dataset in spoken language understanding, where audio recordings of people making flight reservations are collected.

3.3.2 Experiment Settings

Baselines We compare the proposed capsule-based model Capsule-NLM with other alternatives:

- **CNN TriCRF** (Xu and Sarikaya, 2013) introduces a Convolution Neural Network (CNN) based sequential labeling model for slot filling. The hidden states for each word are summed up to predict the utterance intent. We adopt the performance with lexical features.

- **Joint Seq.** (Hakkani-Tür et al., 2016) adopts a Recurrent Neural Network (RNN) for slot filling and the last hidden state of the RNN is used to predict the utterance intent.
- **Attention BiRNN** (Liu and Lane, 2016) further introduces a RNN based encoder-decoder model for joint slot filling and intent detection. An attention weighted sum of all encoded hidden states is used to predict the utterance intent.
- **Slot-gated Full Atten.** (Goo et al., 2018) utilizes a slot-gated mechanism as a special gate function in Long Short-term Memory Network (LSTM) to improve slot filling by the learned intent context vector. The intent context vector is used for intent detection.
- **DR-AGG** (Gong et al., 2018) aggregates word-level information for text classification via dynamic routing. The high-level capsules after routing are concatenated, followed by a multi-layer perceptron layer that predicts the utterance label. We used this capsule-based text classification model for intent detection only.
- **IntentCapsNet** (Xia et al., 2018) adopts a multi-head self-attention to extract intermediate semantic features from the utterances, and uses dynamic routing to aggregate semantic features into intent representations for intent detection. We use this capsule-based model for intent detection only.

We also compare our proposed model CAPSULE-NLM with existing commercial natural language understanding services, including api.ai (Now called DialogFlow)[1], Waston Assistant[2], Luis[3], wit.ai[4], snips.ai[5], recast.ai[6], and Amazon Lex[7].

Implementation Details The hyperparameters used for experiments are shown in Table V.

DATASET	D_W	D_H	D_P	D_L	$iter_{\text{slot}}$	$iter_{\text{intent}}$
SNIPS-NLU	1024	512	512	128	2	2
ATIS	1024	512	512	256	3	3

TABLE V: Hyperparameter settings.

We use the validation data to choose hyperparameters. For both datasets, we randomly initialize word embeddings using Xavier initializer and let them train with the model. In the

[1] https://dialogflow.com/

[2] https://www.ibm.com/cloud/watson-assistant/

[3] https://www.luis.ai/

[4] https://wit.ai/

[5] https://snips.ai/

[6] https://recast.ai/

[7] https://aws.amazon.com/lex/

MODEL	SNIPS-NLU			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
CNN TriCRF (Xu and Sarikaya, 2013)	-	-	-	0.944	-	-
Joint Seq. (Hakkani-Tür et al., 2016)	0.873	0.969	0.732	0.942	0.926	0.807
Attention BiRNN (Liu and Lane, 2016)	0.878	0.967	0.741	0.942	0.911	0.789
Slot-Gated Full Atten. (Goo et al., 2018)	0.888	0.970	0.755	0.948	0.936	0.822
DR-AGG (Gong et al., 2018)	-	0.966	-	-	0.914	-
IntentCapsNet (Xia et al., 2018)	-	0.974	-	-	0.948	-
CAPSULE-NLM	**0.918**	0.973	**0.809**	**0.952**	**0.950**	**0.834**
CAPSULE-NLM w/o Intent Detection	0.902	-	-	0.948	-	-
CAPSULE-NLM w/o Joint Training	0.902	**0.977**	0.804	0.948	0.847	0.743

TABLE VI: Slot filling and intention detection results.

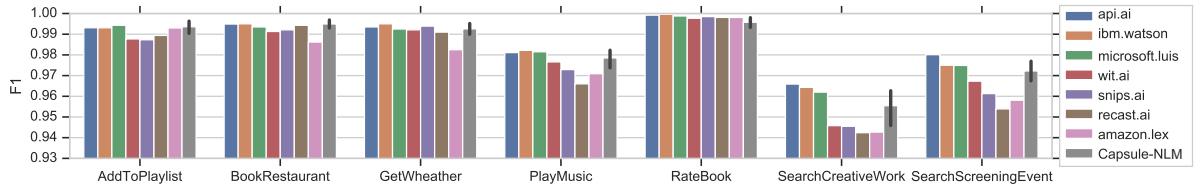


Figure 11: Benchmarking with existing NLU services.

loss function, the down-weighting coefficient λ is 0.5, margins m^+ and m^- are set to 0.8 and 0.2 for all the existing intents. α is set as 0.1. RMSProp optimizer (Tieleman and Hinton, 2012) is used to minimize the loss. To alleviate over-fitting, we add the dropout to the LSTM layer with a dropout rate of 0.2.

3.3.3 Experiment Results

Quantitative Evaluation The intent detection results on two datasets are reported in Table VI, where the proposed capsule-based model performs consistently better than current learning schemes for joint slot filling and intent detection, as well as capsule-based neural net-

work models that only focuses on intent detection. These results demonstrate the novelty of the proposed capsule-based model Capsule-NLM in jointly modeling the hierarchical relationships among words, slots and intents via the dynamic routing between capsules.

Also, we benchmark the intent detection performance of the proposed model with existing natural language understanding services[1] in Figure 11. Since the original data split is not available, we report the results with stratified 5-fold cross validation. From Figure 11 we can see that the proposed model Capsule-NLM is highly competitive with off-the-shelf systems that are available to use. Note that, our model archieves the performance without using pre-trained word representations: the word embeddings are simply trained from scratch.

Ablation Study To investigate the effectiveness of Capsule-NLM in joint slot filling and intent detection, we also report ablation test results in Table VI. "w/o Intent Detection" is the model without intent detection: only a dynamic routing is performed between WordCaps and SlotCaps for the slot filling task, where we minimize $\mathcal{L}_{slot}$ during training; "w/o Joint Training" adopts a two-stage training where the model is first trained for slot filling by minimizing $\mathcal{L}_{slot}$, and then use the fixed slot representations to train for the intent detection task which minimizes $\mathcal{L}_{intent}$. From the lower part of Table VI we can see that by using a capsule-based hierarchical modeling between words and slots, the model Capsule-NLM w/o Intent Detection is already able to outperform current alternatives on slot filling that adopt a sequential labeling

[1]https://www.slideshare.net/KonstantinSavenkov/nlu-intent-detection-benchmark-by-intento-august-2017

schema. The joint training of slot filling and intent detection is able to give each subtask further improvements when the model parameters are updated jointly.

Visualizing Agreement Values between Capsule Layers Thanks to the dynamic routing-by-agreement schema, the dynamically learned agreement values between different capsule layers naturally reflect how low-level features are collectively aggregated into high-level ones for each input utterance. In this section, we harness the intepretability of the proposed capsule-based model via hierarchical modeling and provide case studies and visualizations.

Between WordCaps and SlotCaps First we study the agreement value c_{kt} between the t-th word in the WordCaps and the k-th slot capsule in SlotCaps. Figure 12 shows the distribution of all agreement values between WordCaps and SlotCaps on the test split of SNIPS-NLU dataset. Blue bars indicate the distribution of values after the first iteration and orange bars indicate the distribution after the second iteration. We observe that the dynamic routing-by-agreement is able to converge to an agreement quickly after the first iteration (shown in blue bars). It is able to assign a confident probability assignment close to 0 or 1. After the second iteration (shown in orange bars), the model is more certain about the routing decisions: probabilities are more leaning towards 0 or 1 as the model is confident about routing a word in WordCaps to its most appropriate slot in SlotCaps.

However, we do find that when unseen slot values like new object names emerge in utterances like `show me the movie operetta for the theatre organ` with an intent of `SearchCreativeWork`, the iterative dynamic routing process would be even more appealing. Figure 13 shows the agreement values learned by dynamic routing-by-agreement. A sample from the test split of

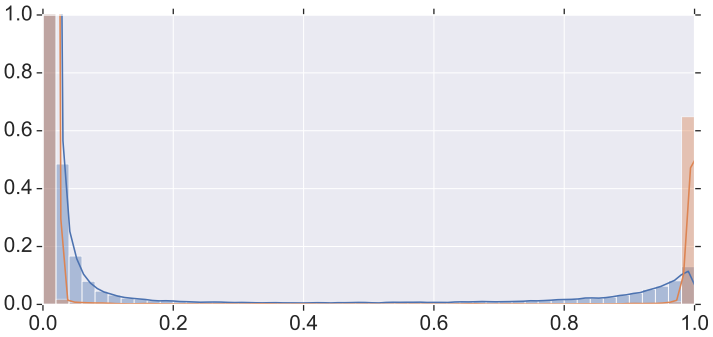


Figure 12: The distribution of agreement values between WordCaps & SlotCaps.

SNIPS-NLU dataset is shown (Left: after the fist routing iteration. Right: after the second iteration). Since the dynamic routing-by-agreement is an iterative process controlled by the variable $iter_{slot}$, we show the agreement values after the first iteration in the left part of Figure 13, and the values after the second iteration in the right part. Due to space limitations, only part of slots (7/72) are shown on the y-axis.

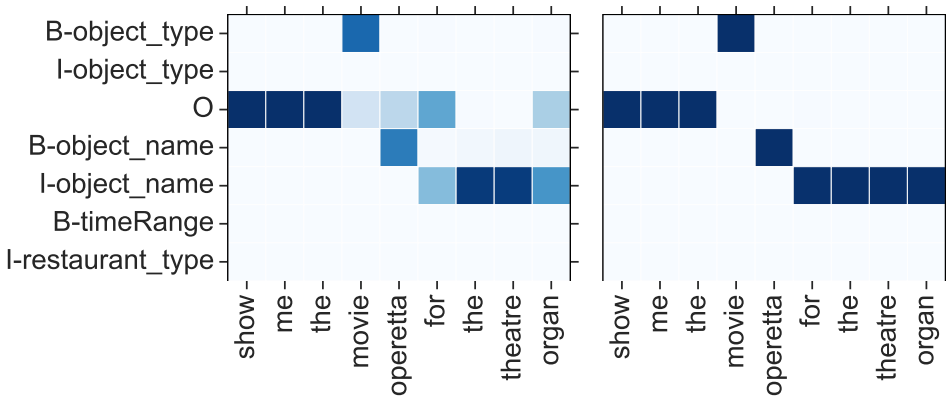


Figure 13: Agreement values between WordCaps (x-axis) and SlotCaps (y-axis).

From the left part of Figure 13, we can see that after the first iteration, the model considers the word `operetta` itself alone is likely to be an object name, probably because the following word `for` is usually a context word being annotated as `O`. Thus it tends to route word `for` to both the slot `O` and the slot `I-object_name`. However, from the right part of Figure 13 we can see that after the second iteration, the dynamic routing found an agreement and is more certain to have `operetta for the theatre organ` as a whole for the slot `B-object_name` and `I-object_name`.

Between SlotCaps and IntentCaps Similarly, we visualize the agreement values between each slot capsule in SlotCaps and each intent capsule in IntentCaps. The left part of Figure 14 shows that after the first iteration, since the model is not able to correctly recognize `operetta for the theatre organ` as a whole, only the context slot `O` (correspond to the word `show me the`) and `B-object_name` (correspond to the word `operetta`) contribute significantly to the final intent capsule. From the right part of Figure 14, we found that with the word `operetta for the theatre organ` being recognized in the lower capsule, the slots `I-object_name` and `B-object_type` contribute more to the correct intent capsule `SearchCreativeWork`, when comparing with other routing alternatives to other intent capsules.

3.4 Related Works

Intent Detection With recent developments in deep neural networks, user intent detection models (Hu et al., 2009; Xu and Sarikaya, 2013; Zhang et al., 2016; Liu and Lane, 2016; Zhang et al., 2017; Chen et al., 2016; Xia et al., 2018) are proposed to classify user intents given their diversely expressed utterances in the natural language. As a text classification task, the decent

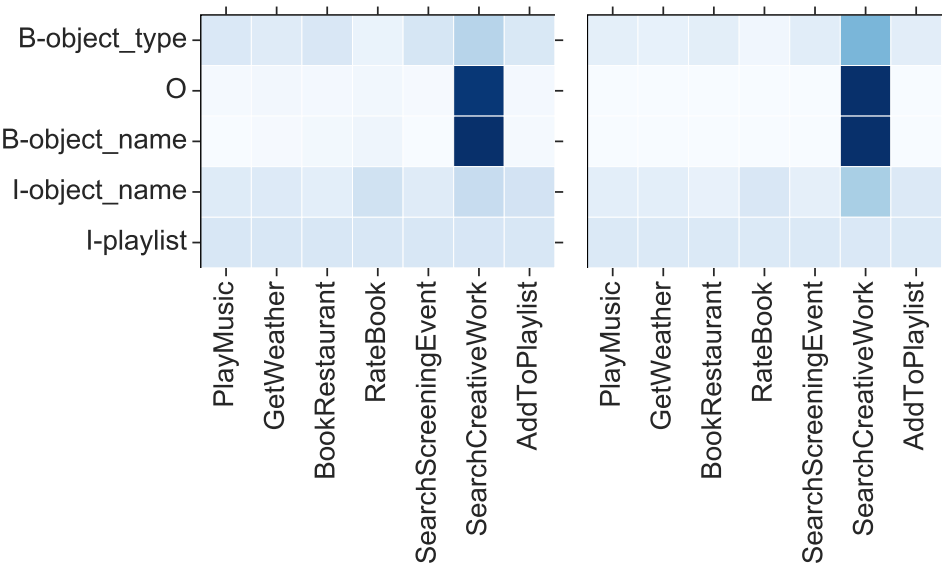


Figure 14: Agreement values between SlotCaps (y-axis) and IntentCaps (x-axis).

performance on utterance-level intent detection usually relies on hidden representations that are learned in the intermediate layers via multiple non-linear transformations.

Recently, various capsule based text classification models are proposed that aggregate word-level features for utterance-level classification via dynamic routing-by-agreement (Gong et al., 2018; Zhao et al., 2018; Xia et al., 2018). Among them, (Xia et al., 2018) adopts self-attention to extract intermediate semantic features and uses a capsule-based neural network for intent detection. However, existing works do not study word-level supervisions for the slot filling task. In this work, we explicitly model the hierarchical relationship between words and slots on the word-level, as well as intents on the utterance-level via dynamic routing-by-agreement.

Slot Filling Slot filling annotates the utterance with finer granularity: it associates certain parts of the utterance, usually named entities, with pre-defined slot tags. Currently, the slot

filling is usually treated as a sequential labeling task. A recurrent neural network such as Gated Recurrent Unit (GRU) or Long Short-term Memory Network (LSTM) is used to learn context-aware word representations, and Conditional Random Fields (CRF) are used to annotate each word based on its slot type. Recently, (Shen et al., 2018; Tan et al., 2018) introduce the self-attention mechanism for CRF-free sequential labeling.

Joint Modeling via Sequence Labeling To overcome the error propagation in the word-level slot filling task and the utterance-level intent detection task in a pipeline, joint models are proposed to solve two tasks simultaneously in a unified framework. (Xu and Sarikaya, 2013) propose a Convolution Neural Network (CNN) based sequential labeling model for slot filling. The hidden states corresponding to each word are summed up in a classification module to predict the utterance intent. A Conditional Random Field module ensures the best slot tag sequence of the utterance from all possible tag sequences. (Hakkani-Tür et al., 2016) adopt a Recurrent Neural Network (RNN) for slot filling and the last hidden state of the RNN is used to predict the utterance intent. (Liu and Lane, 2016) further introduce an RNN based encoder-decoder model for joint slot filling and intent detection. An attention weighted sum of all encoded hidden states is used to predict the utterance intent. Some specific mechanisms are designed for RNNs to explicitly encode the slot from the utterance. For example, (Goo et al., 2018) utilize a slot-gated mechanism as a special gate function in Long Short-term Memory Network (LSTM) to improve slot filling by the learned intent context vector. However, as the sequence becomes longer, it is risky to simply rely on the gate function to sequentially

summarize and compress all slots and context information in a single vector (Cheng et al., 2016).

In this paper, we harness the capsule neural network to learn a hierarchy of feature detectors and explicitly model the hierarchical relationships among word-level slots and utterance-level intent. Also, instead of doing sequence labeling for slot filling, we use a dynamic routing-by-agreement schema between capsule layers to route each word in the utterance to its most appropriate slot type. And we further route slot representations, which are learned dynamically from words, to the most appropriate intent capsule for intent detection.

CHAPTER 4

Generative Structured Knowledge Expansion

This chapter was previously published as "On the Generative Discovery of Structured Medical Knowledge", in KDD'18 (Zhang et al., 2018a). DOI: `https://doi.org/10.1145/3219819.3220010`.

4.1 Introduction

Knowledge Graphs such as WordNet (Miller, 1995), Yago (Fabian et al., 2007) and Freebase (Bollacker et al., 2008) have been playing an essential role in many applications, such as knowledge inference, question answering, relation extraction, and so on. A large-scale of structured knowledge is embodied in Knowledge Graphs, in the form of triplets (head entity, tail entity and the relationship, denoted as $h \xrightarrow{r} t$). For example, the $Disease \xrightarrow{Cause} Symptom$ relationship indicates a "Cause" relationship from a disease entity (*e.g.* `synovitis`) to a symptom entity (*e.g.*`joint pain`) which is caused by this disease. Various linguistic expressions are usually observed among different triplets. For example, `nose plugged`, `blocked nose` and `sinus congestion` are symptom entities that share the same meaning but expressed very differently. The expression diversity is also widely observed for triplets of the same relation: a relationship may also be instantiated by entity pairs in varying granularities or different relationship strength. For instance, Disease $\xrightarrow{Cause}$ Symptom relationship may include coarse-grained entity pairs like <`rhinitis`, `nose plugged`>, while <`acute rhinitis`, `nose plugged`>, <`chronic`

`rhinitis`, `nose plugged`> are considered as fine-grained entity pairs. As for the relationship strength, <`cold`, `fatigue`> has greater relationship strength than <`cold`, `ear infections`> as cold rarely cause serious complications such as ear infections. It is straightforward for human beings yet still challenging for a machine to understand the commonalities between different triplets.

Since most knowledge graphs were built either collaboratively or (partly) automatically (Ji et al., 2015), they are far from complete (Socher et al., 2013). The knowledge graph completion task aims at predicting relationships between entities based on existing triplets in a knowledge graph. Many works have focused on extending existing knowledge graphs using well-trained classifiers to predict whether or not there is a relationship between two existing or new entities (Socher et al., 2013; Bordes et al., 2013; Komninos and Manandhar, 2017; Trouillon et al., 2017; He et al., 2018). Existing models such as for relation extraction (Agichtein and Gravano, 2000; Baeza-Yates and Tiberi, 2007; Jiang et al., 2017; Liu et al., 2017; Mintz et al., 2009; Sahay et al., 2008; Wang et al., 2015a) or knowledge graph completion (Socher et al., 2013; Komninos and Manandhar, 2017; Trouillon et al., 2017; He et al., 2018; Gardner and Mitchell, 2015; Lin et al., 2016; Wang et al., 2015b; Zeng et al., 2014) adopt a discriminative setting. Although achieving decent performance in identifying the correctness of candidate triplets, their performances rely on well-prepared annotated triplets as the training data, as well as high-quality candidate triplets for testing. Relation extraction methods aim to examine if a semantic relationship exists between two entities in the given context. And they also require a substantial collection of contexts over a full spectrum of relationships we would like to work on.

However, they can be vulnerable to the "garbage-in, garbage-out" situation: the meaningful relational triplets for a specific relationship cannot be identified when no high-quality relational triplets having that relationship are among the candidate relational triplets. The choice of candidates may involve additional human annotation, which is tedious and labor-intensive. In both tasks mentioned above (Knowledge Graph Completion and Relation Extraction), the lacking preparation of external resource or additional human annotation is fatal to the successful discovery of structured knowledge (Ma et al., 2019). Therefore, it is crucial for us to discover structured knowledge without substantial data requirement.

To reduce human annotation efforts for effective structured knowledge discovery, in this chapter we propose a novel research problem called Generative Structured Knowledge Expansion, which aims at understanding each relationship between entities solely from the existing triplets via their diverse expressions. With the help of rich semantic information embodied in entity representations learned from a massive text corpus, we aim to discover meaningful and novel triplets of a specific relationship in a generative fashion, without sophisticated feature engineering and substantial data requirement such as large-scale text corpora as contexts, or further data preparation.

We introduce a generative perspective to increase the scale of high-quality structured knowledge harnessing the massiveness of the unannotated text corpus. The proposed model explores the generative modeling capacity for entity pairs and their relationships while incorporating deep learning for hands-free feature engineering. It is able to generate meaningful triplets that are not yet observed, which expand the scale of existing structured knowledge.

Specifically, the model takes the triplets as the input. It encodes each triplet r into a latent space conditioned on the relationship type. Based on pre-trained entity representations from a massive text corpus, the encoding process further addresses relationship-enhanced entity representations, entity interactions, and expressive latent variables. The latent variables are decoded to reconstruct both the head and tail entity. Once trained, the generator samples directly from the learned latent variables and decodes them into novel triplets that expand the scale of structured knowledge with minimized additional human annotations. The performance of the proposed method is evaluated on real-world structured knowledge data in the medical domain both quantitatively and qualitatively.

4.2 Preliminaries

In this section, we briefly review preliminaries that relate to the proposed model.

Autoencoder (AE) The traditional autoencoder (Bengio and others, 2009) is a multi-layer non-recurrent neural network architecture which has been widely used for unsupervised representation learning. When given an input data x, the autoencoder starts with an encoder net where the input is mapped into a low-dimensional latent variable $z = encoder\_net(x)$ through one or more layers of non-linear transformations, followed by a decoder net where the resulting latent variable z is mapped to an output data $x' = decoder\_net(z)$ which has the same number of units as the input data x, via one or more non-linear hidden layers. The objective of the AE is to minimize the data reconstruction loss:

$$\mathcal{L}_{AE}(x) = \|x - x'\|^2 = \|x - decoder\_net(encoder\_net(x))\|^2, \tag{4.1}$$

and the resulting latent variable z is the low-dimensional latent feature learned from the data x in a totally unsupervised fashion.

Variational Autoencoder (VAE) The concept of automatic encoding and decoding makes AE suitable for generative models. Unlike the traditional autoencoder (Bengio and others, 2009) where the hidden variable z has unspecified distributions, the variational autoencoder (VAE) (Kingma and Welling, 2014) roots in Bayesian inference and inherits the architecture of AE to encode the Bayes automatically for an expressive generation. VAE assumes that the input data x can be encoded into a set of latent variables z with certain distributions, such as multivariate Gaussian distributions. The resulting Gaussian latent variables z are generated by the generative distribution $P_\theta(z)$ and x' is generated with a Bayesian model by a conditional distribution on z: $P_\theta(x'|z)$. VAE infers the latent distribution $P(z)$ using $P_\theta(z|x)$. $P_\theta(z|x)$ can be considered as some mapping from x to z, which is inferred by variational inference as one of the popular Bayesian inference methods. In VAE, $P_\theta(z|x)$ is usually inferred using a simpler distribution $Q_\phi(z|x)$ such as a Gaussian distribution. The objective of VAE is to optimize its variational lower bound:

$$\mathcal{L}_{VAE}(x, y; \theta, \phi) = -KL\left[Q_\phi\left(z|x\right)||P_\theta\left(z|x\right)\right] + \log\left(P_\theta\left(x\right)\right), \tag{4.2}$$

where the first term uses the KL-divergence to minimize the difference between the simple distribution $Q_\phi(z|x)$ and its true distribution $P_\theta(z|x)$, while the second term maximizes the $log\left(P_\theta(x)\right)$.

Conditional Variational Autoencoder (CVAE) Although the VAE can generate data that belongs to different types, the latent variable z is only modeled by x in $P_\theta(z|x)$ without knowing the type of it. Thus it cannot generate an output x' that belongs to a particular type y. The conditional variational autoencoder (CVAE) (Sohn et al., 2015) is an extension to VAE that generates x' with conditions. CVAE models both the data x and latent variables z. However, both x and z are conditioned on a class label y:

$$\mathcal{L}_{CVAE}(x, y; \theta, \phi) = -KL\left[Q_\phi\left(z|x, y\right)||P_\theta\left(z|x\right)\right] + \log\left(P_\theta\left(x|y\right)\right). \tag{4.3}$$

In this way, the real latent variable is distributed under $P_\theta(z|y)$ instead of $P_\theta(z)$. With such appealing formulation, we can have a separate $P_\theta(z|y)$ for each class y.

4.3 Proposed Approach

In this section, we introduce the Conditional Relationship Variational Autoencoder (CR-VAE) model for the Generative Structured Knowledge Expansion problem. The proposed model consists of three modules: encoder, decoder, and generator. The encoder module takes entity pairs and their relationship indicator as the input, trained to enhance entity representations and encode the diversely expressed entity pairs for each relationship to a latent space as Q_ϕ. The decoder is jointly trained to reconstruct the entity pairs as P_θ. The generator model shares the same structure with the decoder. However, instead of reconstructing the relational entity pair given in the input, it directly samples from the learned latent variable distribution to generate meaningful relational entity pairs for a particular relationship. Figure 15 gives an overview of

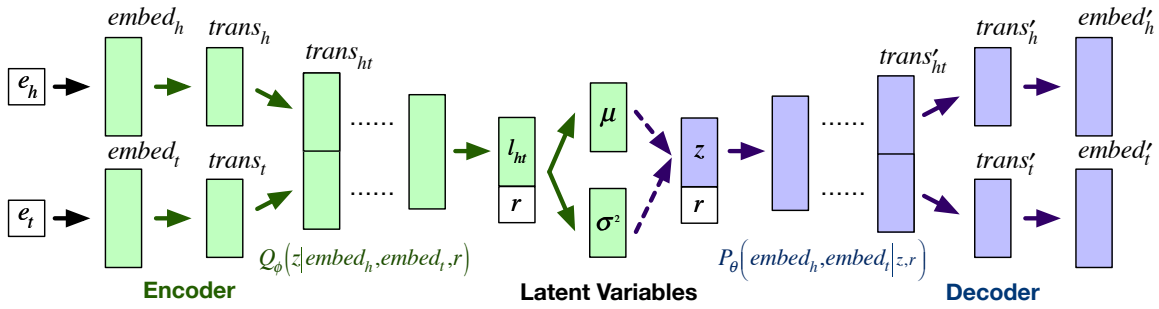


Figure 15: An overview of the proposed model CRVAE during training.

the proposed model, where the encoder module is show in green color and the decoder module is show in blue. Model inputs are in white color.

The model takes a tuple $<e_h, e_t>$ and a relationship indicator r as the input, where e_h and e_t are head and tail entity of a relationship r. For example, e_h ="*synovitis*" and e_t="*joint pain*", while the corresponding r is an indicator for *Disease* $\xrightarrow{Cause}$ *Symptom*.

To effectively represent entities, pre-trained word embeddings that embody rich semantic information can be obtained as initial entity representations for e_h and e_t. For simplicity, we adopt 200-dimensional word embeddings pre-trained using Skip-gram (Mikolov et al., 2013a). After a table lookup on the pre-trained word vector matrix $W_{embed} \in \mathbb{R}^{V \times D_E}$ where V is the vocabulary size (usually tens of thousands) and D_E is the dimension of the initial entity representation (usually tens or hundreds), $embed_h \in \mathbb{R}^{1 \times D_E}$ and $embed_t \in \mathbb{R}^{1 \times D_E}$ are derived as the initial embedding of entities.

4.3.1 Encoder

With the initial entity representation $embed_h$ and $embed_t$ and their relationship indicator r, the encoder first translates and then maps entity pairs to a latent space as $Q_\phi(z|embed_h, embed_t, r)$.

Translating for Relationship-enhancing The initial embedding obtained from word embedding reflects semantic and categorical information. However, it is not specifically designed to model the relationship between entities.

To get entity representations that address relationship information, the encoder learns to translate each entity from its initial embedding space to a relationship-enhanced embedding space that distills relational commonalities. For example, a non-linear transformation can be used: $translate(x) = f(x \cdot W_{trans} + b_{trans})$ where f can be an non-linear activation function such as the Exponential Linear Unit (ELU) (Clevert et al., 2015). $W_{trans} \in \mathbb{R}^{D_E \times D_R}$ is the weight variable and $b_{trans} \in \mathbb{R}^{1 \times D_R}$ is the bias where D_R is the dimension for relationship-enhanced embeddings.

$$trans_h = translate(embed\_h), \quad trans_t = translate(embed\_t) \tag{4.4}$$

are obtained as relationship-enhanced embeddings for e_h and e_t.

Mapping to Latent Variables The relationship-enhanced entity representation $trans_h$ and $trans_t$ are concatenated

$$trans_{ht} = [trans_h, trans_t] \tag{4.5}$$

and mapped to the latent space by multiple fully connected layers. For example, we can obtain a variable l_{ht} that addresses the relationship information, as well as entity interactions from two medical entities, by applying three consecutive non-linear fully connected layers on $trans_{ht}$.

As a variational inference model, we assume a simple Gaussian distribution of $Q_\phi(z|embed_h, embed_t, r)$ for the entity pairs $<e_h, e_t>$ with a relationship r. Therefore, for each entity pair $<e_h, e_t>$ and a relationship indicator r, a mean vector μ and a variance vector σ^2 can be learned as latent variables to model $Q_\phi(z|embed_h, embed_t, r)$:

$$\mu = [l_{ht}, r] \cdot W_\mu + b_\mu, \quad \sigma^2 = [l_{ht}, r] \cdot W_\sigma + b_\sigma, \tag{4.6}$$

where a one-hot indicator $r \in \mathbb{R}^{1\times|R|}$ is used for the relationship r and $|R|$ is the number of all relationships. $W_\mu, W_\sigma \in \mathbb{R}^{\left(D_{l_{ht}}+|R|\right)\times D_L}$ are weight terms and $b_\mu, b_\sigma \in \mathbb{R}^{1\times D_L}$ are bias terms. D_L is the dimension for latent variables and $D_{l_{ht}}$ is the dimension for l_{ht}. To stabilize the training, we model the variation vector σ^2 by its log form $\log \sigma^2$ (to be explained in Equation 4.12).

4.3.2 Decoder

Once we obtain latent variables μ, σ^2 for an input tuple $<e_h, e_t>$ which has the relationship r, the decoder uses latent variables and the relationship indicator r to reconstruct the relational medical entity pair. The decoder implements the $P_\theta(embed_h, embed_t|z, r)$.

Given μ, σ^2, it is intuitive to sample the latent value z from the distribution $N(\mu, \sigma^2)$ directly. However, such operator is not differentiable thus optimization methods failed to calculate its gradient. To solve this problem, a reparameterization trick is introduced in (Kingma

and Welling, 2014) to divert the non-differentiable part out of the network. Instead of directly sampling from $N(\mu, \sigma^2)$, we sample from a standard normal distribution $\epsilon \sim N(0, \mathrm{I})$ and convert it back to z by $z = \mu + \sigma\epsilon$. In this way, sampling from ϵ does not depend on the network.

Similarly as the use of multiple non-linear fully connected layers for the mapping in the encoder, multiple non-linear fully connected layers are used for an inverse mapping in the decoder. After the inverse mapping we obtain $trans'_{ht} \in \mathbb{R}^{1\times 2D_R}$. The first D_R dimensions of $trans'_{ht}$ are considered as a decoded relationship-enhanced embedding for e_h, while the last D_R dimensions are for e_t:

$$trans'_h = trans'_{ht}\left[: D_R\right], \quad trans'_t = trans'_{ht}\left[D_R :\right], \tag{4.7}$$

where $trans'_h, trans'_t \in \mathbb{R}^{1\times D_R}$. $trans'_h$ and $trans'_t$ are further inversely translated back to the initial embedding space $\mathbb{R}^{D_E}$:

$$embed'_h = f(trans'_h \cdot W_{trans\_inv} + b_{trans\_inv}), \quad embed'_t = f(trans'_t \cdot W_{trans\_inv} + b_{trans\_inv}), \tag{4.8}$$

where $embed'_h, embed'_t \in \mathbb{R}^{1\times D_E}$ are considered as reconstructed representations for $embed_h$ and $embed_t$.

4.3.3 Training

Inspired by the loss function of CVAE, the loss function of CRVAE is formulated to minimize the variational lower bound:

$$\mathcal{L}_{CRVAE}(embed_h, embed_t, r; \theta, \phi) = \\ - KL\left[Q_\phi\left(z|embed_h, embed_t, r\right)||P_\theta\left(z|embed_h, embed_t, r\right)\right] + \log\left(P_\theta\left(embed_h, embed_t|r\right)\right). \tag{4.9}$$

The first term minimizes the KL divergence loss between the unknown true distribution $P_\theta\left(z|embed_h, embed_t, r\right)$ and a simple distribution $Q_\phi\left(z|embed_h, embed_t, r\right)$. The second term models the entity pairs by $\log\left(P_\theta\left(embed_h, embed_t|r\right)\right)$. The above equation can be reformulated as:

$$\mathcal{L}_{CRVAE}(embed_h, embed_t, r; \theta, \phi) = \\ - KL\left[Q_\phi\left(z|embed_h, embed_t, r\right)||P_\theta\left(z|r\right)\right] + \mathbb{E}\left[\log\left(P_\theta\left(embed_h, embed_t|z, r\right)\right)\right], \tag{4.10}$$

where $P_\theta\left(z|r\right)$ describes the true latent distribution z given a certain relationship r and

$$\mathbb{E}\left[\log\left(P_\theta\left(embed_h, embed_t|z, r\right)\right)\right] \tag{4.11}$$

estimates the maximum likelihood. Since we want to sample from $P_\theta(z|r)$ in the generator, the first term aims to let $Q_\phi(z|embed_h, embed_t, r)$ be as close as possible to $P_\theta(z|r)$ which has a simple distribution $N(0, \mathrm{I})$ so that it is easy to sample from. Furthermore, if $P_\theta(z|r) \sim$

$N(0, \mathrm{I})$ and $Q(z|embed_h, embed_t, r) \sim N(\mu, \sigma^2)$, then a closed-form solution for the first term in Equation 4.9 is derived as:

$$\begin{aligned} -KL\left[Q_\phi\left(z|embed_h, embed_t, r\right)||P_\theta\left(z|r\right)\right] = -KL\left[N(\mu,\sigma)||N(0,\mathrm{I})\right] \\ = -\frac{1}{2}(tr(\sigma^2) + \mu^T\mu - D_L - \log\det(\sigma^2)) = -\frac{1}{2}\sum_l^{D_L}(\sigma_l^2 + \mu_l^2 - 1 - \log\sigma_l^2), \end{aligned} \tag{4.12}$$

where l in the subscript indicates the l-th dimension of the vector. Since it is more stable to have exponential term than a log term, we model $\log\left(\sigma^2\right)$ as σ^2 which results in the final closed-form of Equation 4.12:

$$-\frac{1}{2}\sum_l^{D_L}\left(\exp\left(\sigma^2\right)_l + \mu_l^2 - 1 - \sigma_l^2\right). \tag{4.13}$$

The second term in Equation 4.9 penalizes the maximum likelihood, where is the conditional probability $P_\theta(embed_h, embed_t|z, r)$ of a certain entity pair $<e_h, e_t>$ given the latent variable z and the relationship indicator r. The mean squared error (MSE) is adopted to calculate the difference between $<embed_h, embed_t>$ and $<embed'_h, embed'_t>$:

$$\begin{aligned} &\mathbb{E}\left[\log\left(P_\theta\left(embed_h, embed_t|z, r\right)\right)\right] = \\ &\frac{1}{2D_E}\left(||embed_h - embed'_h||_2^2 + ||embed_t - embed'_t||_2^2\right), \end{aligned} \tag{4.14}$$

where $\|\cdot\|_2$ is the vector ℓ_2 norm.

To minimize the $\mathcal{L}_{CRVAE}$, existing gradient-based optimizers such as Adadelta (Zeiler, 2012) can be used. Furthermore, a warm-up technique introduced in (Sønderby et al., 2016) can let

the training start with deterministic and gradually switch to variational, by multiplying β to the first term. The final loss function used for training is formulated as:

$$\begin{aligned}\mathcal{L}_{CRVAE} = &-\frac{\beta}{2}\sum_{l}^{D_L}\left(\exp\left(\sigma^2\right)_l + \mu_l^2 - 1 - \log\sigma_l^2\right) \\ &+ \frac{1}{2D_E}\left(||embed_h - embed'_h||_2^2 + ||embed_t - embed'_t||_2^2\right),\end{aligned} \tag{4.15}$$

where β is initialized as 0 and increase by 0.1 at the end of each training epoch, until it reaches 1.0 as its maximum.

4.3.4 Generator

When we would like to generate entity pairs of a specific relationship, a density-based sampling method is introduced for the generator to sample $\hat{z}$ from the distribution of latent variables conditioned on that relationship r.

Instead of using the latent variable z provided by certain μ and $\log\sigma^2$ in the encoding process from a certain e_h, e_t and r, the generator tries to sample $\hat{z}$ directly from $P_\theta(\hat{z}|r)$ to get the latent space value $\hat{z}$ for a particular relationship r. Once $\hat{z}$ is obtained, the decoder structure is used to decode the entity pair. Figure 16 illustrates the generative process. The denser region in the latent space $P_\theta(\hat{z}|r)$ indicates that more densely entity pairs are located in the manifold. Therefore, a sampling method that considers the density distribution of $P_\theta(\hat{z}|r)$ samples more often from that region to preserve the true latent space distribution. Specifically, for each relationship r, the density-based sampling samples $\hat{z}$ directly from $P_\theta(\hat{z}|r) \sim N(0, \mathrm{I})$, when trained properly. The resulting vectors $\hat{e}mbed_h$ and $\hat{e}mbed_t$ are mapped back to entity names

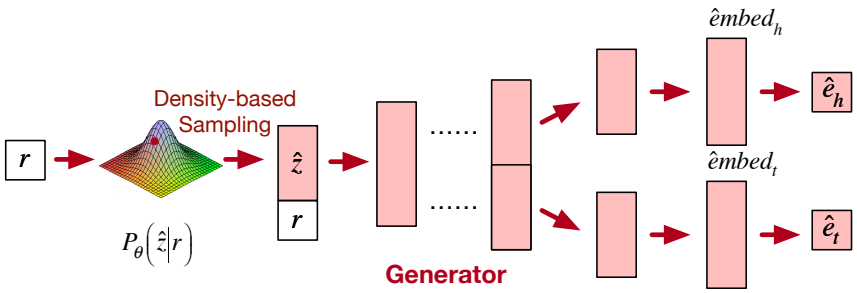


Figure 16: An overview of the proposed model CRVAE during generation.

in natural language, namely $\hat{e}_t$ and $\hat{e}_h$, by finding the nearest neighbor in their initial embedding space $\mathbb{R}^{1\times D_E}$ using W_{embed}. The ℓ-2 distance measure is used for the nearest neighbor search.

Note that the vocabulary of pre-trained word embedding is way more comprehensive than entities from labeled triplets in training. Using the pre-trained word embedding gives our model the ability to introduce unseen entities that are in the vocabulary, but not necessarily in the training data.

4.4 Evaluation

4.4.1 Dataset

The dataset consists of 46,018 real-world triplets in Chinese, and it covers six different types of medical relationships, where 70% data are used for training and 30% validation data are used for hyperparameter tuning. Since the proposed model discovers entity pairs by directly sampling from the latent space, not by verifying pre-determined test cases, we evaluate the generated entity pairs directly. Table VII shows the statistics and representative samples for each medical relationship. We use 200-dimensional word embeddings learned from a Chinese medical corpus

on the healthcare forum as the initial entity representation. The vocabulary covers 126,270 words.

RELATIONSHIP	COUNT	ENTITY PAIRS
Disease $\xrightarrow{Cause}$ Body Part	2320	<tricuspid insufficiency (三尖瓣闭锁), tricuspid valve (三尖瓣)> <vaginal cancer (阴道癌), reproductive system (生殖)> <hydrocephaly (脑积水), head (头部)>
Disease $\xrightarrow{RelatedTo}$ Disease	4614	<infant hydrocephalus (婴儿脑积水), congenital hydrocephalus (先天性脑积水)> <urethritis (尿道炎), cystitis (膀胱炎)> <retention of food in the stomach (食滞胃脘), infantile indigestion (小儿消化不良)>
Disease $\xrightarrow{Need}$ Examine	4185	<salicylates poisoning (水杨酸类中毒), routine urianlysis (尿常规)> <tetralogy triad (法洛三联症), electrocardiogram, ECG (心电图)> <epididymitis (附睾炎) , cremasteric reflex (提睾反射)>
Symptom $\xrightarrow{BelongTo}$ Department	8595	<anchylosis, stiffness of a joint (关节强直), orthopedics (骨科)> <female lower abdominal pain (女性小腹疼痛), gynecology (妇科)> <absent infant sucking reflex (吸吮反射消失), neonatology (新生儿科)>
Disease $\xrightarrow{Cause}$ Symptom	16642	<peritonitis (腹膜炎), abdominal venous engorgement (腹部静脉怒张)> <urethritis (尿道炎), urethra itching (尿道痒感)> <radial nerve palsy (桡神经麻痹), upper extremity weakness (上肢无力)>
Symptom $\xrightarrow{RelatedTo}$ Symptom	9662	<redness and swelling around the umbilicus (脐周红肿), periumbilical swelling (脐周肿胀)> <muscular contusion (肌肉挫伤), disinsertion (肌腱断裂)> <fingers benumbed with cold (手指冻肿), skin frostbite (皮肤冻伤)>

TABLE VII: Sample medical relationships and entity pairs.

4.4.2 Experiment Settings

Evaluation Metric Three evaluation metrics are introduced to quantitatively measure the generated relational medical entity pairs: quality, support, and novelty.

Quality Since it is hard for the machine to evaluate whether a entity pair is meaningful or not, human annotation is involved in assessing the quality of the generated entity pairs. We deploy a human annotation task on Amazon Mechanical Turk. Annotators need to pass at

least four in five sample cases to qualify the annotation. Majority voting of three annotators is adopted. The quality is measured by:

$$quality = \frac{\#\text{ of entity pairs that are meaningful}}{\#\text{ of all the generated entity pairs}}. \tag{4.16}$$

Support Besides human annotations, a support score quantitatively measures the belongingness of an entity pair generated by a specific relationship to existing entity pairs with that relationship. For each generated entity pair $<\hat{e}_h, \hat{e}_t>$, the support score measures its similarities to known entity pairs of each relationship r_c:

$$support_{<\hat{e}_h,\hat{e}_t,r_c>} = \frac{1}{1 + distance(\hat{e}mbed_h, \hat{e}mbed_t, r_c)}, \tag{4.17}$$

where $distance(\hat{e}mbed_h, \hat{e}mbed_t, r_c)$ calculates the distance between the vector $\hat{e}mbed_h - \hat{e}mbed_t$ and $NN_{r_c}\left(\hat{e}mbed_h - \hat{e}mbed_t\right)$ using distance measure such as cosine distance. The NN_{r_c} implements the nearest neighbor search over the $embed_h - embed_t$ space among all the entity pairs having the relationship r_c. For each generated entity pair, the support scores of all relationships are normalized:

$$norm\_support_{<\hat{e}_h,\hat{e}_t,r_c>} = \frac{support_{<\hat{e}_h,\hat{e}_t,r_c>}}{\sum_{r_i}^{|R|} support_{<\hat{e}_h,\hat{e}_t,r_i>}}. \tag{4.18}$$

The generated entity pair $<\hat{e}_h, \hat{e}_t>$ finds support from its estimated relationship which has the highest score, while the relationship r given during the generating process is considered as the

ground truth for $<\hat{e}_h, \hat{e}_t>$. The final support value is based on the accuracy of the estimated relationship and the ground truth relationship.

Novelty The ability to generate novel entity pairs is one of our key contributions. Due to different scope of knowledge among individuals, human annotators are not able to precisely evaluate the novelty. We measure the novelty of the generation process by:

$$novelty = \frac{\text{\# of entity pairs that do not exist in the dataset}}{\text{\# of all the generated entity pairs}}. \tag{4.19}$$

Baselines Considering that no known methods are currently available for the Generative Structured Knowledge Expansion problem, and we consider it unfair to compare with discriminative methods which have external resources or further data requirements, the performance on the following models are compared:

- **CRVAE-MONO**: The proposed model that works with all entity pairs having the same relationship in both training and generation. For each relationship, we train a separate CRVAE with entity pairs having that relationship.
- **RVAE**: The unconditional version of the model CRVAE where the relationship indicator r is not provided during model training and generation.
- **CRVAE-RAND**: The proposed model CRVAE with a random sampling based generator. Rather than using the density-based sampling strategy, the generator of CRVAE-RAND samples randomly from the latent space.

- **CRVAE**: The proposed method where entity pairs with all types of relationships are used together to train the model. The training is conditioned on relationships, and density-based sampling is used.
- **CRVAE-WA**: The proposed method with the warm-up strategy introduced in Section 4.3.3.

MODEL	QUALITY	SUPPORT	NOVELTY	LOSS (TRAIN / VALID)
CRVAE-MONO	0.6698	0.9550	0.5118	47.3002 / 116.6739
CRVAE-RAND	0.2550	0.3764	0.9952	43.0954 / 83.6589
CRVAE	0.7308	0.9048	0.5682	43.0954 / 83.6589
CRVAE-WA	0.7717	0.9291	0.6193	33.4399 / 57.9470

TABLE VIII: Performance comparison results.

4.4.3 Experiment Results

We generate 1000 entity pairs for each medical relationship for evaluation. Table VIII summarizes the performance of the proposed method when comparing with other alternatives. In summary, CRVAE-MONO demonstrates the power of generative model that learns commonalities purely from the diversely expressed entity pairs without substantial data requirements. By comparing CRVAE-RAND and CRVAE we show the effectiveness of the density-based sampling in generating high-quality entity pairs. The warm up technique adopted in CRVAE-WA is able to give CRVAE a further performance boost. As a qualitative measure, we also provide entity

pairs generated by the proposed model in Table IX, from which we can see the meaningful and novel structured knowledge discovered in a generative fashion.

Disease $\xrightarrow{Cause}$ Body Part
<dysentery (痢疾), intestine (肠)>
<brain tumor (脑瘤), head (头部)>
<leukopenia (白细胞减少症), vascular system (血液)>
Disease $\xrightarrow{RelatedTo}$ Disease
<foreign body in esophagus (食管异物), bowel obstruction (肠梗阻)>
<brain contusion (脑挫裂伤), amnesia (记忆障碍)>
<respiratory acidosis (呼吸性酸中毒), pulmonary edema (肺水肿)>
Disease $\xrightarrow{Need}$ Examine
<uremia (尿毒症), routine urianlysis (尿常规)>
<bacterial meningitis (细菌性脑膜炎), cranial CT (头颅CT)>
<bowel obstruction (肠梗阻), abdominal x-ray (腹部平片)>
Symptom $\xrightarrow{BelongTo}$ Department
<retained placenta (胎盘滞留), obstetrics (产科)>
<fluid retention (水潴留), nephrology (肾内科)>
<stuffy nose (鼻塞), otolaryngology (耳鼻咽喉科)>
Disease $\xrightarrow{Cause}$ Symptom
<otogenic brain abscess (耳源性脑脓肿), earache (耳痛)>
<neuritis (神经炎), numbness in the hands (手麻)>
<open head injury (开放性颅脑损伤), loss of consciousness (意识模糊)>
Symptom $\xrightarrow{RelatedTo}$ Symptom
<fatigue (乏力), feel wobbly and rough (四肢无力)>
<joint pain (关节痛), limited joint mobility (关节活动受限)>
<blurred vision (雾视), eye discomfort (眼睛不舒服)>

TABLE IX: Novel and meaningful entity pairs generated by the proposed method.

Generative Modeling Capacity Unlike discriminative models which utilize the discrepancies among instances of different classes to discriminate one class from another, the generative nature of the proposed method makes it generate entity pairs only when it fully understands the diverse expressions within each relationship. To validate such appealing property, we introduce the baseline CRVAE-MONO which works with all entity pairs having the same relationship in both training and generation.

Table X compares the fine-grained quality, support and novelty of the generated entity pairs of CRVAE-MONO and CRVAE on each relationship. The CRVAE-MONO achieves a reasonable performance on each relationship, which shows that the generative modeling has the ability to learn directly from the existing entity pairs without additional data requirement. Furthermore, when all types of entity pairs are trained altogether in CRVAE, we observe a consistent improvement in not only quality but also novelty.

Effectiveness of Density-based Sampling To validate the effectiveness of the density-based sampling for the generator, we compare the proposed method with CRVAE-RAND where a random sampling strategy is adopted. From Table VIII we can see that when the distribution of the latent space is not considered, the random sampling strategy in CRVAE-RAND tends to generate more entity pairs that are not seen in the existing dataset. However, the generated entity pairs are of low quality and support.

CRVAE adopts a density-based sampling. The dense region in the latent space indicates that more entity pairs are located. Therefore, in CRVAE, the quality and support of the generated

CRVAE-MONO	**QUALITY**	**SUPPORT**	**NOVELTY**	**LOSS (TRAIN/VALID)**
Disease $\xrightarrow{Cause}$ Body Part	0.6830	1.0000	0.4880	54.9830 / 126.7426
Disease $\xrightarrow{RelatedTo}$ Disease	0.6890	0.8700	0.4830	51.5131 / 155.0721
Disease $\xrightarrow{Need}$ Examine	0.7080	1.0000	0.5210	54.7635 / 136.4802
Symptom $\xrightarrow{BelongTo}$ Department	0.6870	1.0000	0.4660	39.0959 / 72.5872
Disease $\xrightarrow{Cause}$ Symptom	0.5870	0.9400	0.5730	37.3276 / 83.8797
Symptom $\xrightarrow{RelatedTo}$ Symptom	0.6650	0.9200	0.5400	46.1180 / 125.2818
CRVAE				
Disease $\xrightarrow{Cause}$ Body Part	0.7560	0.9990	0.7240	
Disease $\xrightarrow{RelatedTo}$ Disease	0.6910	0.7440	0.8670	
Disease $\xrightarrow{Need}$ Examine	0.7570	0.9810	0.8710	43.0954 / 83.6589
Symptom $\xrightarrow{BelongTo}$ Department	0.7680	0.9950	0.6130	
Disease $\xrightarrow{Cause}$ Symptom	0.7020	0.8820	0.9270	
Symptom $\xrightarrow{RelatedTo}$ Symptom	0.7110	0.8280	0.8880	

TABLE X: Performance comparison between CRVAE-MONO and CRVAE.

entity pairs benefit from sampling more often at denser regions in the latent space, resulting in less novel but higher quality entity pairs.

Ability to Infer Conditionally To effectively discover structured medical knowledge, one of our key contributions is to generate relational medical entity pairs for a specific relationship. That is, the ability to infer new entity pairs for a particular relationship without additional data preparation. Besides seamlessly incorporating this idea in the model design, we also show such conditional inference ability by visualization.

Figure 17 shows the μ of validation samples after being mapped into a two-dimensional space using Primary Component Analysis for dimension reduction. The samples are colored based on their ground truth relationship indicators. The left figure indicates that when the relationship indicator r is not given during the training/validation, RVAE is still able to map

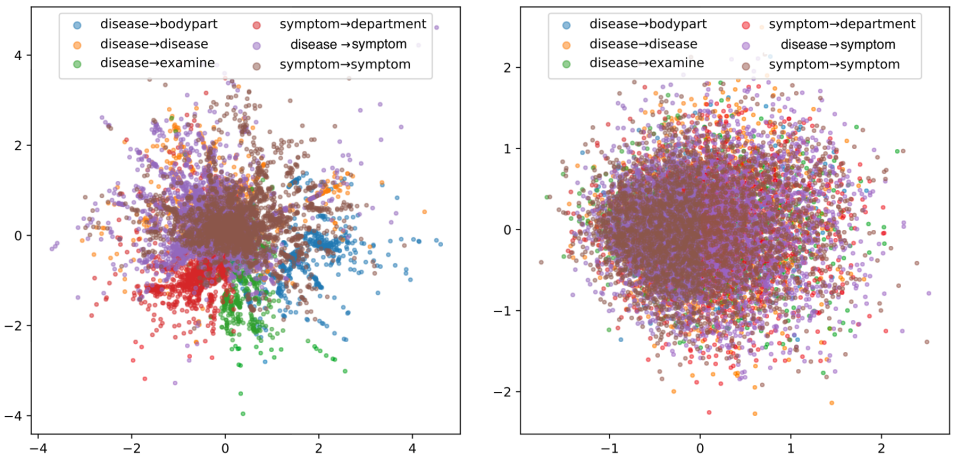


Figure 17: Visualizing the latent variable μ of RVAE (left) and CRVAE (right).

different relationships into various regions in the latent space, while a single distribution models all types of relationships. Such property is appealing for an unsupervised model, but since the relationship indicator r is not given during training, RVAE fails to generate entity pairs having a particular relationship, unless we manually assign a boundary for each relationship in the latent space. The right figure shows that when the relationship indicator r is incorporated during the training, CRVAE learns to let each relationship have a unified latent representation $P_\theta(\hat{z}|r)$. A separate but nearly identical distribution is used to model each relationship. Such property may enable the generator of our model to sample the expression variations from a relationship-independent latent space, while the relationship indicator r provides the categorical information regarding what type of relationship should the expression variation applies on.

Relationship-enhancing Entity Adjustment To show the effectiveness of relationship-enhancement, Table XI shows the nearest neighbors of a disease entity `genital tract`

• genital tract malformation (生殖道畸形) NN in the relationship-enhanced space $\mathbb{R}^{1\times D_R}$	NN in the initial embedding space $\mathbb{R}^{1\times D_E}$
genital tract (生殖道)	reproductive system (生殖系统)
reproductive system (生殖系统)	reproductive tract tumors (生殖道肿瘤)
heart malformations (心脏畸形)	urinary system malformations (泌尿系畸形)
chromosome abnormalities (染色体异常)	infertility (不孕)
reproductive tract tumors (生殖道肿瘤)	vaginal atresia (阴道闭锁)
generative organs (生殖器官)	genital tract (生殖道)
urinary system malformations (泌尿系畸形)	generative organs (生殖器官)
gastrointestinal malformations (消化道畸形)	acyesis (不孕症)
• muscle strain (肌肉拉伤) NN in the relationship-enhanced space $\mathbb{R}^{1\times D_R}$	NN in the initial embedding space $\mathbb{R}^{1\times D_E}$
strain (拉伤)	拉伤 (strain)
ligament strain (韧带拉伤)	muscle tear (肌肉撕裂)
sprain (扭伤)	pull-up (引体向上)
foot pain (足痛)	sprain (扭伤)
muscle tear (肌肉撕裂)	muscle fatigue (肌肉疲劳)
plantar fasciitis (足底筋膜炎)	tenosynovitis (腱鞘炎)
joint sprain (关节扭伤)	tendonitis (肌腱炎)
repetitive strain injury, RSI (劳损)	amount of exercise (运动量)

TABLE XI: The effectiveness of relationship-enhancing adjustment.

malformation (生殖道畸形) and a symptom entity muscle strain (肌肉拉伤) in their original embedding space, as well as in the space after relationship-enhancing.

From these cases we can see that the original entity representations trained with skip-gram (Mikolov et al., 2013a) tend to put entities in proximity when they appear in similar contexts. In the first case, the entity genital tract malformation (生殖道畸形) is in close proximity to infertility (不孕) and acyesis (不孕症). In the second case, entities that have similar context like pull-up (引体向上) and amount of exercise (运动量) are found near by the entity muscle strain (肌肉拉伤).

The translation layer adjusts the original entity representation so that they are more suitable for Generative Structured Knowledge Expansion. The nearest neighbors in the adjusted space are not necessarily entities that co-occur in the same context, but more relation-wise similar with the given entity. For example, `heart malformations` (心脏畸形) and `chromosome abnormalities` (染色体异常) may not be semantically similar with the given word `genital tract malformation` (生殖道畸形), but they may serve similar functionalities in a Disease $\xrightarrow{Cause}$ Symptom relationship.

4.4.4 Hyperparameter Analysis

We train the proposed model with a wide range of hyperparameter configurations, which are listed in Table XII. We vary the batch size from 64 to 256. The dimension D_R for translating the initial entity embeddings is set from 64 to 2048. We try two to seven hidden layers from $trans_{ht}$ to l_{ht} and from $[z, r]$ to $trans'_{ht}$, with different non-linear activation functions. For each hidden layer, the hidden unit number D_H is set from 2 to 1024. The latent dimension D_L is set from 2 to 200.

Parameter	Value
Batch Size	64, 128, 256
D_R	64, 128, 256, 512, 640, 768, 1024, 1280, 1536, 1792, 2048
D_H	2, 4, 8, 16, 32, 64, 128, 256, 512, 640, 768, 1024
D_L	2, 3, 4, 5, 10, 20, 50, 100, 200
Activation	ELU (Clevert et al., 2015), ReLU (Nair and Hinton, 2010), Sigmoid, Tanh
Optimizer	Adadelta (Zeiler, 2012), Adagrad (Duchi et al., 2011), Adam (Kingma and Ba, 2014), RMSProp (Tieleman and Hinton, 2012)

TABLE XII: Hyperparameter configurations.

The top-5 hyperparameter settings with low validation losses are shown in Table XIII. Among the combinations of hyperparameter configurations, we find that for fully connected hidden layers from $trans_{ht}$ to l_{ht}, a sequence of six consecutive layers: 1792-640-640-512-256-64 works the best for the encoder with ELU as the activation function. For $[z, r]$ to $trans'_{ht}$ in the decoder, such layer setting is organized in a reverse order. A batch size of 64 and the Adadelta optimizer work the best for our task. $D_R = 640$ is used. The latent dimension $D_L = 200$ is adopted for μ and σ^2. We use Xavier initialization (Glorot and Bengio, 2010) for weight variables and zeros for biases. Such configuration achieves a training loss of 43.0954 and a validation loss of 83.6589.

Batch	D_R	$\{D_H\}$	D_L	Act.	Optimizer	Loss(Training /Valid)
64	640	1792-640-640-512-256-64	200	ELU	Adadelta	43.0954 / 83.6589
64	640	1792-256-640-512-256-128	200	ELU	Adadelta	51.0695 / 86.9153
64	640	1792-256-640-512-256-64	200	ELU	Adadelta	50.4392 / 88.6438
128	640	1792-640-768-512-64-128	50	ELU	Adadelta	50.5997 / 89.0125
256	640	512-768-640-256-512	50	ELU	Adam	62.1955 / 89.2014

TABLE XIII: Hyperparameter analysis.

4.5 Related Works

Deep Generative Models: Recent years have witnessed an increasing interest in deep generative models that generate observable data based on hidden parameters. Various deep generative models have been developed, such as Generative Adversarial Networks (GANs) (Radford et al.,

2015) and Variational Autoencoders (VAEs) (Kingma and Welling, 2013). Unlike Generative Adversarial Networks (GANs) (Radford et al., 2015) which generate data based on arbitrary noises, the Variational Autoencoders (VAEs) (Kingma and Welling, 2013) setting we adopted is more expressive since it tries to model the underlying probability distribution of the data by latent variables so that we can sample from that distribution to generate new data accordingly. An increasing number of models and applications are proposed which consider data in different modalities, such as generating images (Pu et al., 2016; Gregor et al., 2015) or natural language (Bowman et al., 2016; Marcheggiani and Titov, 2016; Xu et al., 2017). (Yao et al., 2011) works on generative relation discovery with a probabilistic graphic model that requires hand-crafted relation-level features. As far as we know, the Generative Structured Knowledge Expansion problem we studied in this work, which is suitable for deep generative modeling, has not been studied in a generative perspective with restricted data requirement.

Knowledge Graph Completion: Existing knowledge graph completion methods (Bordes et al., 2011; Wang et al., 2014; Sun et al., 2012; Gardner and Mitchell, 2015; Wang et al., 2015b; Lin et al., 2016) are discriminative models. During training, those methods are trained to distinguish entity pairs of one relationship from another (Zeng et al., 2014; Lin et al., 2016), or to identify meaningful entity pairs from randomly sampled negative entity pairs with no relationships (Bordes et al., 2013; Socher et al., 2013). During testing, some candidate entity pairs are prepared ahead of time and given to the model. The model examines what kind of, and how likely there is a relationship for each candidate entity pair. Other works such as (Zhang et al., 2019) aligns entities from multiple existing knowledge graphs for synergistic completion.

The proposed model can be seen as augmenting an existing knowledge graph in a generative way. Although both knowledge graph completion task and our task provide additional entity pairs as their results, they share different objectives, and adopt entirely different approaches. The knowledge base completion models rely on the discrepancies among entity pairs of different relationships to distinguish one from another. Otherwise, random negative samples are used for discriminative training. Our model does not rely on discrepancies among relationships: it exploits the commonalities from diverse expressions within each relationship for a rational generation. Knowledge graph completion methods are also vulnerable to low-quality candidate entity pairs during testing: the truly meaningful entity pairs cannot be even obtained when they are not a part of the candidate entity pairs for discriminative models to examine. The choice of candidates involves additional human annotation to improve efficiency; otherwise, any dyadic combinations of medical entities need to be fed to and tested by the model. While the generative nature of our model makes it only generate rational entity pairs by learning from the existing rational ones: no additional data needs to be prepared for generative discovery.

Relationship Extraction: There is another related research area that studies relation extraction (Baeza-Yates and Tiberi, 2007; Agichtein and Gravano, 2000; Sahay et al., 2008; Mintz et al., 2009; Wang et al., 2015a; Jiang et al., 2017; Liu et al., 2017), which usually amounts to examining whether or not a relation exists between two given entities in a context (Culotta et al., 2006). Most relationship extraction methods require large amounts of high-quality external information, such as a large text corpus (Baeza-Yates and Tiberi, 2007; Agichtein and Gravano, 2000; Sahay et al., 2008; Li et al., 2016) and knowledge graphs (Chang et al., 2014; Syed et

al., 2010; Verga et al., 2017). However, in specific domains such as the medical domain, it is tedious and label-intensive to obtain a sufficient amount of free-text corpora which contains the co-occurrence of all kinds of entity pairs. Thus, we propose an effective generative method that learns from the existing entity pairs directly. Pre-trained word vectors are used in our model to provide initial entity representations, which do not introduce further labeling cost.

CHAPTER 5

Synonym Refinement on Structured Knowledge

Part of this chapter was published as "SynonymNet: Multi-context Bilateral Matching for Entity Synonyms", on ArXiv (Zhang et al., 2018b): `https://arxiv.org/abs/1901.00056`.

5.1 Introduction

Discovering synonymous entities from a massive corpus is an indispensable task for automated knowledge discovery. For each entity, its synonyms refer to the entities that can be used interchangeably under certain contexts. For example, `Clogged Nose` and `Nasal Congestion` are synonyms relative to the context in which they are mentioned. Given two entities, the synonym discovery task determines how likely these two entities are synonym with each other. The main goal of synonym discovery is to learn a metric that distinguishes synonym entities from non-synonym ones.

The synonym discovery task is challenging to deal with, a part of which due to the various entity expressions. For example, `U.S.A`/ `United States of America`/ `United States`/ `U.S.` refer to the same entity but are expressed quite differently. Recent works on synonym discovery focus on learning the similarity from entities and their character-level features (Neculoiu et al., 2016; Mueller and Thyagarajan, 2016). These methods work well for synonyms that share a lot of character-level features like `airplane`/ `aeroplane` or an entity and its abbreviation like `Acquired Immune Deficiency Syndrome`/ `AIDS`. However, a much larger number of synonym

entities in the real world do not share a lot of character-level features, such as `JD`/ `law degree`, or `clogged nose`/ `nasal congestion`. With only character-level features being used, these models hardly obtain the ability to discriminate entities that share similar semantics but are not alike verbatim.

Context information is helpful in indicating entity synonymity, as the meaning of an entity can be better reflected by the contexts in which it appears. Modeling the context for entity synonym usually suffers from following challenges: 1) **Semantic Structure**. Context, as a snippet of natural language sentence, is essentially semantically structured. Some existing models encode the semantic structures in the contexts implicitly during the entity representation learning (Mikolov et al., 2013b; Pennington et al., 2014; Peters et al., 2018). The context-aware entity representations embody meaningful semantics: entities with similar contexts are likely to live in proximity in the embedding space. Some other works extract and model contexts in an explicit manner with structured annotations. Structured annotations such as dependency parsing (Qu et al., 2017), user click information (Wei et al., 2009), or signed heterogeneous graphs (Ren and Cheng, 2015) are introduced to guide synonym discovery. 2) **Diverse Contexts**. An entity can be mentioned under a wide range of circumstances. Previous works on context-based synonym discovery either focus on entity information only (Neculoiu et al., 2016; Mueller and Thyagarajan, 2016), or use a single piece of context for each entity (Liao et al., 2017; Qu et al., 2017) to learn a similarity function for entity matching. While in practice, similar context is only a sufficient but not necessary condition for context matching. Notably, in some domains such as medical, the context expression preference varies a lot from individuals. For example,

`sinus congestion` is usually referred by medical professionals in the medical literature, while patients often use `stuffy nose` on social media. It is not practical to assume that each piece of context is equally informative to represent the meaning of an entity: a context may contribute differently when matched with different contexts of other entities. Thus it is imperative to focus on multiple pieces of contexts with a dynamic matching schema for accuracy and robustness.

In light of these challenges, we propose a framework to discover synonym entities from a massive corpus without additional structured annotation. Candidate entities are obtained from a massive text corpus unsupervisely. A novel neural network model SynonymNet is proposed to detect entity synonyms based on two given entities via a bilateral matching among multiple pieces of contexts in which each entity appears. A leaky unit is designed to explicitly alleviate the noises from uninformative context during the matching process.

The contribution of this work is summarized as follows:

- We propose SynonymNet, a context-aware bilateral matching model to detect entity synonyms. SynonymNet utilizes multiple pieces of contexts in which each entity appears, and a bilateral matching schema with leaky units to determine entity synonymity.
- We introduce a synonym discovery framework that adopts SynonymNet to obtain synonym entities from a free-text corpus without additional structured annotation.
- Experiments on generic and domain-specific real-world datasets in English and Chinese demonstrate the effectiveness of the proposed model for synonym discovery.

5.2 Proposed Approach

We introduce SynonymNet, our proposed model that detects whether or not two entities are synonyms to each other based on a bilateral matching between multiple pieces of contexts in which entities appear. Figure 18 gives an overview of the proposed model. The diamonds are entities. Each circle is associated with a piece of context in which an entity appears. SynonymNet learns to minimize the loss calculated using multiple pieces of contexts via bilateral matching with leaky units.

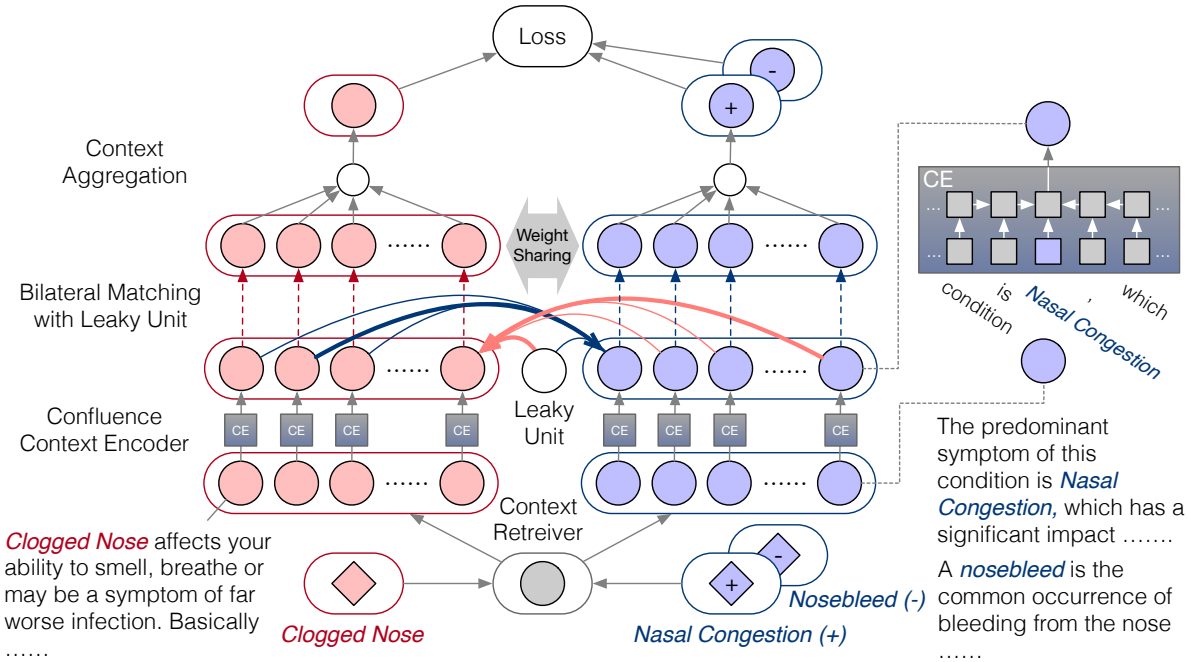


Figure 18: An overview of the proposed model SynonymNet.

5.2.1 Context Retriever

For each entity e, the context retriever randomly fetches P pieces of contexts from the corpus D in which the entity appears. We denote the retrieved contexts for e as a set $C = \{c_1, c_2, ..., c_P\}$, where P is the number of context pieces. Each piece of context $c_p \in C$ contains a sequence of words $c_p = (w_p^{(1)}, w_p^{(2)}, ..., w_p^{(T)})$, where T is the length of the context, which varies from one instance to another. $w_p^{(t)}$ is the t-th word in the p-th context retrieved for an entity e.

5.2.2 Confluence Context Encoder

For the p-th context c_p, an encoder tries to learn a continuous vector that represents the context. For example, a recurrent neural network (RNN) such as a bidirectional LSTM (Bi-LSTM) (Hochreiter and Schmidhuber, 1997) can be applied to sequentially encode the context into hidden states:

$$\overrightarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{t})}} = \text{LSTM}_{fw}(\mathbf{w}_p^{(t)}, \overrightarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{t-1})}}), \tag{5.1}$$

$$\overleftarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{t})}} = \text{LSTM}_{bw}(\mathbf{w}_p^{(t)}, \overleftarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{t+1})}}), \tag{5.2}$$

where $\mathbf{w}_p^{(t)}$ is the word embedding vector used for the word $w_p^{(t)}$. We could concatenate the last hidden state $\overrightarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{T})}}$ in the forward LSTM_{fw} with the first hidden state $\overleftarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{1})}}$ from the backward LSTM_{bw} to obtain the context vector $\mathbf{h}_p$ for c_p: $\mathbf{h}_p = [\overrightarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{T})}}, \overleftarrow{\mathbf{h}_\mathbf{p}^{(\mathbf{1})}}]$. However, such approach does not explicitly consider the location where the entity is mentioned in the context. As the context becomes longer, it is getting risky to simply rely on the gate functions of LSTM to properly encode the context.

We introduce an encoder architecture that models contexts for synonym discovery, namely the confluence context encoder. The confluence context encoder learns to encode the local information around the entity from the raw context, without utilizing additional structured annotations. It focuses on both forward and backward directions. However, the encoding process for each direction ceases immediately after it goes beyond the entity word in the context: $\mathbf{h}_p = [\mathbf{h}_\mathbf{p}^{(\overrightarrow{\mathbf{t_e}})}, \mathbf{h}_\mathbf{p}^{(\overleftarrow{\mathbf{t_e}})}]$, where t_e is the index of the entity word e in the context and $\mathbf{h}_p \in \mathbb{R}^{1\times d_{CE}}$. By doing this, the confluence context encoder summarizes the context while explicitly considers the entity's location in the context, where no additional computation cost is introduced.

Comparing with existing works for context modeling (Cambria et al., 2018) where the left context and right context are modeled separately, but with the entity word being discarded, the confluence context encoder preserves entity mention information as well as the inter-dependencies between the left and right contexts.

5.2.3 Bilateral Matching with Leaky Unit

Considering the base case, where we want to identify whether or not two entities, say e and k, are synonyms with each other, we propose to find the consensus information from multiple pieces of contexts via a bilateral matching schema. Recall that for entity e, P pieces of contexts $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_P\}$ are randomly fetched and encoded. And for entity k, we denote Q pieces of contexts being fetched and encoded as $G = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_Q\}$. Instead of focusing on a single piece of context to determine entity synonymity, we adopt a bilateral matching between multiple pieces of encoded contexts for both accuracy and robustness.

$H \rightarrow G$ matching phrase: For each $\mathbf{h_p}$ in H and $\mathbf{g_q}$ in G, the matching score $m_{p \rightarrow q}$ is calculated as:

$$m_{p \rightarrow q} = \frac{\exp(\mathbf{h}_p \mathbf{W}_{\text{BM}} \mathbf{g}_q^{\text{T}})}{\sum_{p' \in P} \exp(\mathbf{h}_{p'} \mathbf{W}_{\text{BM}} \mathbf{g}_q^{\text{T}})}, \tag{5.3}$$

where $\mathbf{W}_{\text{BM}} \in \mathbb{R}^{d_{CE} \times d_{CE}}$ is a bi-linear weight matrix.

Similarly, the $H \leftarrow G$ matching phrase considers how much each context $\mathbf{g}_q \in G$ could be useful to $\mathbf{h}_p \in H$:

$$m_{p \leftarrow q} = \frac{\exp(\mathbf{g}_q \mathbf{W}_{\text{BM}} \mathbf{h}_p^{\text{T}})}{\sum_{q' \in Q} \exp(\mathbf{g}_{q'} \mathbf{W}_{\text{BM}} \mathbf{h}_p^{\text{T}})}. \tag{5.4}$$

Note that $P \times Q$ matching needs to be conducted in total for each entity pair. We write the equations for each $\mathbf{h}_p \in H$ and $\mathbf{g}_q \in G$ for clarity. Regarding the implementation, the bilateral matching can be easily written and effectively computed in a matrix form, where a matrix multiplication is used $\mathbf{H}\mathbf{W}_{\text{BM}}\mathbf{G}^T \in \mathbb{R}^{P \times Q}$ where $\mathbf{H} \in \mathbb{R}^{P \times D_{CE}}$ and $\mathbf{G} \in \mathbb{R}^{Q \times D_{CE}}$. The matching score matrix $\mathbf{M}$ can be obtained by taking softmax on the $\mathbf{H}\mathbf{W}_{\text{BM}}\mathbf{G}^T$ matrix over certain axis (over 0-axis for $\mathbf{M}_{p \rightarrow q}$, 1-axis for $\mathbf{M}_{p \leftarrow q}$).

Not all contexts are informative during the matching for two given entities. For example, some contexts may contain intricate contextual information even if they mention the entity explicitly. In this work, we introduce a leaky unit during the bilateral matching, so that uninformative contexts can be routed via the leaky unit rather than forced to be matched with any informative contexts. The leaky unit is a domain-dependent vector $\mathbf{l} \in \mathbb{R}^{1 \times d_{CE}}$ learned

with the model. For simplicity, we keep $\mathbf{l}$ as a zero vector. If we use the $H \rightarrow G$ matching phrase as an example, the matching score from the leaky unit $\mathbf{l}$ to the q-th encoded context in $\mathbf{g}_q$ is:

$$m_{l \rightarrow q} = \frac{\exp(\mathbf{l}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}})}{\exp(\mathbf{l}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}}) + \sum_{p' \in P} \exp(\mathbf{h}_{p'}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}})}. \tag{5.5}$$

Then, if there is any uninformative context in H, say the $\tilde{p}$-th encoded context, $\mathbf{h}_{\tilde{p}}$ will contribute less when matched with $\mathbf{g}_q$ due to the leaky effect: when $\mathbf{h}_{\tilde{p}}$ is less informative than the leaky unit $\mathbf{l}$.

$$m_{\tilde{p} \rightarrow q} = \frac{\exp(\mathbf{h}_{\tilde{p}}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}})}{\exp(\mathbf{l}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}}) + \sum_{p' \in P} \exp(\mathbf{h}_{p'}\mathbf{W}_{\text{BM}}\mathbf{g}_q^{\text{T}})}. \tag{5.6}$$

5.2.4 Context Aggregation

The informativeness of a context for an entity should not be a fixed value: it heavily depends on the other entity and the other entity's contexts that we are comparing with. The bilateral matching scores indicate the matching among multiple pieces of encoded contexts for two entities. For each piece of encoded context, say $\mathbf{g}_q$ for the entity k, we use the highest matched score with its counterpart as the relative informativeness score of $\mathbf{g}_q$ to k, denote as $a_q = \max(m_{p \rightarrow q} | p \in P)$. Then, we aggregate multiple pieces of encoded contexts for each entity to a global context based on the relative informativeness scores:

$$\begin{aligned} &\text{for entity } e\text{:} \quad \bar{\mathbf{h}} = \sum\nolimits_{p \in P} a_p \mathbf{h}_p, \\ &\text{for entity } k\text{:} \quad \bar{\mathbf{g}} = \sum\nolimits_{q \in Q} a_q \mathbf{g}_q. \end{aligned} \tag{5.7}$$

Note that due to the leaky effect, less informative contexts are not forced to be heavily involved during the aggregation: the leaky unit may be more competitive than contexts that are less informative, thus assigned with larger matching scores. However, as the leaky unit is not used for aggregation, scores on informative contexts become more salient during context aggregation.

5.2.5 Training Objectives

We introduce two architectures for training the SynonymNet: a siamese architecture and a triplet architecture.

Siamese Architecture The Siamese architecture takes two entities e and k, along with their contexts H and G as the input. The following loss function L_{Siamese} is used in training for the Siamese architecture:

$$L_{\text{Siamese}} = yL_{+}(e,k) + (1-y)L_{-}(e,k), \tag{5.8}$$

where it contains losses for two cases: $L_{+}(e,k)$ when e and k are synonyms to each other ($y=1$), and $L_{-}(e,k)$ when e and k are not ($y=0$). Specifically, inspired by (Neculoiu et al., 2016), we have

$$\begin{aligned} L_{+}(e,k) &= \frac{1}{4}(1-s(\bar{\mathbf{h}},\bar{\mathbf{g}}))^2, \\ L_{-}(e,k) &= max(s(\bar{\mathbf{h}},\bar{\mathbf{g}})-m,0)^2, \end{aligned} \tag{5.9}$$

where $s(\cdot)$ is a similarity function, e.g. cosine similarity, and m is the margin value. $L_{+}(e,k)$ decreases monotonically as the similarity score becomes higher within the range of [-1,1]. $L_{+}(e,k) = 0$ when $s(\bar{\mathbf{h}},\bar{\mathbf{g}}) = 1$. For $L_{-}(e,k)$, it remains zero when $s(\bar{\mathbf{h}},\bar{\mathbf{g}})$ is smaller than a margin m. Otherwise $L_{-}(e,k)$ increases as $s(\bar{\mathbf{h}},\bar{\mathbf{g}})$ becomes larger.

Triplet Architecture The Siamese loss makes the model assign rational pairs with absolute high scores and irrational ones with low scores, while the rationality of entity synonymity could be quite relative to the context. The triplet architecture learns a metric such that the global context $\bar{\mathbf{h}}$ of an entity e is relatively closer to a global context $\bar{\mathbf{g}}_+$ of its synonym entity, say k_+, than it is to the global context $\bar{\mathbf{g}}_-$ of a negative example $\bar{\mathbf{g}}_-$ by some margin value m. The following loss function L_{Triplet} is used in training for the Triplet architecture:

$$L_{\text{Triplet}} = \max(s(\bar{\mathbf{h}}, \bar{\mathbf{g}}_-) - s(\bar{\mathbf{h}}, \bar{\mathbf{g}}_+) + m, 0). \tag{5.10}$$

5.2.6 Inference

The objective of the inference phase is to discover synonym entities for a given query entity from the corpus effectively. We utilize context-aware word representations to obtain candidate entities that narrow down the search space. The SynonymNet verifies entity synonymity by assigning a synonym score for two entities based on multiple pieces of contexts. The overall framework is described in Figure 19, which contains four steps (1): Obtain entity representations $\mathbf{W}_{\text{EMBED}}$ from the corpus D. (2): For each query entity e, search in the entity embedding space and construct a candidate entity set E_{NN}. (3): Retrieve contexts for the query entity e and each candidate entity $e_{NN} \in E_{NN}$ from the corpus D, and feed the encoded contexts into SynonymNet. (4): Discover synonym entities of the given entity by the output of SynonymNet.

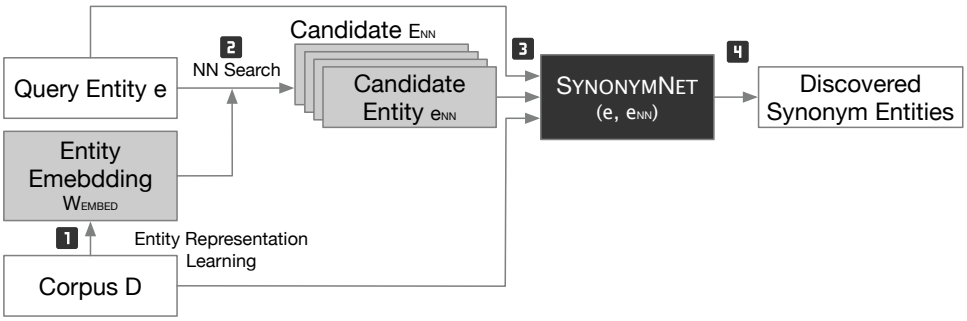


Figure 19: Synonym discovery during the inference phase with SYNONYMNET.

When given a query entity e, it is tedious and very ineffective to verify its synonymity with all the other possible entities. In the first step, we train entity representation unsupervisely from the massive corpus D using methods such as skip-gram (Mikolov et al., 2013b) or GloVe (Pennington et al., 2014). An embedding matrix can be learned $\mathbf{W}_{\text{EMBED}} \in \mathbb{R}^{v \times d_{\text{EMBED}}}$, where v is the number of unique tokens in D. Although these unsupervised methods utilize the context information to learn semantically meaningful representations for entities, they are not directly applicable to entity synonym discovery. However, they do serve as an effective way to obtain candidates as they tend to give entities with similar neighboring context words similar representations. For example, `nba championship`, `chicago black hawks` and `american league championship series` have similar representations because they tend to share some similar neighboring words. But they are not synonyms with each other.

In the second step, we construct a candidate entity list E_{NN} by finding nearest neighbors of a query entity e in the entity embedding space of $\mathbb{R}^{d_{\text{EMBED}}}$. Ranking entities by their proximities

with the query entity on the entity embedding space significantly narrows down the search space for synonym discovery.

For each candidate entity $e_{NN} \in E_{NN}$ and the query entity e, we randomly fetch multiple pieces of contexts in which entities are mentioned, and feed them into the proposed SynonymNet model.

SynonymNet calculates a score $s(e, e_{NN})$ based on the bilateral matching with leaky units over multiple pieces of contexts. The candidate entity e_{NN} is considered as a synonym to the query entity e when it receives a higher score $s(e, e_{NN})$ than other non-synonym entities, or exceeds a specific threshold.

Here we provide pseudo codes for the synonym discovery using SynonymNet.

Algorithm 2 Effective Synonym Discovery via SynonymNet.

Data: Candidate entity e, Entity Word Embeddings $W_{\text{EMBED}} \in \mathbb{R}^{v \times d}$, Document D
Result: Entity Set K where each $k \in K$ is a synonym entity of e
E_{NN} = NearestNeighbor(e, W_{EMBED})
Order E_{NN} by the distance to e;
for e_{NN} in E_{NN} **do**
    Retrieve Contexts for e_{NN} from Document D;
    Apply SynonymNet on e and e_{NN};
    if $s(e, e_{NN})$ >threshold **then**
        Add e_{NN} as a synonym of e to K;
    end if
end for

5.3 Evaluation

5.3.1 Datasets

Three datasets are prepared to show the effectiveness of the proposed model on synonym discovery. The Wiki dataset contains 6.8M documents from Wikipedia[1] with generic synonym entities obtained from Freebase[2]. The PubMed is an English dataset where 0.82M research paper abstracts are collected from PubMed[3] and UMLS[4] contains existing entity synonym information in the medical domain. The Wiki + FreeBase and PubMed + UMLS are public available datasets used in previous synonym discovery tasks (Qu et al., 2017). The MedBook is a Chinese dataset collected by authors where we collect 0.51M pieces of contexts from Chinese medical textbooks as well as online medical question answering forums. Synonym entities in the medical domain are obtained from MKG, a medical knowledge graph. Table XIV shows the dataset statistics.

5.3.2 Experiment Settings

Preprocessing Wiki +Freebase and PubMed + UMLS come with entities and synonym entity annotations, we adopt the Stanford CoreNLP package to do the tokenization. For MedBook, a

---

[1]https://www.wikipedia.org/

[2]https://developers.google.com/freebase

[3]https://www.ncbi.nlm.nih.gov/pubmed

[4]https://www.nlm.nih.gov/research/umls/

Dataset	Wiki + FreeBase	PubMed + UMLS	MedBooK + MKG
#ENTITY	9274	6339	32,002
#VALID	394	386	661
#TEST	104	163	468
#SYNSET	4615	708	6600
#CONTEXT	6,839,331	815,644	514,226
#VOCAB	472,834	1,069,061	270,027

TABLE XIV: Dataset statistics.

Chinese word segmentation tool Jieba[1] is used to segment the corpus into meaningful entities and phrases. We remove redundant contexts in the corpus and filter out entities if they appear in the corpus less than five times. For entity representations, the proposed model works with various unsupervised word embedding methods. Here for simplicity, we adopt skip-gram (Mikolov et al., 2013b) with a dimension of 200. Context window is set as 5 with a negative sampling of 5 words for training.

Evaluation Metric For synonym detection using SynonymNet and other alternatives, we train the models with existing synonym and randomly sampled entity pairs as negative samples. During testing, we also sample random entity pairs as negative samples to evaluate the performance. Note that all test synonym entities are from unobserved groups of synonym entities: none of the test entities is observed in the training data. Thus evaluations are done in a completely cold-start setting.

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

The area under the curve (AUC) and Mean Average Precision (MAP) are used to evaluate the model. AUC is used to measure how well the models assign high scores to synonym entities and low scores to non-synonym entities. An AUC of 1 indicates that there is a clear boundary between scores of synonym entities and non-synonym entities. Additionally, a single-tailed t-test is conducted to evaluate the significance of performance improvements when we compare the proposed SynonymNet model with all the other baselines.

For synonym discovery during the inference phase, we obtain candidate entities E_{NN} from K-nearest neighbors of the query entity in the entity embedding space, and rerank them based on the output score $s(e, e_{NN})$ of the SynonymNet for each $e_{NN} \in E_{NN}$. We expect candidate entities in the top positions are more likely to be synonym with the query entity. We report the precision at position K (P@K), recall at position K (R@K), and F1 score at position K (F1@K).

Baselines We compare the proposed model with the following alternatives.

- **word2vec** (Mikolov et al., 2013b): a word embedding approach based on entity representations learned from the skip-gram algorithm. We use the learned word embedding to train a classifier for synonym discovery. A scoring function $Score_D(u, v) = x_u \mathbf{W} x_v^T$ is used as the objective.
- **GloVe** (Pennington et al., 2014): another word embedding approach. The entity representations are learned based on the GloVe algorithm. The classifier is trained with the same scoring function $Score_D$, but with the learned glove embedding for synonym discovery.

- **SRN** (Neculoiu et al., 2016): a character-level approach that uses a siamese multi-layer bi-directional recurrent neural networks to encode the entity as a sequence of characters. The hidden states are averaged to get an entity representation. Cosine similarity is used in the objective.
- **MaLSTM** (Mueller and Thyagarajan, 2016): another character-level approach. We adopt MaLSTM by feeding the character-level sequence to the model. Unlike SRN that uses Bi-LSTM, MaLSTM uses a single direction LSTM and l-1 norm is used to measure the distance between two entities.
- **DPE** (Qu et al., 2017): a model that utilizes dependency parsing results as the structured annotation on a single piece of context for synonym discovery.
- **SynonymNet** is the proposed model, we used siamese loss (Equation 5.9) and triplet loss (Equation 5.10) as the objectives, respectively.

5.3.3 Experiment Results

We report Area Under the Curve (AUC) and Mean Average Precision (MAP) on three datasets in Table XV.

From the upper part of Table XV we can see that SynonymNet performances consistently better than other baselines on three datasets. SynonymNet with the triplet training objective achieves the best performance on Wiki +Freebase, while the Siamese objective works better on PubMed + UMLS and MedBook + MKG. Word2vec is generally performing better than GloVe. SRNs achieve decent performance on PubMed + UMLS and MedBook + MKG. This is probably

MODEL	Wiki + Freebase		PubMed + UMLS		MedBook + MKG	
	AUC	MAP	AUC	MAP	AUC	MAP
word2vec (Mikolov et al., 2013b)	0.9272	0.9371	0.9301	0.9422	0.9393	0.9418
GloVe (Pennington et al., 2014)	0.9188	0.9295	0.8890	0.8869	0.7250	0.7049
SRN (Neculoiu et al., 2016)	0.8864	0.9134	0.9517	0.9559	0.9419	0.9545
MaLSTM (Mueller and Thyagarajan, 2016)	0.9178	0.9413	0.8151	0.8554	0.8532	0.8833
DPE (Qu et al., 2017)	0.9461	0.9573	0.9513	0.9623	0.9479	0.9559
SynonymNet (Pairwise)	0.9831†	0.9818†	**0.9838**†	**0.9872**†	**0.9685**	**0.9673**
w/o Leaky Unit	0.9827†	0.9817†	0.9815†	0.9847†	0.9667	0.9651
w/o Confluence Encoder (Bi-LSTM)	0.9683†	0.9625†	0.9495	0.9456	0.9311	0.9156
SynonymNet (Triplet)	**0.9877**†	**0.9892**†	0.9788†	0.9800†	0.9410	0.9230
w/o Leaky Unit	0.9705†	0.9631†	0.9779†	0.9821†	0.9359	0.9214
w/o Confluence Encoder (Bi-LSTM)	0.9582†	0.9531†	0.9412	0.9288	0.9047	0.8867

TABLE XV: Test performance in AUC and MAP on three datasets.

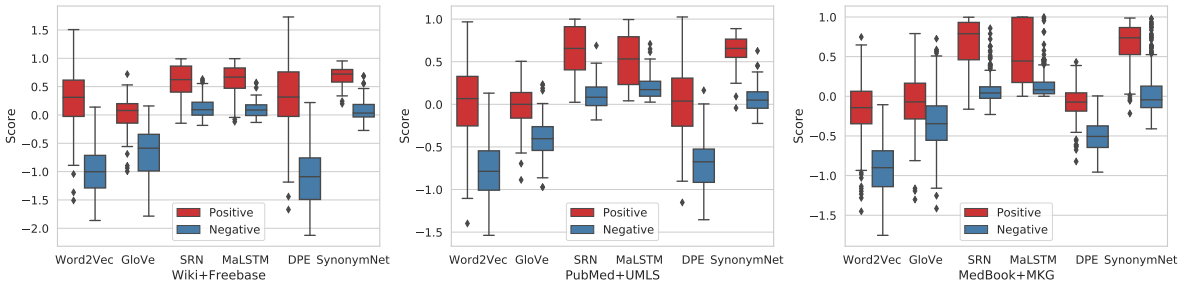


Figure 20: Test synonym score distributions on positive and negative entity pairs.

because the synonym entities obtained from the medical domain tend to share more character-level similarities, such as `6-aminohexanoic acid` and `aminocaproic acid`. However, even if the character-level features are not explicitly used in our model, our model still performances better, by exploiting multiple pieces of contexts effectively. DPE has the best performance among other baselines, by annotating each piece of context with dependency parsing results. However, the dependency parsing results could be error-prone for the synonym discovery task,

especially when two entities share the similar usage but with different semantics, such as `NBA finals` and `NFL playoffs`.

We conduct statistical significance tests to validate the performance improvement. The single-tailed t-test is performed for all experiments, which measures whether or not the results from the proposed model are significantly better than ones from baselines. The numbers with † markers in Table XV indicate that the improvement is significant with $p<0.05$.

Table XVI reports the performance in P@K, R@K, and F1@K.

	Wiki + Freebase			**PubMed + UMLS**			**MedBook + MedKG**		
	P@K	R@K	F1@K	P@K	R@K	F1@K	P@K	R@K	F1@K
K=1	0.3455	0.3455	0.3455	0.2400	0.0867	0.1253	0.3051	0.2294	0.2486
K=5	0.1818	0.9091	0.3030	0.2880	0.7967	0.3949	0.2388	0.8735	0.3536
K=10	0.1000	1.0000	0.1818	0.1800	1.0000	0.2915	0.1418	1.0000	0.2360

TABLE XVI: Performance on Synonym Discovery.

Besides numeric metrics, we also use box plots to represent the score distributions for each method on all three datasets in Figure 20. The red bars indicate scores on positive entity pairs that are synonym with each other, while the blue bars indicate scores on negative entity pairs. A general conclusion is that our model assigns higher scores for synonym entity pairs, marginally higher than other non-synonym entity pairs when compared with other alternatives.

5.3.4 Ablation Study

To study the contribution of different modules of SynonymNet for synonym discovery, we also report ablation test results in the lower part of Table XV. "w/o Confluence Context Encoder" uses the Bi-LSTM as the context encoder. The last hidden states in both forward and backward directions in Bi-LSTM are concatenated; "w/o Leaky Unit" does not have the ability to ignore uninformative contexts during the bilateral matching process: all contexts retrieved based on the entity, whether informative or not, are utilized in bilateral matching. From the lower part of Table XV we can see that both modules (Leaky Unit and Confluence Encoder) contribute to the effectiveness of the model. The leaky unit contributes 1.72% improvement in AUC and 2.61% improvement in MAP on the Wiki dataset when trained with the triplet objective. The Confluence Encoder gives the model an average of 3.17% improvement in AUC on all three datasets, and up to 5.17% improvement in MAP.

5.3.5 Hyperparameters

We train the proposed model with a wide range of hyperparameter configurations, which are listed in Table XVII. For the model architecture, we vary the number of randomly sampled contexts $P = Q$ for each entity from 1 to 20. Each piece of context is chunked by a maximum length of T. For the confluence context encoder, we vary the hidden dimension d_{CE} from 8 to 1024. The margin value m in triplet loss function is varied from 0.1 to 1.75. For the training, we try different optimizers (Adam (Kingma and Ba, 2014), RMSProp (Tieleman and Hinton, 2012), adadelta (Zeiler, 2012), and Adagrad (Duchi et al., 2011)), with the learning rate varying from 0.0003 to 0.01. Different batch sizes are used to train the model. We apply random search

to obtain the best-performing hyperparameter setting on the validation split for each dataset, as shown in Table XVIII.

HYPERPARAMETERS	**VALUE**
P (context number)	{1, 3, 5, 10, 15, 20}
T (maximum context length)	{10, 30, 50, 80}
d_{CE} (layer size)	{8, 16, 32, 64, 128, 256, 512, 1024}
m (margin)	{0.1, 0.25, 0.5, 0.75, 1.25, 1.5, 1.75}
Optimizer	{Adam, RMSProp, Adadelta, Adagrad}
Batch Size	{4, 8, 16, 32, 64, 128}
Learning Rate	{0.0003, 0.0001, 0.001, 0.01}

TABLE XVII: Hyperparameter settings.

DATASETS	P	T	d_{CE}	m	Optimizer	Batch Size	Learning Rate
Wiki + Freebase	20	50	256	0.75	Adam	16	0.0003
PubMed + UMLS	20	50	512	0.5	Adam	16	0.0003
MedBook + MKG	5	80	256	0.75	Adam	16	0.0001

TABLE XVIII: Hyperparameters.

Furthermore, we provide sensitivity analysis of the proposed model with different hyperparameters in Wiki + Freebase dataset in Figure 21. Figure 21 shows the performance curves when we vary one hyperparameter while keeping the remaining fixed. As the number of contexts

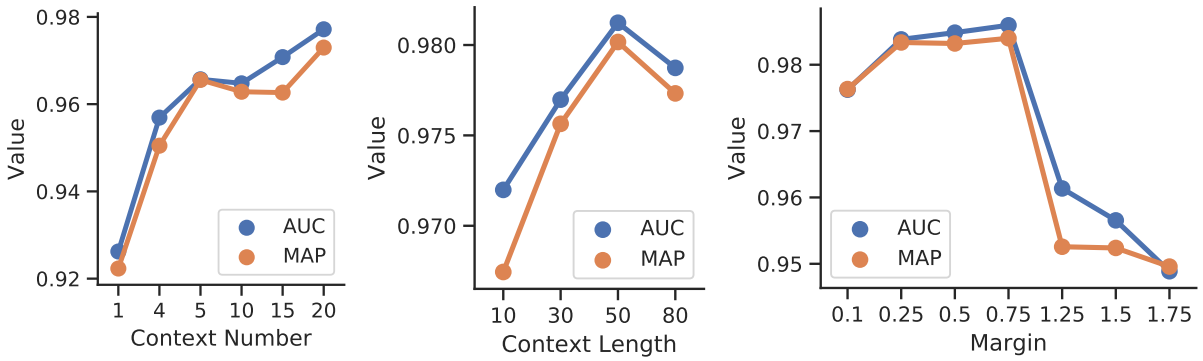


Figure 21: Sensitivity analysis.

P increases, the model generally performs better. Due to limitations on computing resources, we are only able to verify the performance of up to 20 pieces of randomly sampled contexts. The model achieves the best AUC and MAP when the maximum context length $T = 50$: longer contexts may introduce too much noise while shorter contexts may be less informative.

5.3.6 Case Studies

Table XIX and Table XX show a case for entity `UNGA`. The candidate entities in Table XIX are generated with pretrained word embedding using skip-gram. Table XX shows the discovered synonym entities by the proposed SynonymNet model, where a threshold of 0.8 on the SynonymNet score is used.

5.4 Related works

Synonym Discovery The synonym discovery focuses on detecting entity synonyms. Most existing works try to achieve this goal by learning from structured information such as query logs (Ren and Cheng, 2015; Chaudhuri et al., 2009; Wei et al., 2009). While in this work,

Candidate Entities	Cosine Similarity
united_nations_general_assembly||m.07vp7||	0.847374
un_human_rights_council	0.823727
the_united_nations_general_assembly	0.813736
un_security_council||m.07vnr||	0.794973
palestine_national_council	0.791135
world_health_assembly||m.05_gl9||	0.790837
united_nations_security_council||m.07vnr||	0.787999
general_assembly_resolution	0.784581
the_un_security_council	0.784280
ctbt	0.777627
north_atlantic_council||m.05pmgy||	0.775703
resolution_1441	0.773064
non-binding_resolution||m.02pj22f||	0.771475
unga||m.07vp7||	0.770623

TABLE XIX: Candidate entities retrieved for `UNGA`.

Final Entities	SynonymNet Score
united_nations_general_assembly||m.07vp7||	0.842602
the_united_nations_general_assembly	0.801745
unga||m.07vp7||	0.800719

TABLE XX: Discovered synonym entities for `UNGA` using SynonymNet.

we focus on synonym discovery from free-text natural language contexts, which requires less annotation and is more challenging.

Some existing works try to detect entity synonyms by entity-level similarities (Lin et al., 2003; Roller et al., 2014; Neculoiu et al., 2016; Wieting et al., 2016). For example, (Roller et al., 2014) introduce distributional features for hypernym detection. (Neculoiu et al., 2016) use

a Siamese structure that treats each entity as a sequence of characters, and uses a Bi-LSTM to encode the entity information. Such approach may be helpful for synonyms with similar spellings, or dealing with abbreviations. Without considering the context information, it is hard for the aforementioned methods to infer synonyms that share similar semantics but are not alike verbatim, such as `JD` and `law degree`.

Various approaches (Snow et al., 2005; Sun and Grishman, 2010; Liao et al., 2017; Cambria et al., 2018) are proposed to incorporate context information to characterize entity mentions. However, these models are not designed for synonym discovery. (Qu et al., 2017) utilize additional structured annotations, e.g. dependency parsing result, as the context of the entity for synonym discovery. While we aim to discover synonym entities from a free-text corpus without structured annotation.

Sentence Matching There is another related research area that studies sentence matching. Early works try to learn a meaningful single vector to represent the sentence (Tan et al., 2015; Mueller and Thyagarajan, 2016). These models do not consider the word-level interactions from two sentences during the matching. (Wang and Jiang, 2016; Wang et al., 2016; Wang et al., 2017) introduce multiple instances for matching with varying granularities. Although the above methods achieve decent performance on sentence-level matching, the sentence matching task is different from context modeling for synonym discovery in essence. Context matching focuses on local information, especially the words before and after the entity word; while the overall sentence could contain much more information, which is useful to represent the sentence-level semantics, but can be quite noisy for context modeling. We adopt a confluence encoder

to model the context, which is able to aware of the location of an entity in the context while preserving information flow from both left and right contexts.

Moreover, sentence matching models do not explicitly deal with uninformative instances: max-pooling strategy and attention mechanism are introduced. The max-pooling strategy picks the most informative one and ignores all the other less informative ones. In context matching, such property could be unsatisfactory as an entity is usually associated with multiple contexts. We adopt a bilateral matching which involves a leaky unit to explicitly deal with uninformative contexts, so as to eliminate noisy contexts while preserving the expression diversity from multiple pieces of contexts.

CHAPTER 6

CONCLUSION

(Part of the chapter was previously published in (Zhang et al., 2016; Zhang et al., 2017; Zhang et al., 2018a; Zhang et al., 2019; Zhang et al., 2018b).) In this dissertation, we have explored the structured knowledge discovery from the massive text corpus. More specifically, two general and strongly correlated research objectives are explored: one is to harness structured information for natural language understanding and modeling, and the other objective is to effectively expand and refine structured knowledge harnessing the massiveness of the text corpus. We thoroughly studied four different research problems: Structured Intent Detection for Natural Language Understanding, Structure-aware Natural Language Modeling, Generative Structured Knowledge Expansion, and Synonym Refinement on Structured Knowledge. We have evaluated the effectiveness of the proposed approaches on various user-generated text corpora such as the question-answering corpus, web search queries, voice commands, and documents by extensive quantitative experiments and case studies. The main contributions of our works are summarized as follows:

- We studied the Structured Intent Detection problem that aims to understand complicated user intentions in online question-answering discussion forums. An Intent Graph is formulated to possess explicit constraints on concept mentions as nodes and semantic transitions among concepts as directed edges on the Intent Graph, which are key com-

ponents to characterize Structured Intents. A neural network model named coCTI-MTL based on multi-task learning is introduced to extract concept mentions as well as semantic transitions collectively as a sub-graph of the Intent Graph to represent Structured Intents. Empirical results show that the proposed method can accurately detect complicated user intents from real-world information-seeking text corpora generated by users on an online medical question-answering discussion forum. Being able to detect complicated intents may further benefit other tasks such as dialogue management, recommendation, and question rewriting.

- We presented a capsule neural network based model, namely Capsule-NLM, to harness the hierarchical relationships among words, slots, and intents in the utterance for joint slot filling and intent detection. Unlike treating slot filling as a sequential prediction problem, the proposed model Capsule-NLM assigns each word to its most appropriate slots in SlotCaps by a dynamic routing-by-agreement schema. The learned word-level slot representations are further aggregated to get the utterance-level intent representations via dynamic routing-by-agreement. A re-routing schema is proposed to further synergize the slot filling performance using the inferred intent representation. Experiments on two real-world datasets show the effectiveness of the proposed models when compared with other alternatives as well as existing NLU services.

- We introduce a generative perspective to study the Generative Structured Knowledge Expansion problem, which aims to expand the scale of high-quality yet novel structured knowledge from the massive text corpus with minimized annotation and additional data

collection. We propose a model named Conditional Relationship Variational Autoencoder (CRVAE) which capitalizes on rich semantic information learned unsupervisely from a large text corpus as entity representations. The proposed model defines each relationship by solely learning the expression commonalities and differences from existing entity pairs that are diversely expressed. It generates meaningful, novel entity pairs of a specific relationship by directly sampling from the learned latent space without the requirement of additional context information. The performance of the proposed method is evaluated on real-world data both quantitatively and qualitatively.

- We developed a framework for synonym discovery from the text corpus without structured annotation. A novel neural network model SynonymNet is introduced for synonym detection, which tries to determine whether or not two given entities are synonym with each other. The proposed model is able to automatically detect synonym entities from a large corpus, which could help remove duplicate entities in knowledge graphs and thus improve the quality of structured knowledge. SynonymNet makes use of multiple pieces of contexts in which each entity is mentioned, and compares the context-level similarity via a bilateral matching schema to determine synonymity. Experiments on three real-world datasets show that the proposed method SynonymNet can discover synonym entities effectively on both generic datasets (Wiki+Freebase in English), as well as domain-specific datasets (PubMed+UMLS in English and MedBook+MKG in Chinese) with an improvement up to 4.16% in AUC and 3.19% in MAP.

APPENDICES

.1 ACM Copyright Letter

"Authors can reuse any portion of their own work in a new work of their own (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

Contributing complete papers to any edited collection of reprints for which the author is not the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected)." [1]

---

[1] `http://authors.acm.org/main.html`

.2 IEEE Copyright Letter

Title: Bringing semantic structures to user intent detection in online medical queries

Conference Proceedings: 2017 IEEE International Conference on Big Data (Big Data)

Author: Chenwei Zhang; Nan Du; Wei Fan; Yaliang Li; Chun-Ta Lu; Philip S. Yu

Publisher: IEEE

Date: 11-14 Dec. 2017

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.

Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

.3 arXiv.org Copyright Letter

arXiv.org - Non-exclusive license to distribute

The URI http://arxiv.org/licenses/nonexclusive-distrib/1.0/ is used to record the fact that the submitter granted the following license to arXiv.org on submission of an article:

- I grant arXiv.org a perpetual, non-exclusive license to distribute this article.
- I certify that I have the right to grant this license.
- I understand that submissions cannot be completely removed once accepted.
- I understand that arXiv.org reserves the right to reclassify or reject any submission.

Revision history

2004-01-16 - License above introduced as part of arXiv submission process
2007-06-21 - This HTML page created

Contact

.4 ACL Copyright Letter

ACL Anthology

Search...

You're viewing the latest version of the ACL Anthology. Give feedback

What is the copyright for materials in the ACL Anthology?

The ACL materials that are hosted in the Anthology are licensed to the general public under a liberal usage policy that allows unlimited reproduction, distribution and hosting of materials on any other website or medium, for non-commercial purposes. Prior to 2016, all ACL materials are licensed using the Creative Commons 3.0 BY-NC-SA (Attribution, Non-Commercial, Share-Alike) license. As of 2016, this policy has been relaxed further, and all subsequent materials are available to the general public on the terms of the Creative Commons 4.0 BY (Attribution) license; this means both commercial and non-commercial use is explicitly licensed to all.

Note that these policies only cover ACL materials. As with the DOIs, this policy does not cover third-party materials. For reproduction privileges for such materials, please approach the respective organizations.

Corrections Credits FAQ Issues Submitting Volunteering

Matt Post (Editor, 2019–) | Min-Yen Kan (Editor, 2008–2018) | Steven Bird (Editor, 2001–2007)

CITED LITERATURE

[Agichtein and Gravano, 2000]Agichtein, E. and Gravano, L.: Snowball: Extracting relations from large plain-text collections. In Proceedings of the Fifth ACM Conference on Digital Libraries, pages 85–94, 2000.

[Baeza-Yates and Tiberi, 2007]Baeza-Yates, R. and Tiberi, A.: Extracting semantic relations from query logs. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 76–85. ACM, 2007.

[Bengio and others, 2009]Bengio, Y. et al.: Learning deep architectures for ai. Foundations and trends® in Machine Learning, 2009.

[Bengio et al. , 1994]Bengio, Y., Simard, P., and Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.

[Bollacker et al. , 2008]Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pages 1247–1250. ACM, 2008.

[Bordes et al. , 2013]Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems, pages 2787–2795, 2013.

[Bordes et al. , 2011]Bordes, A., Weston, J., Collobert, R., and Bengio, Y.: Learning structured embeddings of knowledge bases. In Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

[Bowman et al. , 2016]Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S.: Generating sentences from a continuous space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 10–21, 2016.

[Cai et al. , 2017]Cai, R., Zhu, B., Ji, L., Hao, T., Yan, J., and Liu, W.: An cnn-lstm attention approach to understanding user query intent from online health communities. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pages 430–437. IEEE, 2017.

[Cambria et al. , 2018]Cambria, E., Poria, S., Hazarika, D., and Kwok, K.: Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[Chang et al. , 2014]Chang, K.-W., Yih, W.-t., Yang, B., and Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 1568–1579, 2014.

[Chaudhuri et al. , 2009]Chaudhuri, S., Ganti, V., and Xin, D.: Exploiting web search to generate synonyms for entities. In Proceedings of the 18th International Conference on World Wide Web, pages 151–160. ACM, 2009.

[Chen et al. , 2016]Chen, Y.-N., Hakkani-Tür, D., Tür, G., Gao, J., and Deng, L.: End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In Interspeech, pages 3245–3249, 2016.

[Cheng et al. , 2016]Cheng, J., Dong, L., and Lapata, M.: Long short-term memory-networks for machine reading. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 551–561, 2016.

[Chiang et al. , 2012]Chiang, F., Andritsos, P., Zhu, E., and Miller, R. J.: Autodict: Automated dictionary discovery. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, pages 1277–1280. IEEE, 2012.

[Cho et al. , 2014]Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[Chung et al. , 2014]Chung, J., Gulcehre, C., Cho, K., and Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[Clevert et al. , 2015]Clevert, D.-A., Unterthiner, T., and Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015.

[Culotta et al. , 2006]Culotta, A., McCallum, A., and Betz, J.: Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pages 296–303. Association for Computational Linguistics, 2006.

[De and Kopparapu, 2010]De, A. and Kopparapu, S. K.: A rule-based short query intent identification system. In Proceedings of the 2010 International Conference on Signal and Image Processing, pages 212–216. IEEE, 2010.

[Devlin et al. , 2019]Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[Duchi et al. , 2011]Duchi, J., Hazan, E., and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.

[Fabian et al. , 2007]Fabian, M., Gjergji, K., Gerhard, W., et al.: Yago: A core of semantic knowledge unifying wordnet and wikipedia. In Proceedings of the 16th International World Wide Web Conference, WWW, pages 697–706, 2007.

[Gardner and Mitchell, 2015]Gardner, M. and Mitchell, T.: Efficient and expressive knowledge base completion using subgraph feature extraction. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1488–1498, 2015.

[Glorot and Bengio, 2010]Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 249–256, 2010.

[Godbole et al. , 2010]Godbole, S., Bhattacharya, I., Gupta, A., and Verma, A.: Building reusable dictionary repositories for real-world text mining. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pages 1189–1198. ACM, 2010.

[Gong et al. , 2018]Gong, J., Qiu, X., Wang, S., and Huang, X.: Information aggregation via dynamic routing for sequence encoding. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2742–2752, 2018.

[Goo et al. , 2018]Goo, C.-W., Gao, G., Hsu, Y.-K., Huo, C.-L., Chen, T.-C., Hsu, K.-W., and Chen, Y.-N.: Slot-gated modeling for joint slot filling and intent prediction. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), volume 2, pages 753–757, 2018.

[Gregor et al. , 2015]Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D.: Draw: A recurrent neural network for image generation. In Proceedings of the International Conference on Machine Learning, pages 1462–1471, 2015.

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

[Hasegawa et al. , 2004]Hasegawa, T., Sekine, S., and Grishman, R.: Discovering relations among named entities from large corpora. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, page 415. Association for Computational Linguistics, 2004.

[He et al. , 2018]He, L., Liu, B., Li, G., Sheng, Y., Wang, Y., and Xu, Z.: Knowledge base completion by variational bayesian neural tensor decomposition. Cognitive Computation, 10(6):1075–1084, 2018.

[Hinton et al. , 2011]Hinton, G. E., Krizhevsky, A., and Wang, S. D.: Transforming auto-encoders. In Proceedings of the International Conference on Artificial Neural Networks, pages 44–51. Springer, 2011.

[Hochreiter, 1998]Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02):107–116, 1998.

[Hochreiter and Schmidhuber, 1997]Hochreiter, S. and Schmidhuber, J.: Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[Hu et al. , 2009]Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z.: Understanding user's query intent with wikipedia. In Proceedings of the 18th International Conference on World Wide Web, pages 471–480. ACM, 2009.

[Ji et al. , 2015]Ji, G., He, S., Xu, L., Liu, K., and Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 1, pages 687–696, 2015.

[Jiang et al. , 2017]Jiang, M., Shang, J., Cassidy, T., Ren, X., Kaplan, L. M., Hanratty, T. P., and Han, J.: Metapad: Meta pattern discovery from massive text cor-

pora. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 877–886. ACM, 2017.

[Kalchbrenner et al. , 2014]Kalchbrenner, N., Grefenstette, E., Blunsom, P., Kartsaklis, D., Kalchbrenner, N., Sadrzadeh, M., Kalchbrenner, N., Blunsom, P., Kalchbrenner, N., and Blunsom, P.: A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 212–217. Association for Computational Linguistics, 2014.

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

[Kingma and Welling, 2013]Kingma, D. P. and Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[Kingma and Welling, 2014]Kingma, D. P. and Welling, M.: Stochastic gradient vb and the variational auto-encoder. In Proceedings of the Second International Conference on Learning Representations, ICLR, 2014.

[Komninos and Manandhar, 2017]Komninos, A. and Manandhar, S.: Feature-rich networks for knowledge base completion. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 324–329, 2017.

[Lafferty et al. , 2001]Lafferty, J. D., McCallum, A., and Pereira, F. C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[Lai et al. , 2015]Lai, S., Xu, L., Liu, K., and Zhao, J.: Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-ninth AAAI conference on artificial intelligence, 2015.

[Legrand and Collobert, 2015]Legrand, J. and Collobert, R.: Joint rnn-based greedy parsing and word composition. In Proceedings of the Third International Conference on Learning Representations, 2015.

[Li et al. , 2016]Li, Y., Liu, C., Du, N., Fan, W., Li, Q., Gao, J., Zhang, C., and Wu, H.: Extracting medical knowledge from crowdsourced question answering website. IEEE Transactions on Big Data, 2016.

[Liao et al. , 2017]Liao, Z., Song, X., Shen, Y., Lee, S., Gao, J., and Liao, C.: Deep context modeling for web query entity disambiguation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 1757–1765. ACM, 2017.

[Liben-Nowell and Kleinberg, 2007]Liben-Nowell, D. and Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 58(7):1019–1031, 2007.

[Limsopatham et al. , 2013]Limsopatham, N., Macdonald, C., and Ounis, I.: Inferring conceptual relationships to improve medical records search. In Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, pages 1–8. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[Lin et al. , 2003]Lin, D., Zhao, S., Qin, L., and Zhou, M.: Identifying synonyms among distributionally similar words. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pages 1492–1493. Morgan Kaufmann Publishers Inc., 2003.

[Lin et al. , 2016]Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M.: Neural relation extraction with selective attention over instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2124–2133, 2016.

[Liu and Lane, 2016]Liu, B. and Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. Interspeech 2016, pages 685–689, 2016.

[Liu et al. , 2016]Liu, C., Sun, H., Du, N., Tan, S., Fei, H., Fan, W., Yang, T., Wu, H., Li, Y., and Zhang, C.: Augmented lstm framework to construct medical self-diagnosis android. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining, pages 251–260. IEEE, 2016.

[Liu et al. , 2017]Liu, L., Ren, X., Zhu, Q., Zhi, S., Gui, H., Ji, H., and Han, J.: Heterogeneous supervision for relation extraction: A representation learning approach. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 46–56, 2017.

[Liu et al. , 2015]Liu, Y., Chen, Y., Tang, J., and Liu, H.: Context-aware experience extraction from online health forums. In Proceedings of the 2015 International Conference on Healthcare Informatics, pages 42–47. IEEE, 2015.

[Ma et al. , 2019]Ma, F., Li, Y., Zhang, C., Gao, J., Du, N., and Fan, W.: Mcvae: Margin-based conditional variational autoencoder for relation classification and pattern generation. In Proceedings of the World Wide Web Conference, pages 3041–3048. ACM, 2019.

[Marcheggiani and Titov, 2016]Marcheggiani, D. and Titov, I.: Discrete-state variational autoencoders for joint discovery and factorization of relations. Transactions of the Association for Computational Linguistics, 4:231–244, 2016.

[Mikolov et al. , 2013a]Mikolov, T., Chen, K., Corrado, G., and Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[Mikolov et al. , 2013b]Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119, 2013.

[Miller, 1995]Miller, G. A.: Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41, 1995.

[Mintz et al. , 2009]Mintz, M., Bills, S., Snow, R., and Jurafsky, D.: Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1003–1011. Association for Computational Linguistics, 2009.

[Mueller and Thyagarajan, 2016]Mueller, J. and Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[Murphy, 2012]Murphy, K. P.: Machine learning: a probabilistic perspective. MIT press, 2012.

[Nair and Hinton, 2010]Nair, V. and Hinton, G. E.: Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, pages 807–814, 2010.

[Neculoiu et al. , 2016]Neculoiu, P., Versteegh, M., and Rotaru, M.: Learning text similarity with siamese recurrent networks. In Proceedings of the 1st Workshop on Representation Learning for NLP, pages 148–157, 2016.

[Nickel et al. , 2016]Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104(1):11–33, 2016.

[Pennington et al. , 2014]Pennington, J., Socher, R., and Manning, C.: Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 1532–1543, 2014.

[Peters et al. , 2018]Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L.: Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), volume 1, pages 2227–2237, 2018.

[Pu et al. , 2016]Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L.: Variational autoencoder for deep learning of images, labels and captions. In Advances in Neural Information Processing Systems, pages 2352–2360, 2016.

[Qu et al. , 2017]Qu, M., Ren, X., and Han, J.: Automatic synonym discovery with knowledge bases. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 997–1005. ACM, 2017.

[Radford et al. , 2015]Radford, A., Metz, L., and Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[Ren and Cheng, 2015]Ren, X. and Cheng, T.: Synonym discovery for structured entities on heterogeneous graphs. In Proceedings of the 24th International Conference on World Wide Web, pages 443–453. ACM, 2015.

[Roller et al. , 2014]Roller, S., Erk, K., and Boleda, G.: Inclusive yet selective: Supervised distributional hypernymy detection. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1025–1036, 2014.

[Sabour et al. , 2017]Sabour, S., Frosst, N., and Hinton, G. E.: Dynamic routing between capsules. In Advances in Neural Information Processing Systems, pages 3856–3866, 2017.

[Sahay et al. , 2008]Sahay, S., Mukherjea, S., Agichtein, E., Garcia, E. V., Navathe, S. B., and Ram, A.: Discovering semantic biomedical relations utilizing the web. ACM Transactions on Knowledge Discovery from Data, 2(1):3, 2008.

[Serban et al. , 2016]Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[Shen et al. , 2018]Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C.: Disan: Directional self-attention network for rnn/cnn-free language understanding. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[Snow et al. , 2005]Snow, R., Jurafsky, D., and Ng, A. Y.: Learning syntactic patterns for automatic hypernym discovery. In Advances in Neural Information Processing Systems, pages 1297–1304, 2005.

[Socher et al. , 2013]Socher, R., Chen, D., Manning, C. D., and Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In Advances in Neural Information Processing Systems, pages 926–934, 2013.

[Sohn et al. , 2015]Sohn, K., Lee, H., and Yan, X.: Learning structured output representation using deep conditional generative models. In Advances in Neural Information Processing Systems, pages 3483–3491, 2015.

[Sønderby et al. , 2016]Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O.: How to train deep variational autoencoders and probabilistic ladder networks. In Proceedings of the 33rd International Conference on Machine Learning, 2016.

[Speer and Havasi, 2012]Speer, R. and Havasi, C.: Representing general relational knowledge in conceptnet 5. In Proceedings of the Eighth International Conference on Language Resources and Evaluation, pages 3679–3686, 2012.

[Spink et al. , 2004]Spink, A., Yang, Y., Jansen, J., Nykanen, P., Lorence, D. P., Ozmutlu, S., and Ozmutlu, H. C.: A study of medical and health queries to web search engines. Health Information & Libraries Journal, 21(1):44–51, 2004.

[Stanton et al. , 2014]Stanton, I., Ieong, S., and Mishra, N.: Circumlocution in diagnostic medical queries. In Proceedings of the 37th International Conference on Research & Development in Information Retrieval, pages 133–142. ACM, 2014.

[Sun and Grishman, 2010]Sun, A. and Grishman, R.: Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 1194–1202. Association for Computational Linguistics, 2010.

[Sun et al. , 2012]Sun, Y., Han, J., Aggarwal, C. C., and Chawla, N. V.: When will it happen?: relationship prediction in heterogeneous information networks. In Proceedings of the Fifth

ACM International Conference on Web Search and Data Mining, pages 663–672. ACM, 2012.

[Sutskever et al. , 2014]Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.

[Syed et al. , 2010]Syed, Z., Viegas, E., Parastatidis, S., et al.: Automatic discovery of semantic relations using mindnet. In Proceedings of the Seventh International Conference on Language Resources and Evaluation, 2010.

[Tan et al. , 2015]Tan, M., Santos, C. d., Xiang, B., and Zhou, B.: Lstm-based deep learning models for non-factoid answer selection. arXiv preprint arXiv:1511.04108, 2015.

[Tan et al. , 2018]Tan, Z., Wang, M., Xie, J., Chen, Y., and Shi, X.: Deep semantic role labeling with self-attention. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[Tieleman and Hinton, 2012]Tieleman, T. and Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 2012.

[Trouillon et al. , 2017]Trouillon, T., Dance, C. R., Gaussier, É., Welbl, J., Riedel, S., and Bouchard, G.: Knowledge graph completion via complex tensor factorization. The Journal of Machine Learning Research, 18(1):4735–4772, 2017.

[Tsoumakas et al. , 2009]Tsoumakas, G., Katakis, I., and Vlahavas, I.: Mining multi-label data. In Data Mining and Knowledge Discovery Handbook, pages 667–685. Springer, 2009.

[Tur et al. , 2010]Tur, G., Hakkani-Tür, D., and Heck, L.: What is left to be understood in atis? In Spoken Language Technology Workshop (SLT), 2010 IEEE, pages 19–24. IEEE, 2010.

[Verga et al. , 2017]Verga, P., Neelakantan, A., and McCallum, A.: Generalizing to unseen entities and entity pairs with row-less universal schema. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 613–622, 2017.

[Wang et al. , 2015a]Wang, C., Song, Y., Roth, D., Wang, C., Han, J., Ji, H., and Zhang, M.: Constrained information-theoretic tripartite graph clustering to identify semantically similar re-

lations. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[Wang et al. , 2015b]Wang, Q., Wang, B., and Guo, L.: Knowledge base completion using embeddings and rules. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[Wang and Jiang, 2016]Wang, S. and Jiang, J.: A compare-aggregate model for matching text sequences. arXiv preprint arXiv:1611.01747, 2016.

[Wang et al. , 2014]Wang, Z., Zhang, J., Feng, J., and Chen, Z.: Knowledge graph and text jointly embedding. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 1591–1601, 2014.

[Wang et al. , 2017]Wang, Z., Hamza, W., and Florian, R.: Bilateral multi-perspective matching for natural language sentences. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pages 4144–4150, 2017.

[Wang et al. , 2016]Wang, Z., Mi, H., Hamza, W., and Florian, R.: Multi-perspective context matching for machine comprehension. arXiv preprint arXiv:1612.04211, 2016.

[Wei et al. , 2009]Wei, X., Peng, F., Tseng, H., Lu, Y., and Dumoulin, B.: Context sensitive synonym discovery for web search queries. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, pages 1585–1588. ACM, 2009.

[Wieting et al. , 2016]Wieting, J., Bansal, M., Gimpel, K., and Livescu, K.: Charagram: Embedding words and sentences via character n-grams. arXiv preprint arXiv:1607.02789, 2016.

[Xia et al. , 2018]Xia, C., Zhang, C., Yan, X., Chang, Y., and Yu, P.: Zero-shot user intent detection via capsule neural networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3090–3099, 2018.

[Xu and Sarikaya, 2013]Xu, P. and Sarikaya, R.: Convolutional neural network based triangular crf for joint intent detection and slot filling. In 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pages 78–83. IEEE, 2013.

[Xu et al. , 2017]Xu, W., Sun, H., Deng, C., and Tan, Y.: Variational autoencoder for semi-supervised text classification. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[Yan et al. , 2009]Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M.: Unsupervised relation extraction by mining wikipedia texts using information from the web. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1021–1029. Association for Computational Linguistics, 2009.

[Yao et al. , 2011]Yao, L., Haghighi, A., Riedel, S., and McCallum, A.: Structured relation discovery using generative models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1456–1466. Association for Computational Linguistics, 2011.

[Zeiler, 2012]Zeiler, M. D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.

[Zeng et al. , 2014]Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J.: Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2335–2344, 2014.

[Zhang et al. , 2017]Zhang, C., Du, N., Fan, W., Li, Y., Lu, C.-T., and Philip, S. Y.: Bringing semantic structures to user intent detection in online medical queries. In Proceedings of the 2017 IEEE International Conference on Big Data, pages 1019–1026. IEEE, 2017.

[Zhang et al. , 2016]Zhang, C., Fan, W., Du, N., and Yu, P. S.: Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In Proceedings of the 25th International Conference on World Wide Web, pages 1373–1384. International World Wide Web Conferences Steering Committee, 2016.

[Zhang et al. , 2018a]Zhang, C., Li, Y., Du, N., Fan, W., and Yu, P. S.: On the generative discovery of structured medical knowledge. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2720–2728. ACM, 2018.

[Zhang et al. , 2018b]Zhang, C., Li, Y., Du, N., Fan, W., and Yu, P. S.: Synonymnet: Multi-context bilateral matching for entity synonyms. arXiv preprint arXiv:1901.00056, 2018.

[Zhang et al. , 2019]Zhang, C., Li, Y., Du, N., Fan, W., and Yu, P. S.: Joint slot filling and intent detection via capsule neural networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.

[Zhang et al. , 2003]Zhang, H.-P., Yu, H.-K., Xiong, D.-Y., and Liu, Q.: Hhmm-based chinese lexical analyzer ictclas. In SIGHAN, pages 184–187. Association for Computational Linguistics, 2003.

[Zhang et al. , 2019]Zhang, J., Zhang, C., Dong, B., Yang, Y., and Yu, P. S.: Missing movie synergistic completion across multiple isomeric online movie knowledge libraries. In Proceedings of the 2019 International Joint Conference on Neural Networks, 2019.

[Zhang et al. , 2016]Zhang, J., Lu, C.-T., Zhou, M., Xie, S., Chang, Y., and Philip, S. Y.: Heer: Heterogeneous graph embedding for emerging relation detection from news. In Proceedings of the 2016 IEEE International Conference on Big Data, pages 803–812. IEEE, 2016.

[Zhao et al. , 2018]Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S., and Zhao, Z.: Investigating capsule networks with dynamic routing for text classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3110–3119, 2018.

VITA

Name: Chenwei Zhang

EDUCATION:

B.Eng. in Computer Science and Technology, Southwest University, 2014.

PUBLICATIONS:

- Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, Philip S. Yu. *Joint Slot Filling and Intent Detection via Capsule Neural Networks.* In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.
- Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma and Philip Yu. *Multi-grained Named Entity Recognition.* In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.
- Jiawei Zhang, Chenwei Zhang, Bowen Dong, Yang Yang, Philip S. Yu. *Missing Movie Synergistic Completion across Multiple Isomeric Online Movie Knowledge Libraries.* In Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2019.
- Fenglong Ma, Yaliang Li, Chenwei Zhang, Jing Gao, Nan Du, Wei Fan. *MCVAE: Margin-based Conditional Variational Autoencoder for Relation Classification and Pattern Generation.* In Proceedings of the 2019 World Wide Web Conference (WWW), 2019.
- Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, Philip S. Yu. *On the Generative Discovery of Structured Medical Knowledge.* In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2018.

- Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, Philip S. Yu. *SynonymNet: Multi-context Bilateral Matching for Entity Synonyms.* arXiv, 2018.
- Congying Xia*, Chenwei Zhang*, Xiaohui Yan, Yi Chang, Philip S. Yu. *Zero-shot User Intent Detection via Capsule Neural Networks.* In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018. (* equally contributed)
- Shaika Chowdhury, Chenwei Zhang, Philip S. Yu. *Multi-Task Pharmacovigilance Mining from Social Media Posts.* In Proceedings of the 27th edition of The Web Conference (WWW), 2018.
- Zhang-Meng Liu, Chenwei Zhang, Philip S. Yu. *Direction-of-Arrival Estimation based on Deep Neural Networks with Robustness to Array Imperfections.* In the IEEE Transactions on Antennas and Propagation, 2018.
- Yue Wang, Chenwei Zhang, Shen Wang, Philip S. Yu, Lu Bai, Lixin Cui. *Market Abnormality Period Detection via Co-movement Attention Model.* In Proceedings of the IEEE International Conference on Big Data (Big Data), 2018.
- Ye Liu, Jiawei Zhang, Chenwei Zhang, Philip S. Yu. *Data-driven Blockbuster Planning on Online Movie Knowledge Library.* In Proceedings of the IEEE International Conference on Big Data (Big Data), 2018.
- Yue Wang, Chenwei Zhang, Shen Wang, Philip S. Yu, Lu Bai, Lixin Cui. *Deep Co-investment Network Learning for Financial Assets.* arXiv, 2018.

- Yaliang Li, Liuyi Yao, Nan Du, Jing Gao, Qi Li, Chuishi Meng, Chenwei Zhang, Wei Fan. *Finding Similar Medical Questions from Question Answering Websites.* arXiv, 2018.
- Chenwei Zhang, Wei Fan, Nan Du, Yaliang Li, Chun-Ta Lu, and Philip S. Yu. *Bringing Semantic Structures to User Intent Detection in Online Medical Queries.* In Proceedings of the IEEE International Conference on Big Data (Big Data), 2017.
- Bokai Cao, Lei Zheng, Chenwei Zhang, Philip S. Yu, Andrea Piscitello, John Zulueta, Olu Ajilore, Kelly Ryan and Alex Leow. *DeepMood: Modeling Mobile Phone Typing Dynamics for Mood Detection.* In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2017.
- Jiawei Zhang, Congying Xia, Chenwei Zhang, Limeng Cui, Yanjie Fu, and Philip S Yu. *BL-MNE: Emerging Heterogeneous Social Network Embedding through Broad Learning with Aligned Autoencoder.* In Proceeding of the IEEE International Conference on Data Mining (ICDM), 2017.
- Junxing Zhu, Jiawei Zhang, Lifang He, Quanyuan Wu, Bin Zhou, Chenwei Zhang and Philip S. Yu. *Broad Learning based Multi-Source Collaborative Recommendation.* In Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM), 2017.
- Junxing Zhu, Jiawei Zhang, Chenwei Zhang, Quanyuan Wu, Yan Jia, Bin Zhou, and Philip S. Yu. *CHRS: Cold Start Recommendation across Multiple Heterogeneous Information Networks.* IEEE Access (2017).

- Chenwei Zhang, Wei Fan, Nan Du and Philip S. Yu. *Mining User Intentions from Medical Queries: A Neural Network Based Heterogeneous Jointly Modeling Approach.* In Proceedings of the 25th International World Wide Web Conference (WWW), 2016.
- Chenwei Zhang, Sihong Xie, Yaliang Li, Jing Gao, Wei Fan and Philip S. Yu. *Multi-source Hierarchical Prediction Consolidation.* In Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM), 2016.
- Chaochun Liu, Huan Sun, Nan Du, Shulong Tan, Hongliang Fei, Wei Fan, Tao Yang, Hao Wu, Yaliang Li, and Chenwei Zhang. *An Augmented LSTM Framework to Construct Medical Self-diagnosis Android.* In Proceeding of the IEEE International Conference on Data Mining (ICDM), 2016.
- Yaliang Li, Chaochun Liu, Nan Du, Wei Fan, Qi Li, Jing Gao, Chenwei Zhang, and Hao Wu. *Extracting Medical Knowledge from Crowdsourced Question Answering Website.* IEEE Transactions on Big Data (2016).
- Chenwei Zhang, Xiaoyan Su, Yong Hu, Zili Zhang, Yong Deng. *An Evidential Spam-Filtering Framework.* Cybernetics and Systems (2016): 1-18.
- Hongping Wang, Xi Lu, Yuxian Du, Chenwei Zhang, Rehan Sadiq, Yong Deng. *Fault Tree Analysis Based on TOPSIS and Triangular Fuzzy Number.* International Journal of System Assurance Engineering and Management (2014): 1-7.

- Chenwei Zhang, Yong Hu, Felix T. S. Chan, Rehan Sadiq and Yong Deng. *A New Method to Determine Basic Probability Assignment Using Core Samples.* Knowledge-Based Systems (2014): 140-149.