Office of Graduate Studies

Dissertation / Thesis Approval Form

This form is for use by all doctoral and master's students with a dissertation/thesis requirement. Please print clearly as the library will bind a copy of this form with each copy of the dissertation/thesis. All doctoral dissertations must conform to university format requirements, which is the responsibility of the student and supervising professor. Students should obtain a copy of the Thesis Manual located on the library website.

Dissertation/Thesis Title: Studies on User Intent Analysis and Mining

Author: Yue Shang

This dissertation/thesis is hereby accepted and approved.

Signatures:

Examining Committee

Chair _____

Members _____

_____

_____

_____

_____

Academic Advisor _____

Department Head _____

Office of Graduate Studies • 3141 Chestnut St. • Randell Hall 240 • Philadelphia, PA 19104
Tel.: 215-895-0366 • **Fax**: 215-895-0495 • **Email**: graduate@drexel.edu • **Web**: www.drexel.edu/provost/graduatestudies

Studies on User Intent Analysis and Mining

A Thesis

Submitted to the Faculty

of

Drexel University

by

Yue Shang

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2017

Dedications

I dedicate this thesis to my parents and family

whose endless love encourage me to finish this work.

Acknowledgments

First, I must sincerely thank my advisor, Professor Xiaohua Tony Hu, for his guidance and inspiration throughout my study in Drexel. I deeply appreciate Prof. Hu for everything I learned from him and for giving me plenty of advice on both research and life. Prof. Hu also provide me a flexible environment for exploring different areas, and always support me with valuable resources. Without his guidance, this dissertation would have been impossible.

I also want to thank my co-advisor Dr. Yuan An. Dr. An gives me important research guidance during my PhD study. He taught me how to find the research topic, how to perform the study, and directed me to write papers.

I would like to express my sincere gratitude to my committee members, Dr. Weimao Ke, Dr. Erjia Yan and Dr. Li Sheng. Their advice and suggestions on this dissertation are insightful and valuable.

I would like to express my thankfulness to researchers at TCL research America: Dr. Haohong Wang, Xiaobo Ren, Zhi Zhang, Ziju Feng, Yang Li, for their support and help. And it has been a precious experience during my entire Ph.D. study. I'd like to especially thank Dr. Wang for research guidance and inspiring ideas. I also want to thank the researchers in Huawei Inc., and my mentor, Jihong Ma, for her patient guidance during my internship.

I want to express my thanks to colleagues and friends for their support and help: Wanying Ding, Xiaoli Song, Mengwen Liu, Yuan Ling, Lifan Guo, Qing Ping, Ni An, Chen Chen, Yetian Fan, Yizhou Zang, Bo Song, and all the friends for the supports.

Finally, I want to thank my family, for their unconditional love, supporting me to achieve this work and move on.

Table of Contents

List of Tables

List of Figures

Abstract

Studies on User Intent Analysis and Mining
Yue Shang

Predicting the goals of users can be extremely useful in e-commerce, online entertainment, information retrieval, and many other online services and applications. In this thesis, we study the task of user intent understanding, trying to bridge the gap between user expressions to online services and their goals behind it.

As far as we know, most of the existing user intent studies are focusing on web search and social media domain. Studies on other areas are not enough. For example, as people more and more rely our daily life on cellphone, our information needs expressing to mobile devices and related services are increasing dramatically. Studies of user intent mining on mobile devices are not much. And the intentions of using mobile devices are different from the ones we use web search engine or social network. So we cannot directly apply the existing user intention to this area. Besides, user's intents are not stable but changing over time. And different interests will impact each other. Modeling such kind of dynamic user interests can help accurately understand and predict user's intent. But there're few existing works in this area. Moreover, user intent could be explicitly or implicitly expressed by users. The implicit intent expression is more close to human's natural language and also have great value to recognize and mine.

To make further studies of these challenges, we first try to answer the question of "*What is the user intent*?". By referring amount of previous studies, we give our definition of user intent as "User intent is a task-specific, predefined or latent concept, topic or knowledge-base that is under an expression from a user who is trying to express his goal of information or service need".

Then, we focus on the driving scenario when a user using cellphone and study the user intent in this domain. As far as we know, it is the first time of user intent analysis and categorization in this domain. And we also build a dataset of user input and related intent category and attributes by crowdsourcing and carefully handcraft. With the user intent taxonomy and dataset in hand, we conduct a user intent classification and user intent attribute recognition by supervised machine learning models. To classify the user intent for a user intent query, we use a convolutional neural network model to build a multi-class classifier. And then we use a sequential labeling method to recognize the intent attribute in the query. The experiment results show that our proposed method outperforms several baseline models in precision, recall, and F-score.

In addition, we study the implicit user intent mining method through web search log data. By using a Restricted Boltzmann Machine, we make use of the correlation of query and click information to learn the latent intent behind a user web search.

We propose a user intent prediction model on online discussion forum using Multivariate Hawkes Process. It dynamically models user intentions change and interact over time.The method models both of the internal and external factors of user's online forum response motivations, and also integrated the time decay fact of user's interests.

We also present a data visualization method, using an enriched domain ontology to highlight the domain-specific words and entity relations within an article.

Chapter 1: Introduction

1.1 Background

User intent understanding has always been a principal problem for many applications, such as information retrieval, text mining, and recommender system. Nowadays, online services have become indispensable parts of modern human life. In particular, the rise and popularity of smartphones make people more involved in different online service platforms to meet needs of daily life. With the explosive growth of information, modeling user intent to meet individual user needs is essential. From the user's perspective, understanding user intent could improve the recommendation, personalized search to provide better user experiences. And from the platform's perspective, a better understanding of user intent could provide accurate products, services to users, so as potentially improve the page view and gross merchandise volume.

However, first of all, what is user intent? We need some ideas about this concept. Here are some examples. A query such as "Sony MDR 1000x" may have an intent of checking price information of a product, or looking for ways to purchase it. The intent behind "shuffle the track" is about control music playing. User intention is not merely a concept or category. It may also contain attributes. For example, one user may talk to Siri that "go to the nearest Starbucks." This message may express the user's intent about "navigation", and it also has an attribute "Navigation Destination", the value of which is the"Starbucks" closest to the current location.

Most of the definitions about user intent came from the earlier studies in web search and information retrieval. Jansen[10] defined user intent of web search as "User intent is the resource specified by the affective, cognitive, or situational goal expressed in an interaction with a Web search engine". That is to say, unlike goals, user intent is how the goals are expressed. Referring to Belkin's states of a searching episode [11], the intent is akin to goal, and expression akin to method of interaction. Unlike goals, however, intent is concerned with how the goal is expressed because the expression determines what type of resource the user desires to address his or her overall goal.

To better scope the problem, studies have indicated how to characterize and classify user intent. In most of the recent studies, user intents could be represented as a set of keyword vectors, a semantic class vectors, a set of concepts, or an instance of predefined ontology or knowledge bases[12]. So here we come to a definition of user intent in this thesis. *User intent is a task-specific, predefined or latent concept, topic or knowledge-base that is under an expression from a user who is trying to express his goal of information or service need.*

User intent understanding and modeling is a challenging task.

1. User intent is not always explicitly expressed. For example, on social media, people may discuss various topics such as what they do, how they feel and where they are. And often there're some intents behind these expressions. "I lose my cellphone" may indicate to purchase a new one. The intention behind "I'm so hungry" may be to find a restaurant or food delivery. So unlike ad-hoc information retrieval, there're many other scenarios where people express their intent in an implicit way. And we need to find a way to bridge the gap between user expression and their target services.

2. User intent is something like knowledge graph, or ontology, which is not always available for many domains. Build such kind of knowledge graph is necessary for modeling user intention.

3. User intent is changing. Different people have very disparate interests. And even for one individual, his/her interest is always changing over time. Suppose Andy is a Reddit user. He is a super sci-fi fan who never misses any super-hero movies; he like traveling with family, and he also has a dog. According to Andy's Reddit timeline(records for all his activities on the platform with timestamps), sometimes he thumbs up other's pet photos. When new Marvel movie is released, he will discuss the hero stories with others. Thus, Andy's interests may contain sci-fi, pet, and travel. But proportions of the three categories are not always the same. When "Captain American" released, his interests in sci-fi is greatly increased. Moreover, the interest persistent over different topics is also different. User's attention to a piece of news or event may vanish shortly after several days. But for a book fan of "Harry Potter", continuously

following the related novels, games, movies, and events for years are highly possible. All the conditions make the user interests modeling and understanding a tough task.

1.2 Research Questions

User intent mining is to use terms, topics, concepts, or other representation methods, explicitly or implicitly, modeling the user's preferences. It's helpful to better understand user behaviors and provides useful guidance to design user-centric applications, such as personalized information retrieval, recommendation, user profile construction, etc.

According to our previous discussions, we will raise our research questions as follows.

1. How to model user intent in a specific domain?
   (a) How to define the user intent in a specific domain?
      - We choose a scenario in mobile usage, and analyze the user intent in this domain, such as "Navigation","Find Parking", etc.
      - We also define entity types in each domain. "Location", "Parking Type" are entities for "Find Parking".
   (b) Given a user intent information, how to leverage it for information extraction, or intent mining?
      - We build a dataset with user intent labeling.
      - We conduct a supervised machine learning method to classify the user input to corresponding intention category, and recognize the entities and attributes if exist using sequential labeling models.
2. For the implicit user intent, how to model it?
   - We try to analyze user intent in web query.
   - We conducted an unsupervised model to learn the underlying intent according query text and corresponding clicked url.
   - Implicit user intents are modeled as latent variables. We will discuss this in Chapter 4.

3. How to model the temporal dynamic user intent?

   (a) How to represent the temporal dynamic user interests?

   - User interests' distribution is different at the different time.
   - Each interest could be regarded as a Hawkes Process. Hawkes Process is an arrival model, which is suitable to predict if the user will be likely to do an event of this category in the future.

   (b) How to model the long-term and short-term user interests?

   - There're long term and short term interests exist for one user. The long-term interests are the user's continuous preferences and may be related to his habits or personalities. The short-term interests are more likely to be interests excited by external factors, such as top news, popular events, etc.
   - Long-term interests are captured by Hawkes process base intensity. Short-term interests are modeled by the exciting function. A short-term interest can be regarded as an interest that has a large exciting impact to the current time.

   (c) How to model the interests' decaying speed difference?

   - Time-forgetting mechanism states that user interests will vanish as time passing by without stimulation. Thus, it's necessary to model the time-decaying feature within the user interests model.
   - We use a log-normal kernel to describe the interest's decaying feature of interest.

1.3 Contributions

In this thesis, we try to solve a user intent mining within a vertical domain scenario: using the smartphone while driving. As far as we known, this is the first time to thoroughly study the user intent on this domain. We define the possible intent category in this scenario and illustrate all the possible attribute types. The intent framework could facilitate many mobile applications, such as mobile audio assistance, user profiling. With a built dataset, we proposed a user intent classification model to identify the intention of a user's input. And we also applied a named entity recognizer to

identify the possible attributes within it. Our model outperforms several baseline methods. This solution pipeline could also be applied to other domain.

Also, we study the implicit user intent mining method through web search log data. By using a Restricted Boltzmann Machine, we make use of the correlation of query and click information to learn the latent intent behind a user web search.

We propose a user intent prediction model on online discussion forum using Multivariate Hawkes Process. It dynamically models user intentions change and interact over time.The method models both of the internal and external factors of user's online forum response motivations, and also integrated the time decay fact of user's interests.

1.4 Thesis Organization

The rest of this thesis is organized as following parts. Chapter 2 introduces the previous studies about the concept of user intent, user intent mining in different domains, user intent tasks, and the applications for user intent mining. In Chapter 3, we choose a use scenario and firstly try to address the question of **How to represent the user intent?** by defining the user intent and related attributes in that domain. And then we address the questions of **How to make use of such information to better meet users' information need?**. In Chapter 4, we try to discuss **modeling users' implicit intent** by analyzing click-through dataset. In Chapter 5, we address the question of **How to represent the temporal dynamic user interests**. And in Chapter 6, we try to visualize the concept, key phrases in an article as a form of user intent by presenting an ontology extension method. Finally, we conclude this thesis and introduce future directions in Chapter 7.

Chapter 2: Related Works

User intent mining is an area received many research attention for a long time. But most of the previous works are focusing on user intent understanding in web search domain. In our work, user intent understanding and mining will include user intent classification and user intent attribute detection. These works are closely related to several natural language processing and machine learning domain, such as short text classification, named entity recognition.

In this chapter, we will first study the existing user intent definitions, applications in various domains in previous studies. Then we will focus on applications and methods for user intent mining.

2.1 User Intent Definitions

User intent is the resource specified by the affective, cognitive, or situational goal expressed in an interaction with a Web search engine. Referring to Belkin's states of a searching episode [11], the intent is akin to goal, and expression akin to method of interaction. Unlike goals, intent is concerned with how the goal is expressed because the expression determines what type of resource the user desires to address his or her overall goal. User intent is a broad concept, and have been studied in many domains. We will first review the previous research about user intent in different domains.

In most applications, the user intent is task-specific and could facilitate further information mining, such as named entity extraction, in the back-end system. An example of this could be "I want to fly from Seattle to Miami tomorrow morning". The intent of this user input could be "FindFlight" and there're several task-specific slots, such as "DestinationLocation", "DepartureLocation" and "DepartureDate".

Hollerit et al.[13] studied tweets with commercial intent. They first give a definition about a tweet containing "commercial intent" as "A tweet contains at least one verb and describes the user's intention to commit a commercial activity in a recognizable way". And they also notice another nature of intent, which is, whether a tweet's intent is explicit or implicit. Explicit vs. Implicit: The

tweet "Facing Repossession, Let us buy your house for cash now" explicitly expresses the intent to buy a house. In contrast, the tweet "Debating on buying the pair of 80s cop shades..." contains to a certain extent commercial intent, but it is not explicitly stated rather as a possibility in the future. And they notice that implicit commercial intent also has commercial value.

Wang et al.[3] proposed a semi-supervised user intent classification task on Twitter data. They first give a clear definition of "intent tweet", "intent-indicator" and "intent-keyword". They define a tweet as an **intent tweet** if (1) it contains at least one verb and (2) explicitly describes the user's intent to perform an activity (3) in a recognizable way. And they also define "intent-indicator" and "intent-keyword". **Intent-Indicator**: It comprises a group of terms that are used by users to express their intents. It is a verb or infinitive phrase that immediately follows a subject word, e.g., "I". For example, in tweet "I want to buy an xbox", "want to" is an intent-indicator, indicating the tweet is likely to be an intent tweet. **Intent-Keyword**: It is a noun, verb, multi-word verb or compound noun (consisting of several nouns) contained in a verb or noun phrase which immediately follows an intent- indictor, e.g., in the previous example, "buy" and "xbox" which are contained in the phrase "buy an xbox" are intent-keywords.

2.2 User Intent in Various Domains

The user intent on the different application may vary. In most of the recent studies, user intent could be regarded as the type of user's information need, represented as an expression such as class labels, or tags.

Much of research in user intent detection has focused on understanding the intent of search queries. The general intent of user's web search could be navigational, informational and transactional[1]. However, understanding the intention of a search query is very different from user intention for content creation. In a survey of bloggers, findings of Nardi et al.[14] indicate that blogs are used as a tool to share daily experiences, opinions and commentary. And studies[2] of twitter and microblogs shows that the main types of user intentions are: daily chatter, conversations, sharing information and reporting news. So let's get some ideas about user intent mining in the prospectives from web search, social network, and mobile.

Table 2.1: Web Search Intent defined by Broder et al. 2002 [1]

Intent Type	Description
Navigational.	The immediate intent is to reach a particular site.
Informational	The intent is to acquire some information assumed to be present on one or more web pages.
Transactional	The intent is to perform some web-mediated activity.

2.2.1 Web Search

In the domain of user intent mining in Information retrieval, characterizing and identifying the intent of user queries is one of the most important challenges of modern information retrieval systems [15]. The first classification that is found in the literature was done by Broder [1], who proposed the taxonomy of user's intent as *navigational*, *informational* and *transactional*. He claimed that in the web context the "need behind the query" is often not informational. The purpose of "Navigational queries" is to reach a particular site that already in mind. Purpose of "Information queries" is to find information assumed to be available on the webpage. And "Transactional queries" are search intent for resources for further actions, such as downloading, or stream playing. Broder made a classification of queries through a user survey and manual classification of a query log.

This work was later taken up by Rose and Levinson [5], who developed a framework for manual classification of search goals by extending the classes proposed by Broder.

Chapelle et al. [16] extract a dataset from the logs of Yahoo! search by randomly sampled 2,492 user sessions. There are 3,658 queries and 18,296 documents in this data set. For each user session, a professional editor examined the user activity in that session and judged the user intents. Their work focused on shopping related queries. And there're five shopping intents considered by this work: **buying guide, reviews, support, official product page, and shopping site/purchase**. In their scope, the consumer first examines buying guides to understand how to shop for the desired product, then consults reviews and goes to the official product homepage. Finally, the consumer makes their purchase at a shopping site, and uses support pages for post-purchase information.

Pantel[9] conducted a query log entity mining through user intent understanding. To evaluate the model, they generate a training dataset by automatically align some selected freebase entity

SEARCH GOAL	DESCRIPTION	EXAMPLES
1. Navigational	My goal is to go to specific known website that I already have in mind. The only reason I'm searching is that it's more convenient than typing the URL, or perhaps I don't know the URL.	aloha airlines duke university hospital kelly blue book
2. Informational	My goal is to learn something by reading or viewing web pages	
2.1 Directed	I want to learn something in particular about my topic	
2.1.1 Closed	I want to get an answer to a question that has a single, unambiguous answer.	what is a supercharger 2004 election dates
2.1.2 Open	I want to get an answer to an open-ended question, or one with unconstrained depth.	baseball death and injury why are metals shiny
2.2 Undirected	I want to learn anything/everything about my topic. A query for topic X might be interpreted as "tell me about X."	color blindness jfk jr
2.3 Advice	I want to get advice, ideas, suggestions, or instructions.	help quitting smoking walking with weights
2.4 Locate	My goal is to find out whether/where some real world service or product can be obtained	pella windows phone card
2.5 List	My goal is to get a list of plausible suggested web sites (I.e. the search result list itself), each of which might be candidates for helping me achieve some underlying, unspecified goal	travel amsterdam universities florida newspapers
3. Resource	My goal is to obtain a resource (not information) available on web pages	
3.1 Download	My goal is to download a resource that must be on my computer or other device to be useful	kazaa lite mame roms
3.2 Entertainment	My goal is to be entertained simply by viewing items available on the result page	xxx porno movie free live camera in l.a.
3.3 Interact	My goal is to interact with a resource using another program/service available on the web site I find	weather measure converter
3.4 Obtain	My goal is to obtain a resource that does not require a computer to use. I may print it out, but I can also just look at it on the screen. I'm not obtaining it to learn some information, but because I want to use the resource itself.	free jack o lantern patterns ellis island lesson plans house document no. 587

Figure 2.1: Web Search User Intent expanded by Rose et al 2004.[5]

types to the queries. And their testset is created through carefully human labeling.

However, such a simple model is often inadequate for capturing the complexity of information seeking in the real world, and search engines need a better understanding of user intent and its multi-dimensional nature. Hence, with the aim to provide a more comprehensive understanding of the user and his/her intents, facets have the most representative abilities are selected. The studied facets are: *genre, objective, specificity, scope, topic, task, authority sensitivity, spatial sensitivity and time sensitivity* [17].

Community Question Answering (CQA) services, such as Yahoo! Answers, are specifically designed to address the innate limitation of Web search engines by helping users obtain information

Table 2.2: Main user intentions on Twitter defined by Java et al. 2007[2]

Intent Type	Description
Daily Chatter	Tweets talk about daily routine or what people are currently doing. This is the largest and most common user of Twitter
Conversations	People comment or reply to their friend's posts
Sharing information/URLs	About 13% of all the posts in the collection contain some URL in them.
Reporting news	Many users report latest news or comment about current events on Twitter.

from a community. Understanding the user intent of questions would enable a CQA system to identify similar questions, find relevant answers, and recommend potential answerers more effectively and efficiently. Chen[18] studied user intent of questions in CQA(community question answering) domain. They classified the questions into three categories: **subjective, objective, and social**. Subjective questions are to get personal opinions or general advice about something. Objective questions are looking for factual knowledge about something. While social questions' intent is to have social interactions with other users. The dataset consists of 1,539 questions that are randomly selected from the original Yahoo! Answers dataset and manually labeled according to their user intent. Feng et al. categories the questions into vertical domains, such as Weather, Restaurants, and Maps, hence better organizing the knowledge base and providing more accurate answers[19].

2.2.2 Social Networks

Posting short messages through social network services (e.g., Facebook, Twitter) has become an indispensable part of the daily life for many users. Through online activities such as chatting with friends and posting short status updates, online social networks have become major platforms where users discuss their needs and desire [2, 3].

Java et al. [2] conducted a earlier work for user intent analysis in twitter by studies on user and community levels. They found some main user intentions on Twitter as shown in Table 2.2.

Several previous works focus on extracting and inferring user's commercial intents from twitter[13, 3, 20]. Hollerit and Kröll[13] conduct a binary commercial intent classification to identify tweets that containing explicit and implicit commercial intent. They discover several discriminative patterns users may use when express purchase or selling intent. Works from Zhao et al.[20] focusing

Table 2.3: User Intent on Twitter in "commerce marketing" context defined by Wang et al. 2015 [3]

Intent Type	Description
Food & Drink	the tweet authors plan to have some food or drink
Travel	the tweet authors are interested in visiting some specific points of interests or places.
Career & Education	the tweet authors want to get a job, get a degree or do something for self-realization.
Goods & Services	the tweet authors are interested in or want to have some non-food/non-drink goods (e.g., car) or services (e.g., haircut)
Event & Activities	the tweet authors want to participate in some activities which do not belong to the aforementioned categories (e.g., concert).
Trifle intent	the tweets talks about daily routine, or some mood trifles.

on capture users' explicit purchase intent from tweets to facilitate product recommendation for e-commerce. They first generate list of candidate tweets buy some seed words such as "buy", "purchase", "on sale". And then, the tweets are classified using textual features from tweets as well as user's demographic features. Wang et al. [3] proposed a work categorized user intent in a more broad context: "commerce marketing" on Twitter. First of all, they define six types of intent on Twitter based on a large number of tweets and studied the taxonomy of Groupon. The categories are defined as follows:

Shen et al.[6] tried to link named entities in tweets with the knowledge base. It's a named entity linking task, however, is greatly helpful for the task of content recommendation, and user profiling. Figure 2.2 illustrates an example about named entity based user interests in tweet mining.

2.2.3 Mobile

As the smartphones becoming more and more popular ways of information access, mobile search has become a popular way to locate content on the Internet. There have been a number of studies to date that examine mobile search behavior[21, 22, 23, 24]. Web search on mobile is different from laptop, not just because of the devices but also because people's information needs also different. Mobile users, on the move, are likely to be interested in locating different types of content, for example. When users are using cellphones, their location and temporal information is available and their information needs are mostly related to these information[25]. Traditional Web intent

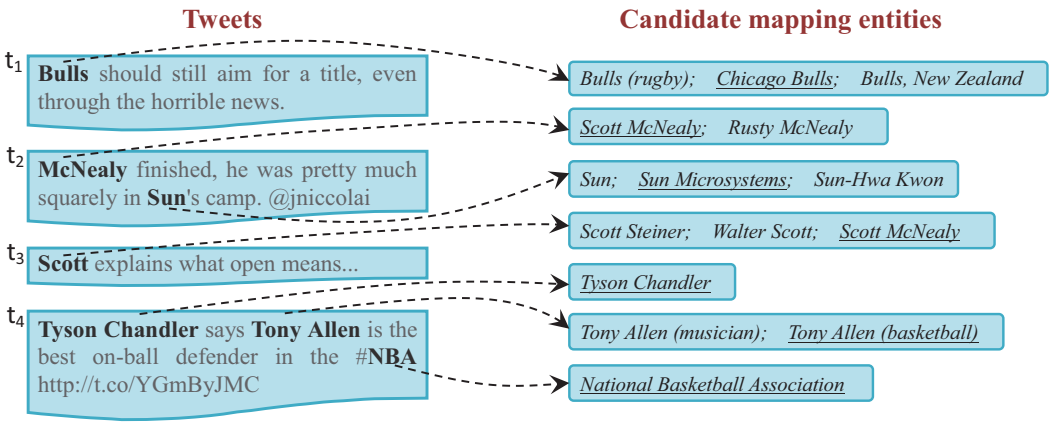


Figure 2.2: An illustration of the tweet entity linking task. Named entity mentions detected in tweets are in bold; candidate mapping entities for each entity mention are ranked by their prior probabilities in decreasing order; true mapping entities are underlined.[6]

taxonomies such as navigational and transactional needs were non-existent among the diary entries, thus requiring the addition of two new taxonomies that capture the unique constraints of mobility[25].

Work by Taylor et al. [4] focuses on motivations of users' information access on the Mobile Internet. The authors tracked 11 early U.S. mobile Internet users over a five day period and conducted a qualitative study. The authors first give a classification for motivations and behaviors. The authors found the most frequent motivation for accessing information by smartphone is awareness, a motivation usually satisfied with status checking behavior. The authors define awareness as the desire to stay current, to keep oneself informed in general. e.g. "scanning email and checking news sites". While status checking involves checking dynamic information like weather, news or sports scores during a game.

2.3 User Intent Mining Tasks

In this section, we will examine existing user mining tasks and the mainstream solutions. Generally, there're two categories of user intent mining tasks, which are user intent classification and user intent entity slot detection. For user intent classification, most of the studies first defined or applied a category criteria and conducted a classification model using features generated from textual, user sides[26, 7]. And for user intent entity detection, we found several works about recognizing named

Table 2.4: Classification of Motivation of Users information Access via Cellphone from Taylor et al. [4]

Motivation	Description	Example
Awareness	The desire to stay current, to keep oneself informed in general	scanning email and checking news sites
Time Management	The desire to be efficient, to manage projects, or get things done.	looking up an address; checking traffic maps; looking for supplies/ jobs
Curiosity	The interest in an unfamiliar topic, often based on a tip or chance encounter.	looking up information about a country of interest
Diversion	The desire to kill time or alleviate boredom	browsing favorite sites; checking social networking sites
Social Connection	The desire to engage with other people	Arranging to get together; sending email; posting to social networking sites; seeking information as a group
Social Avoidance	The desire to separate oneself from others	using cell phone activity as a "cover" to prevent a conversation.

entity type in web search.

2.3.1 User Intent Classification

Web Search Intent Classification Discovering and categorizing the user intent of Web searcher is a research area with a long history. Some initial work is from Broder [1], Rose and Levinson[5] and Jansen[26].

Broder first proposed the taxonomy of user's intent as *navigational*, *informational* and *transactional*. Navigational queries intent to reach a particular website. For example, "youtube", "amazon" are all types of navigational queries. Informational queries intent to obtain information about a topic or answer to a specific question. For example, "76ers" or "thanksgiving 2017 holidays" are examples of informational queries. Finally, transactional queries reflect user's intent to perform a particular action. For example, queries that involve playing games, downloading music, interacting with some online service, etc. are all examples of transactional queries.

Works from Rose, Levinson[5] and Jansen[26] extended Broad's web search intent framework. Rose et al. give more subclasses to the informational intent and conduct a data analysis through a sample of query log to study the percentage of different kinds of web queries. Using queries sampled from the AltaVista query logs, Rose et al. found that almost 40% of queries were non-

informational and a large proportion of the informational queries were requested to locate a product or service. They found a significant volume of resource queries (21.7- 27%) while navigational queries represented the smallest goal class with between 11.7-15.3% of queries.

Jansen et al.[26] examined the intent of Web search queries using seven transaction logs from three different Web search engines containing more than five million queries.Their findings indicated that more than 80% of Web queries were informational in nature, with about 10% navigational and just under 10% transactional. To date, this is an area that has not yet been examined within the mobile search space. However, it is likely that given the high prevalence of adult and multimedia content on the mobile Internet that the volume of transactional queries will be quite high. Although such log analysis studies provide valuable insights into what people search for, how they search for information and what the goal/intent is behind user queries, these types of analysis cannot tell us about the actual information needs of mobile users.

Hu et al. [7] proposed an intent classification method using Wikipedia. The authors claimed three major challenges for query intent classification problem: (1) Intent representation; (2) Domain coverage and (3) Semantic interpretation. They identify search intents of queries by considering Wikipedia categories and articles as possible search intents. And they consider three categories of intents: travel, personal name and job as examples to evaluate their classification performance. ODP(Open Directory Project) topical hierarchy is another source of a knowledge base for user interests representation in several studies[7][27].

Pinterest User Intent Classification[28] Cheng et al. [28] conducted a study of Pinterest users' intent behind their activity on the website. They categories user intent depending on whether they were goal-specific or goal-nonspecific, or if they were planning to take action in the short-term, long-term, or take no action at all. And they use the discovered user signals to predict user's intent.

Online Commercial Intent(OCI) Classification Online commercial intention (OCI) identification focuses on capturing commercial intention. An earlier study is based on the user query and web browsing history to identify whether a query contains commercial intent[29].

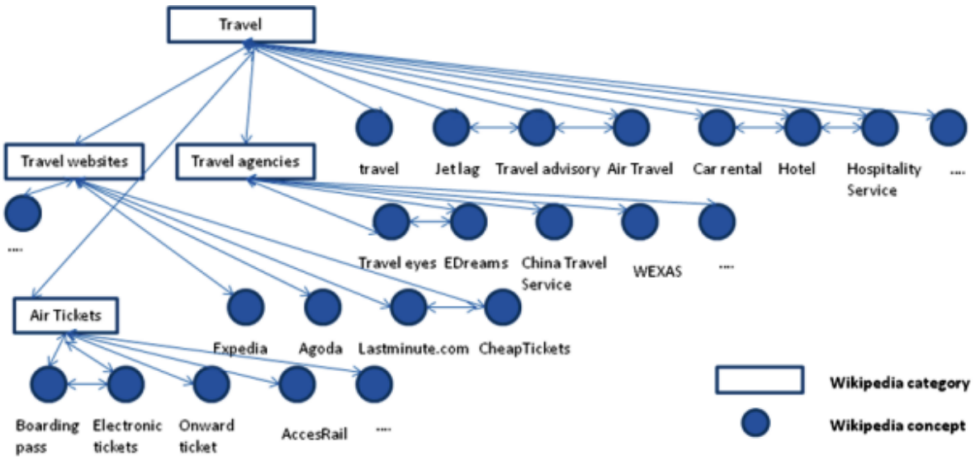


Figure 2.3: An example of travel intent concepts link graph in work from Hu et al[7]

Hollerit et al.[13] studied tweets with commercial intent. They first give a definition about a tweet containing "commercial intent" as "A tweet contains at least one verb and describes the user's intention to commit a commercial activity in a recognizable way". And they also notice another nature of intent, which is, whether a tweet's intent is explicit or implicit. Explicit vs. Implicit: The tweet "Facing Repossession, Let us buy your house for cash now" explicitly expresses the intent to buy a house. In contrast, the tweet "Debating on buying the pair of 80s cop shades..." contains to a certain extent commercial intent, but it is not explicitly stated rather as a possibility in the future. And they notice that implicit commercial intent also has commercial value. They conduct a Chi-square statistic on their labeled commercial intent dataset, and find a list of useful features for recognizing twitter with commercial intent. And they also conduct supervised classification on the dataset. Best recall scores 77.4% were achieved using a Bayes Complement Nave Bayes classifier[30]. Best precision score of 57.1% was achieved by using a linear logistic regression classifier.

Tweet Daily Life Intent Classification[3] Wang et al.[3] proposed a semi-supervised user intent classification task on Twitter data. They first give a clear definition of "intent tweet", "intent-indicator" and "intent-keyword". They define a tweet as an **intent tweet** if (1) it contains at least one verb and (2) explicitly describes the user's intent to perform an activity (3) in a recognizable way. And they also define "intent-indicator" and "intent-keyword". **Intent-Indicator**: It comprises

Rank	Attribute	Example Tweets
1	buy cheap	'buy cheap alberto vo5 shampoo strawberries'
2	to buy	'np pink fridayi think im going to buy it tomorrow'
3	for sale	for sale apple iphone 4g 32g/apple iphone 3gs 32gb buy 2 get 1 free
4	check out	#quilt lovers, check out @heyporkchop's flea market fancy scraps for auction
5	VB DT JJ	'dear allstarweekend please come back to michigan so we can buy those new shirts d','RB RB VB VB RB TO VB IN PRP MD VB DT JJ NNS LS'
6	VB JJ NN CD	'buy cheap braun 5270 silkpil x'
7	NN NN CD CD	'classifieds i am selling my gmc envoy xl 2003 for gooddemand sr 35 000 slightly negotiablei am the secon httpbitlyd3g1e4'
8	have to buy	cooking carbonnade and for drink just wine ... i have to buy food tomorrow :S
9	low price	buy cheap blue banana dresses low price everyday @amazon.co.uk http://amzn.to/9hzjhq
10	NN NN JJ CD	'buy cheap 25 usb 20 to sata hard drive hdd aluminum external 25 usb 20 to sata hard drive hdd aluminum e httpbitlybzsitp'
11	NN NN CD CD NN	'for sale apple iphone 4g 32gapple iphone 3gs 32gb buy 2 get 1 free httpbitlyhcazwp'
12	i want to	'delhi buy sell i want to sell my nokia n97 i want to sell my nokia n97 which is new brand phone with all f httpbitlyb7kglt'
13	NN IN DT JJS	'buy your new or used bmw in ebay for the best possible price more info httpbitlybf0zqr'
14	VB DT JJ NN	'about to go to first Friday with codynotontor to find an xmas present for dianevicars anyone want to join free booze and cheese'
15	VB JJ NN CD NN	'xbox 360 system link cable buy cheap xbox 360 system link cable buy low price from here now consider yourself con httpbitly9wiiej'

Figure 2.4: Top Discriminative Attributes for Commercial Intent Selected by Chi-square in Hollerit's work.

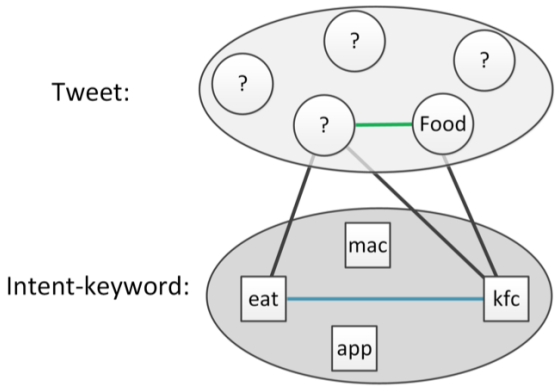


Figure 2.5: Intent-graph illustrating relations among tweets and intent-keyword in Wang's work. Two types of nodes, which are intent-tweet node and intent-keyword node respectively.

a group of terms that are used by users to express their intents. It is a verb or infinitive phrase that immediately follows a subject word, e.g., "I". For example, in tweet "I want to buy an xbox", "want to" is an intent-indicator, indicating the tweet is likely to be an intent tweet. **Intent-Keyword**: It is a noun, verb, multi-word verb or compound noun (consisting of several nouns) contained in a verb or noun phrase which immediately follows an intent- indictor, e.g., in the previous example, "buy" and "xbox" which are contained in the phrase "buy an xbox" are intent-keywords.

The authors conduct a intent classification by graph-based semi-supervised approach. They firstly prepare dataset by bootstrapping method. Specifically, they first use a set of seed intent-indicators, such as "want to" to find intent phrases from tweets. And the extracted intent phrase with high confidence can further help to find more intent-indicators. And then, they construct a intent graph, with tweet and intent keywords. Based on the intent-graph, the problem of inferring intent categories from a small number of labeled tweets is formulated as an optimization problem.

Intention Post Classification on Online Forum Chen et al[31] proposed a transfer learning based framework to learn a binary classifier for identifying intention posts in the online discussion forum. In their work, they aim to recognize posts with explicit intention, such as "I am looking for a brand new car to replace my old Ford Focus" or "I plan to buy a new TV". In their work, they proposed a new transfer learning method, called Co-Class. Suppose there're two datasets, called

source data and target data, which source data is a labeled dataset we already have and target data is the unlabeled dataset. And the transfer learning process is as follows:

1. Build a classifier using the labeled data from the source data.
2. Apply this classifier to target data.
3. Perform a feature selection based on the predicted label of target data.
4. Use the selected feature set to build two new classifiers from source data and target data.
5. Use the two new classifiers together on target data
6. Iterate this process until the label for target data is stable.

2.3.2 User Intent Attribute and Entity Detection

Query Named Entity Recognition Guo et al.[32] addresses the problem of Named Entity Recognition in Query, which involves detection of the named entity in a given query and classification of the named entity into predefined classes. In their study, they focus on four named entity classes, which are, "game", "movie", "music", and "book". Given query "harry potter walkthrough", "harry potter" will be detected as a named entity. "Game" would be most likely be assigned as the class of this named entity, while "Movie" and "Book" are less likely classes, and "Music" is unlikely class. And this work is very similar to user intent attribute detection. Because looking for some information about "Game" is the latent **intention** behind this query, and "Harry Potter" actually is the **attribute**, or content of this intention. In their work, they regard the intent of query as the latent variable and conduct a weakly-supervised LDA to model the problem.

Pantel et al[9] jointly model the user intent and entity type in a web search user query. They theorize that search queries are governed by a latent user intent, which will, in turn, influences the choice of query words, and the clicked link. The intent based model is proposed in Figure 2.7.

Guo et al. [33] propose a query similarity measure using user intent, and they argue that the similarity between queries should be defined upon search intents. They conducted a topic-modeling approach to learn potential search intents.

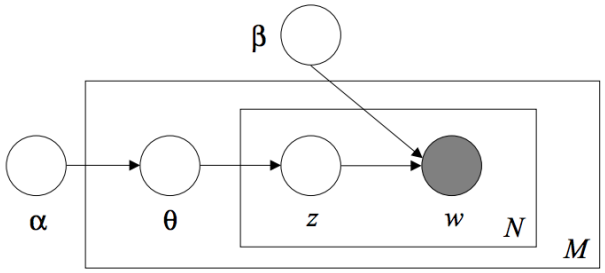


Figure 2.6: Graphical model representation of LDA[8]

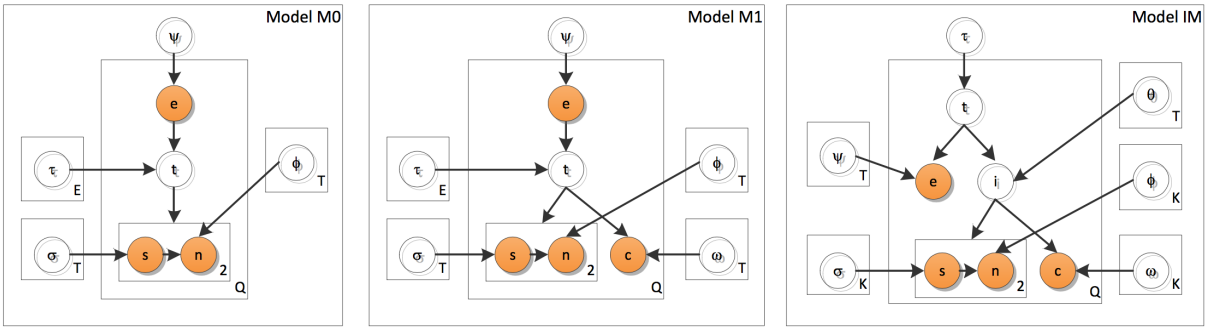


Figure 2.7: Graphical representation of the intent-based model proposed in [9]

Suppose there're N queries sharing K potential search intents, and each query is represented by a M words. By viewing queries as virtual "documents", words from top search result snippets as "words", and potential search intents as "topics", we can apply the Probabilistic Latent Semantic Indexing (PLSI)[34] to model the generation of each query and its words from top search result snippets by the following scheme:

1. Select a query q_i with probability $P(q_i)$

2. Pick a potential search intent s_k with probability $P(s_k|q_i)$

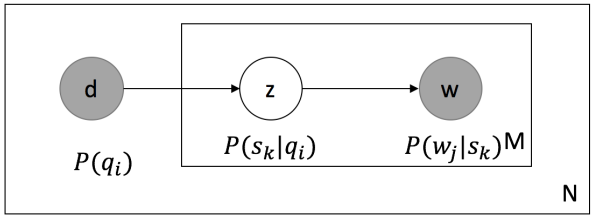


Figure 2.8: Plate Notion for the PLSA model

3. Generate a word w_i with probability $P(w_j|s_k)$

2.3.3 Dynamic User Intent Modeling

As we analyzed in the previous section, we are more concerning the temporally dynamic nature of user intent. To dealing with the user interests changing over time, for years a lot of studies have been done on user's interest drift, long-term, and short-term user interests mining. We will review the studies and analysis our considerations.

The most often used approach to deal with user interests prediction considering the temporal dynamic character is the so-called time window[35]. It learns the interests only from the newest observation[36]. An improvement of this approach is the use of heuristics to adjust the size of the window according to the current predictive accuracy of the system [37]. Shen et al. [6] linked the entities in tweets and learn user interested topics from user's previous tweets with a fix window size.

Abel et al.[38] have modeled user interests in a given timestamp as a set of weighted concepts which are entities or hashtags extracted from the user's tweets in that timestamp. For calculating the weight of each concept, the tweets with shorter temporal distance to the given timestamp are assigned greater weight since they are considered to be more important. The authors have also shown that considering temporal dynamics of the user interests can improve the performance of a personalized news recommender system.

Ahmed et al. [39] have used an exponential decay function to model the dynamic user behavior in search logs. But they have assumed that the parameter of the decay function remains constant for all topics. Our work uses a different approach considering there're parameters to control the time decaying of each topic of interests.

The temporal submission pattern in user behavior history carries valuable information. The existing temporal based methods make use of the historical information to predict the future. However, those existing methods only use them for either simply splitting sequence of activities into temporal demarcated sessions, or transforming them as features [40]. We believe by directly modeling temporal information as part of the user behavior modeling in a richer way, we can substantially improve the prediction result.

Chapter 3: User Intent Mining on Mobile Devices

3.1 Introduction

Along with the popularized of smartphone, people's daily life is more tightly bounded with apps and online services than ever. For many people, the smartphone has become an indispensable life partner. Through smartphones, they can order food, call Uber, entertainment, and even work such as attend video meetings, and so on. Personal assistants of smartphone, such as Siri from Apple, Alexa from Amazon, make the expression of user intent to cellphone or mobile devices more directly. And people are more and more used to express their need explicitly or implicitly through mobile devices.

Users may use a short sentence to express their needs for mobile functionalities. For example, uses may use "go home" for a navigation function from Google Map. "shuffle the track" indicates a needs for random play the current music track. In addition to the intent category, the expression may also include an intent attributes. For example,"call Alice" indicate an intent for making a phone call, and the intent attribute is "Contact Name", which is a person whose name is "Alice". Then in next step, if there's a person named Alice in user's contact, her number could be dialed. Another example is "find a parking garage in union square". This expression needs to open an App with parking information. And it also has intent attributes, such as "Parking type" is "Garage Parking" and "Location" is "Union Square".

Referring to Belkin's states of a searching episode [11], intent is akin to goal, and expression akin to method of interaction. In the scenario of mobile usage, user intent is expressed as a short text from either a text box or voice input assistant. And the answer to this expression should be an action or series of actions that could fulfill user's need. User intent behind this expression is a latent concept or entity that could bridge the expression from the user and the action from smartphone. From what we have discussed above, in this work, we define the user intent as an application-oriented knowledge graph. In this work, we will first broadly give a user intent knowledge graph within a mobile usage

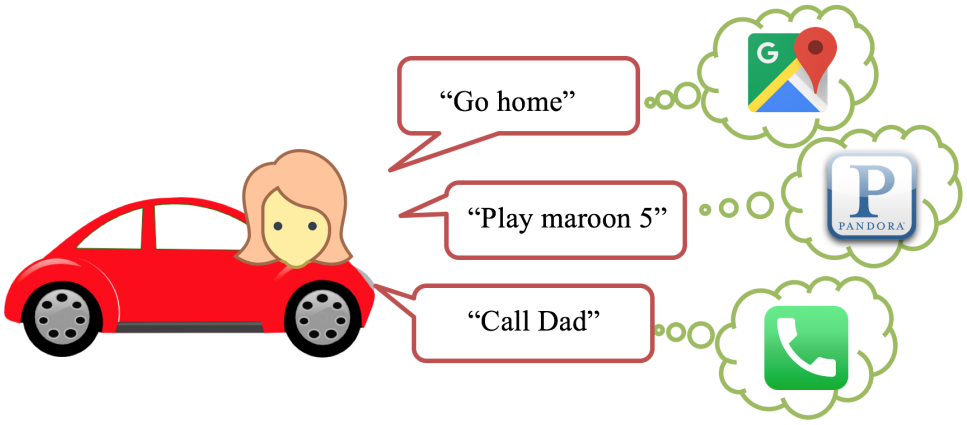


Figure 3.1: Example about User Intents of using cellphone while driving

domain. And then propose a solution for user intent understanding and mining according to this intent knowledge graph.

Since the user intent of using mobile devices are too broad compared to authors' time and energy limitation, we first narrow this problem down into a mobile usage domain. We choose to categorize intent when user using smartphone during driving. We choose this scenario for several reasons. First of all, operate cellphone using voice command is more safety during driving. So it's more necessary to develop a system to understand user's intent behind. Also, driving is a typical mobile usage scenario when user is on the move with many contextual information from cellphone sensor, such as location, time.

In this work, our contribution is as follows. First of all, we build a user intent knowledge graph in a driving scenario. The knowledge graph contains both user intent categories and related attributes. Besides, a dataset with user queries and related intent is created for boosting the work. Thirdly, we conduct a novel short sentence classifier framework using word embedding for user intent classification. Word embedding could deal with out-of-vocabulary words for the real application. And by fine-tuning the embedding with our dataset, the classifier's accuracy could be further improved. To recognize the intent attributes, such as location, person's name, we develop effective features from context information and external resources and feed them into a sequential tagging model.

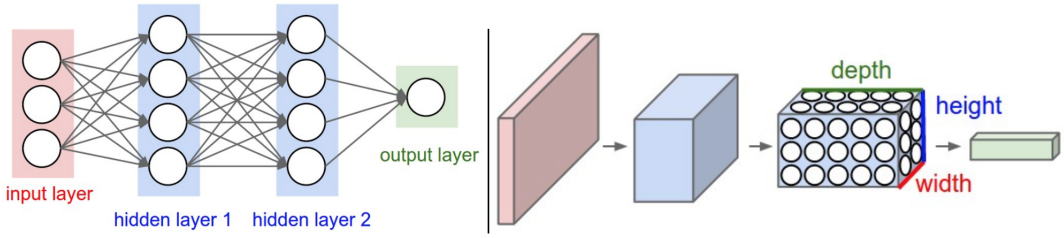
3.2 Proposed Methods

In this work, our user intent mining includes two part, which are user intent classification and user intent recognition. In the following part, we will first illustrate a Convolutional Neural Network based classifier framework for user intent classification in short text. And then we will apply a rich context feature based Named Recognition using CRF for intent object detection.

3.2.1 User Intent classification

General Framework

Convolutional Neural Networks(CNN) are very popular framework in visual recognition. And recently, it receives many attentions in text mining domain[41]. From the prospective of image processing, convolutional neural network receive a raw image as input, and output a class score. But unlike regular neural network, which neurons between layers are fully-connected, the neurons in CNN in a layer will only be connected to a small region of the layer before it.



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

Figure 3.2: Architecture of Convolutional Neural Network for Image Processing

Our mode is similar to the CNN based sentence classifier architecture of Kim[41]. Sentences are represented as a 2-D matrix, each raw of which is a word vector. Unlike applications of Convolutional Networks on image, here we use a filter, which dimension is `window_size * word_vector_dimension`. For image, the filter is a K*K window, which can involve context impact of the current pixel. For natural language, context information is also important for understanding the meaning, semantic information of the current word. Formally, for a given sentence S with n words $S = (x_1, x_2, ...x_n)$,

each word with m dimensions, $x_i \in \mathbb{R}^m$. Then the sentence could be regarded as a $m * n$ matrix. To perform a convolutional operation, let define a learnable filter $\mathbf{w} \in \mathbb{R}^{k*m}$. The filter could slide(or convolve) from the first row of sentence matrix to bottom. Each time, the filter will compute a dot product between a k-word sequence and the filter.

$$c_i = f(w * x_{i:i+k-1} + b) \tag{3.1}$$

Here $b \in \mathbb{R}$ is a bias term, and f is a non-linear activation function, such as the hyperbolic tangent. As we stated previously, this filter will be applied to each possible word window of the sentence, and then we will get a feature map $c \in \mathbb{R}^{n-k+1}$

$$\mathbf{c} = [c1, c2, ..., c_{n-k+1}] \tag{3.2}$$

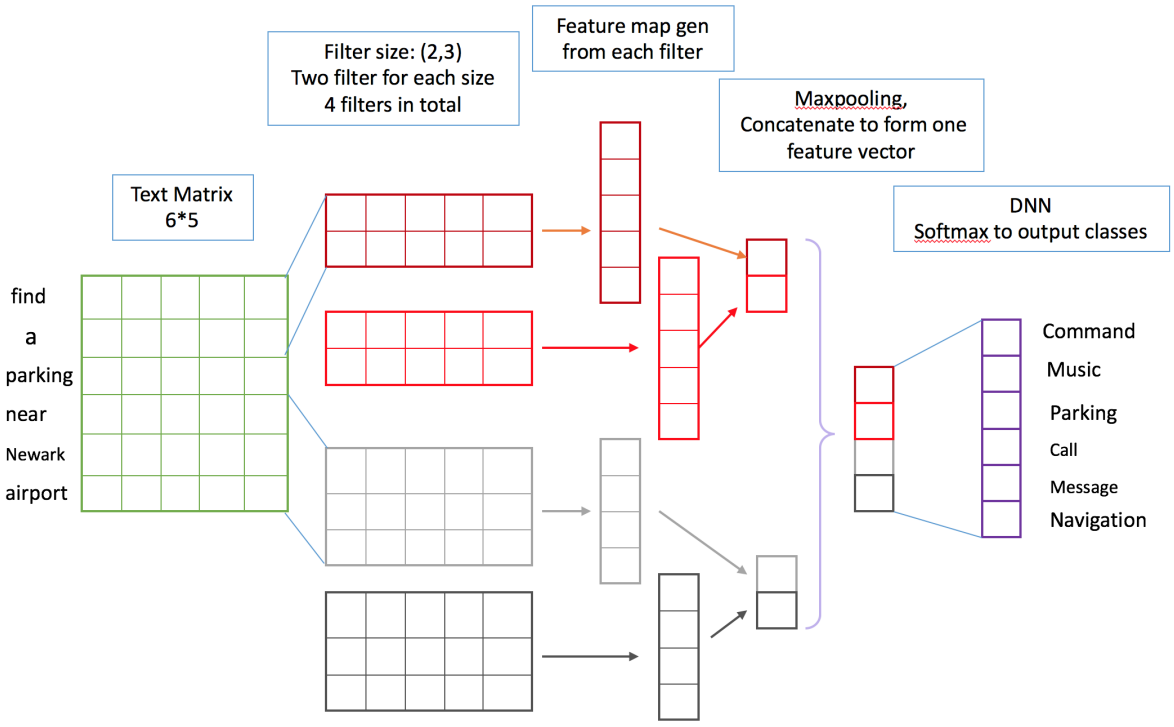


Figure 3.3: Illustration of model architecture

We then apply a maxpooling mechanism[42] which will output the maximum value $\hat{c} = \max(\mathbf{c})$

of this feature map as the feature returned by the given filter **w**. The intuition behind this practice is to capture the most important feature from each feature map. If we have m_1 filter sizes, and m_2 filters for each filter size, then $m_1 * m_2$ features will be generated from convolutional layers. And then these features will be passed to a fully connection softmax layer. Softmax could be regarded as a multiple classes generation of Logistic Regression. And it will give a probability distribution over target classes.

Dropout is an important technique to improve the neural network performance. In this work, we also employ dropout on the concatenate layer(the layer concatenate the maxpooling result from each feature maps.).Dropout prevents co-adaptation of hidden units by randomly dropping out[41]. A node may be randomly set to zero with a given probability. And here, we set it to 0.5.

3.2.2 Intent Entity Detection

Linear statistical models, such as Hidden Markov Models(HMM), Maximum entropy Markov Models(MeMMs), and Conditional Random Fields(CRF)[43, 44, 45] have been widely used in sequence labeling tasks.

Conditional Random Fields (CRFs)[43] are undirected graphical models, a special case of which correspond to conditionally-trained finite state machines. While based on the same exponential form as maximum entropy models, they have efficient procedures for complete, non-greedy finite-state inference and training. CRFs have shown empirical successes recently in POS tagging[43], noun phrase segmentation[46] and Chinese word segmentation[47]. The capability of wide array of features of these models give great flexibility incorporate rich features for solving the problem.

Conditional Random Fields

Conditional Random Fields (CRFs)[43] are undirected graphical models used to calculate the conditional probability of the output nodes Y given assigned values on input observation nodes X. It offer advantages over both generative models like HMMs and classifiers applied at each sequence position.

Formally, we define $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$

corresponding to each of the random variables representing an element Y_v of Y. If each random variable Y_v obeys the Markov property with respect to G, then (Y, X) is a conditional random field. When modeling sequences tagging task, the most common used graph structure is that in which the nodes corresponding to elements of Y form a simple first-order chain, as illustrated in Figure 3.4.

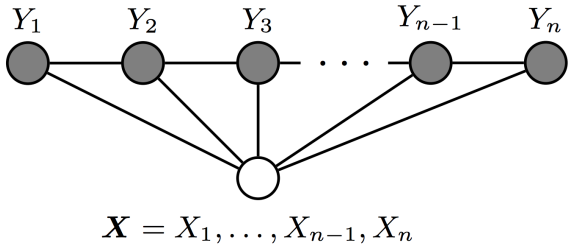


Figure 3.4: Linear Chain Conditional Random Fields

Let $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ be an sequence of input words of length n. Let Y be a set of finite states, each element in the set is a label $l \in L$. Let $y = \{y_1, y_2, ..., y_n\}$ be the sequence of labels corresponding to the input words. Conditional random field is a discriminative model defined a conditional probability of possible labels sequence y given an input sequence x.

$$p(\boldsymbol{y}|x) = \frac{1}{Z(\boldsymbol{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t) \tag{3.3}$$

where each function Ψ_t specifically defined as :

$$\Psi_t(y_t, y_{t-1}, x_t) = \exp(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t))$$

$Z(\boldsymbol{x})$ is a normalization factor over all the state sequences, defined as:

$$Z(\boldsymbol{x}) = \sum_{y} \prod_{t=1}^{T} \exp(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)) \tag{3.4}$$

$Z(\boldsymbol{x})$ is the sum of the scores of all possible state sequences, and that the number of state sequences is exponential in the input sequence length n.

$f_k(y, y', x_t)$ is one of K arbitrary feature functions over its arguments. In this work, we use linear-chain CRF model with binary feature functions.

The most probable label sequence for input sequence x is

$$\hat{y} = argmax_x p(y|x) = argmax_x \theta F(y, x) \tag{3.5}$$

For example, a feature function may be defined to have value 0 in most cases, and have value 1 if and only if y_{t-1} is state "OTHER", and y_t is "LOCATION", and the observation at position F in is a word appearing in a list of country names. Features we used here including vocabulary features, gazetteer features, orthographic features, and statistic features.

θ_k is a weight learned for each feature function. Intuitively, a higher value of weight indicates that the feature is more correlated with the target output label. And a close to zero weight indicates a not much related feature. Higher θ_k weights make their corresponding finite state machines transitions more likely.

Training CRFs

The weights of CRF, $\boldsymbol{\theta} = \theta, ...$ are learned by maximize the conditional log-likelihood of the labeled sequences in a training set $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$.

$$l(\theta) = \sum_{i=1}^{N} \log(y^{(i)}|x^{(i)}; \theta) \tag{3.6}$$

When the training labels make the state sequence unambiguous (as they often do in practice), the likelihood function in exponential models such as CRFs is convex, so there are no local maxima, and thus finding the global optimum is guaranteed. It has recently been shown that quasi-Newton methods, such as L-BFGS, are significantly more efficient than traditional iterative scaling and even conjugate gradient [48, 46]. This method approximates the second-derivative of the likelihood by keeping a running, finite-sized window of previous first-derivatives.

Newton methods for nonlinear optimization use second order(curvature) information to find search directions. It is not practical to obtain exact curvature information for CRF training. Limited-memory BFGS (L-BFGS) is a second-order method that estimates the curvature numerically from previous gradients and updates, avoiding the need for an exact Hessian inverse computation. L-BFGS

can also handle large-scale problems but does not require a specialized Hessian approximations. Studies indicated that L-BFGS performs well in maximum-entropy classifier training[48].

Most of studies used a heuristic method to estimate how much information from previous steps we should keep to obtain sufficiently accurate curvature. In our studies, we are not focusing on model optimization, thus we treat L-BFGS as a black-box optimization procedure, requiring only that one provide the first-derivative of the function to be optimized. Assuming that the training labels on instance make its state path unambiguous, let $\boldsymbol{s}^{(j)}$ denote that path, and then the first-derivative of the log-likelihood is:

$$\frac{\partial l}{\partial \theta} = (\sum_{j=1}^{N} C_k(x^{(j)}, y^{(j)})) - (\sum_{j=1}^{N} \sum_{s} P_\Lambda(\boldsymbol{y}|x^{(j)}) C_k(\boldsymbol{y}, \boldsymbol{o}^{(j)})) - \frac{\lambda_k}{\sigma^2} \tag{3.7}$$

where $C_k(\boldsymbol{x}, \boldsymbol{y})$ is the "count" for feature k given $\boldsymbol{y}$ and $\boldsymbol{x}$. It is equal to $\sum_{t=1}^{n} f_k(y_{t-1}, y_t, \boldsymbol{x}, t)$, which is the sum of $f_k(y_{t-1}, y_t, \boldsymbol{x}, t)$ values for all time t in the sequence.

Features

We try to train a Named Entity Recognition model from our labeled dataset. We have several type of entities to extract, such as location, person's name, calendar, and etc. We extract features for the user input short text for user intent attribute extraction. The features are context feature, semantic features, statistical features, and gazetteer features.

Context Features For context features, we generate unigram and bi-gram within a certain window size of a given word. We also include POS parser result for these bag-of-word features.

Semantic Features Part of speech tagging(POS) is applicable to a wide range of NLP tasks including named entity segmentation and information extraction.

Statistical Features Word frequency information is used as an efficient feature. Word are grouped by its frequency ranking, such as top100, top500, top1000, top5000. This feature proved to be useful for reduce sparsity and differentiate important words.

Table 3.1: User Intents for using mobile devices while driving

Intent Type	**Description**	Examples
Call	The intent to call someone.	1. Call Bob 2. Dial 123456
Command	The intent to operate or interact with cellphone's general functions, such as agree or disagree, hangup the call, etc.	1. Dismiss 2. Stop the navigation
Message	The intent for send, reply messages.	1. Text my mom 2. reply Diana
Music	The intent to play music, choose playlist, get music information, etc.	1. Play some Taylor Swift 2. Shuffle the track
Navigation	The intent to find routing information to some places	1. Locate the bests sushi restaurant 2. Show me the nearest Starbucks
Parking	The intent to find, list or reserve a parking space	1. Reserve a parking lot at 6 pm to 10 pm near park 2. Look for a parking garage closest to the mall

Gazetteer Features We also involve various kind of gazetteer information. This feature is very important to reduce sparsity and add additional information from external resources.

3.3 Experiment and Results

In this part, we will first describe the dataset we use and how we label the dataset. And then, we will illustrate our experimental performance and discuss the results.

3.3.1 Data Preparation

In order to perform experiment for the task, we need dataset of user query, and the related user intent. But as far as we know, there's no available dataset. Due to the time and label limitations, we narrow down the user intent on mobile app usage into several categories, and collect possible queries from crowdsourcing platform. Basically, we have generally six categories of user intent, which are Call, Command, Music, Message, Navigation and Parking. And to further bridge the user intent to the downstream services, we give subcategory for Call and Music. So we have 8 categories in general. And here are some detail intent type description.

And we also test the task of intent entity detection on parking domain. For parking domain, there're several intention entities, and we give the definitions as Table 3.2. Some of the entity types with be recognized by rules, such as "price reference" and "distance reference" and etc. And we will

Table 3.2: Parking Entity Types and Values in our Application

Intent Entity Type	**Description**	Intent Entity Value
Price Reference	A reference about the service price	Cheapest Free
Distance Reference	A distance needs about the parking place	Closest Nearest
Parking Type	Type of parking	Garage Valet Lot
Location	Location information of the parking	Restaurant Houston airport
Calendar	Date and time when the parking is use	From 3 am to 6 am After 5 pm

Table 3.3: Dataset Statistics

Intent Type	Dataset Size
Call_a	619
Call_b	311
Command	761
Message	1355
Music_a	1759
Music_b	442
Navigation	2344
Parking	1803
sum.	9394

use CRF model to learn a NER model for "location" type.

The dataset is collected from Amazon crowdsourcing platform, Mechanical Turk[1]. We totally collected 9394 sentences by the time we conducted our experiments. And the data is not very balanced among categories. We illustrate our dataset statistics as Figure 3.3

To generate training and testing dataset, we randomly shuffle it and then split it into 90% for training and 10% for testing. In addition, some of user input may not contain target intent. For example, users may input some queries such as "Nice weather" or "hello". It also possible that we cannot offer further services for users query, for example: "let's see a movie!"(the intent is open a video app and play a video, which is not appropriate during driving). For such queries, we need a "ignore" category to pass these queries out. The "English-900" sentence set contains 900 sentences used in conversational English. Most sentences are colloquial and conversational. We collect the conversation data of "English-900" and split the conversation into sentences. We annotate the

[1]www.mturk.com

sentences not related to the target classes we given as a new category for training.

3.3.2 Experiments on User Intent Classification

We use following multi-class classification methods as compare benchmark to evaluate the performance of our proposed method.

Naive Bayes Naive Bayes or Multinomial Naive Bayes is a probabilistic learning method. The probability of a document d being in class c is computed as $P(c|d) \propto P(c)\prod_{k=1}^{n_d} P(t_k|c)$, where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. $P(c)$ is the prior probability of a document occurring in class c. If a document's terms do not provide clear evidence for one class versus another, the prior is another evidence for choosing document label.

Multinomial Logistic Regression Logistic regression is a popular and strong benchmark for binary classification problem. A sigmoid function is used as activation function to map the result of linear regression to a probability. Extend this setting to a multi-class classification, softmax function is used as action to get the probability distribution over multiple classes[49]. Suppose there're k classes, the probability of $x^{(i)}$ being assigned to class j is $P(y = j|z^{(i)}) = \frac{\boldsymbol{e}^{z^{(i)}}}{\sum_{j=0}^{k} \boldsymbol{e}^{z_k^{(i)}}}$. $z^{(i)}$ is the dot product of the feature vector $\boldsymbol{x}$ and weight vector $\boldsymbol{w}$ plus a bias term, defined as: $z^{(i)} = w_0x_0 + w_1x_i + ... + x_mx_m = \sum_{l=0}^{m} x_lw_l = \boldsymbol{w}^T\boldsymbol{x}$.

KNN Classifier The KNN algorithm is a robust and versatile classifier that is often used as a benchmark for more complex classifiers such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Despite its simplicity, KNN can outperform more powerful classifiers and is used in a variety of applications such as economic forecasting, data compression and genetics. Given a positive integer K, an unseen observation $\boldsymbol{x}$ and a similarity metric $\boldsymbol{d}$, KNN classifier performs the following two steps[50, 51]: 1. It runs through the whole dataset computing $\boldsymbol{d}$ between $\boldsymbol{x}$ and each training observation. We will call the K points in the training data that are closest to $\boldsymbol{x}$ the set $\boldsymbol{A}$. K is usually odd to prevent tie situations. 2. It then estimates the conditional probability for each class, that is, the fraction of points in $\boldsymbol{A}$ with that given class label.

Table 3.4: Result Compare with all benchmark methods

	Precision	Recall	F Score
KNN Classifier	0.9076	0.9041	0.9046
Decision Tree	0.9197	0.9172	0.9171
Random Forest	0.9385	0.9385	0.938
Nave Bayes	0.9321	0.9302	0.9271
Logistic Regression	0.9137	0.9041	0.9011
SVM	0.941	0.9361	0.9351
Proposed Method	0.9621	0.9615	0.9603

SVM Support vector machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.The operation of the SVM algorithm is based on finding the hyperplane that gives the largest margin to the training examples. I this chapter, we use a 'one-vs-the rest' strategy for multiple classes prediction[52, 53].

Decision Tree Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)[54]. It is a commonly used classification and regression data mining method.

Random Forest Random Forest classifiers create a whole bunch of decision trees (hence "forest"). Each decision tree is trained on random subsets of training samples (drawn with replacement) and features (drawn without replacement). And have the decision trees work together to make a more accurate classification[55].

We compare our CNN based method with the baselines and the performance are illustrated as Table 3.4.

The result outperforms the existing benchmark methods. For all the compared methods, random forest and SVM achieves the best performances, which are 93.8% and 93.51% in F-score. And our proposed method's F-score reaches 96.03%, with an improvement of 2.38% and 2.69% respectively.

Our proposed method has the ability to deal with out-of-vocabulary words. Currently, we are using a crowdsourcing dataset to build a training dataset to boost our model. And we use word

Table 3.5: Evaluation on intention sentences with OOV word

Methods	Accuracy
KNN Classifier	0.5556
Decision Tree	0.8889
Random Forest	0.8519
Nave Bayes	0.5926
Logistic Regression	0.7037
SVM	0.7407
Proposed Method	0.9629

embeddings as the input of our model. Although our training data is not enough to handle all the word in test dataset, a pretrained word embedding like GloVe, or Word2vec can be a helpful supplement. Here we sample a dataset with a few sentences, each of which contains one or more than one out-of vocabulary word. Because in our proposed model, our embedding is tuned during the neural network training process, thus it's hard to deal with words that are not included in the training dataset. To leverage the pretrained word embedding, we firstly scan the input sentence to see if there're OOV word. If there're OOV words, they will be replaced by a similar word if exist in the training data. Otherwise, it will be label as "UNK".

Table 3.5 illustrate the performance on OOV sentences. One sentence in our OOV testset is "let's find some Mongolian bbq" and the label for this sentence is "Navigation". But several classifiers will assign this sentence to "Music". Because without knowing that "Mongolian bbq" is a place of restaurant, the classifier will guess this sentence with the known words, and there's a similar sentence is music domain "let's hear some rock n roll".

3.3.3 Experiments on Intent Attributes Detection

In this part of work, we would like to evaluate our CRF-based intent attribute detection work. We trained this model use an open-source CRF library, Mallet[2][56].

Our input data is user input short sentences, as it is not formal like news or document data. So we would like to compare our intent entity recognition model TwitterNLP[57, 58]. TwitterNLP is an open named entity recognition and event extraction project. And it has pretrained model to recognize entities such as movie, person, and location. For a single user input sentences, we will

[2]http://mallet.cs.umass.edu/

Table 3.6: Performance of our model on both the training and test sets of the two tasks compare with baseline method

Model	Train			Test		
	Precision	Recall	F1	Precision	Recall	F1
Twitter_NLP	0.8723	0.8452	0.8585	0.8000	0.7692	0.7843
Proposed_Model	0.9755	0.9625	0.9690	0.9044	0.8913	0.8978

extract features such as unigram, word in a window, if word in any sources, such as top frequency word dictionary, city name dictionary, first name dictionary, etc. The evaluation of the proposed model is illustrate as Table 3.6.

3.4 Conclusion

In this chapter, we study the user intent on mobile devices on driving domain. We define six intent types for user using smartphone during driving, which are "Call", "Command", "Message","Music","Navigation" and "Parking". And for each user intention type, we also define several related attributes, such as "Parking Type", "Navigation Destination". We build a dataset through crowdsourcing platform. With this dataset, we apply a classifier using convolutional neural network to classify user's intent, and it outperforms several baseline models. For intent attribute recognition, we use Conditional Random Fields with carefully feature engineering.

Chapter 4: Implicit Query Intent Mining using Multimodal RBM

Nowadays, search engines have become indispensable parts of modern human life, which create hundreds and thousands of search logs every second throughout the world. With the explosive growth of online information, a key issue for web search service is to better understand user's need through the short search query to match the user's preference as much as possible. However, due to the lack of the personal information in some scenario and the huge calculation when seeking for relevant user group, personalized search becomes a quite a challenging problem. In this work, we propose a novel scalable framework based on multimodal Restricted Boltzmann Machine (RBM) to do the user intent mining and prediction. This scalable framework works in an unsupervised manner, and is flexible to various situations regardless of the amount of individual information, in other words, it can handles scenarios without personal history information or limited personal history information, the more individual data the better accuracy of user intent prediction and more capable to reflect the individual's interests changing. The framework outputs a binary representation for each query log, thus to some extent, could solve data sparsity problem and reduce the computation complexity when looking for users with similar interests. The experiment results shown that, the model can learn reasonable user intent category during the learning procedure, according to the qualitative analysis of the top ranked context and websites for each class. And it can get a competitive performance when no individual data is offered. Moreover, by offering more individual data, the overall performance improves up to 10% of precision.

4.1 Introduction

Search engine plays an important role in life for people to find information and for years it has greatly facilitate people's daily life. However, it's always not an easy problem for machines to understand what people are looking for. Moreover, different people have very different interests. And even for one individual, his/her interest will change over time. Thus it's necessary for online search service

to meet the need of personalized searching and adapt to the change of user intent over time. As a result, user specific information, e.g. user profile, user query history, or previous view content information become significant when identify the user's taste.

Studies have shown that personalization algorithm can have a promoting result when there's sufficient amount of user data[59]. However, it's always challenging to acquire adequate user information because of the privacy issues. So, many studies seek solutions by developing group level personalization, which combines limited individual information with other related people to perform a collaborative filtering[60]. But to find similar users to enrich personalization is also challenging because of the data sparsity and have to compute the distance among each user. Moreover, the user information suffers from great imbalance. The imbalance amount of user personal data is resulted from various reasons, but the situation is that some user may have plenty of online activity records while others may be almost no trace at all. And this requires the model to flexible enough to fit different scenario. When there's no personal data, the model can learn to mine the user intent from public dataset. And it should scale up the personal model training when there's adequate individual data.

Compared to other resources like tweets, blogs, etc., search engine query logs can more directly reflect users' interests and needs. When use search engine, people tend to use brief and direct words to describe their needs, mostly they will use named entities. In domains of data mining, a named entity refers to a phrase that clearly identifies an item from other items that with similar attributes. Examples of named entities are location, person's first and last name, address, product names, and etc. Different users may look for different aspects of a named entities and it's difficult for the search engines to tell users' exact search intent. Query logs from search engine provide huge amount of user search information. And studies have shown that nearly 70% of query logs contain single named entities (e.g. "Gone girl trailer")[59]. These named entities cover varies categories of named entities such as movies, music, books, autos, electronic products and etc.

In this work, we propose a novel scalable framework to learn from both huge amounts of public query logs and an individual's own query activity to understand user's intent. By offering more

personal search history, the model can learn user's intent more accuracy. And without history activity records, it also can work by leveraging the model learned from public and try to make a reasonable decision. Furthermore, the model also adapts and reflects the user interests changing. The input of the model is a large query log dataset. Each example in the dataset is a query log data, containing user query and the URL the user clicked. By learning from this dataset, when given a new query and the user's search history, the framework will return several high possible URLs that the user may interest.

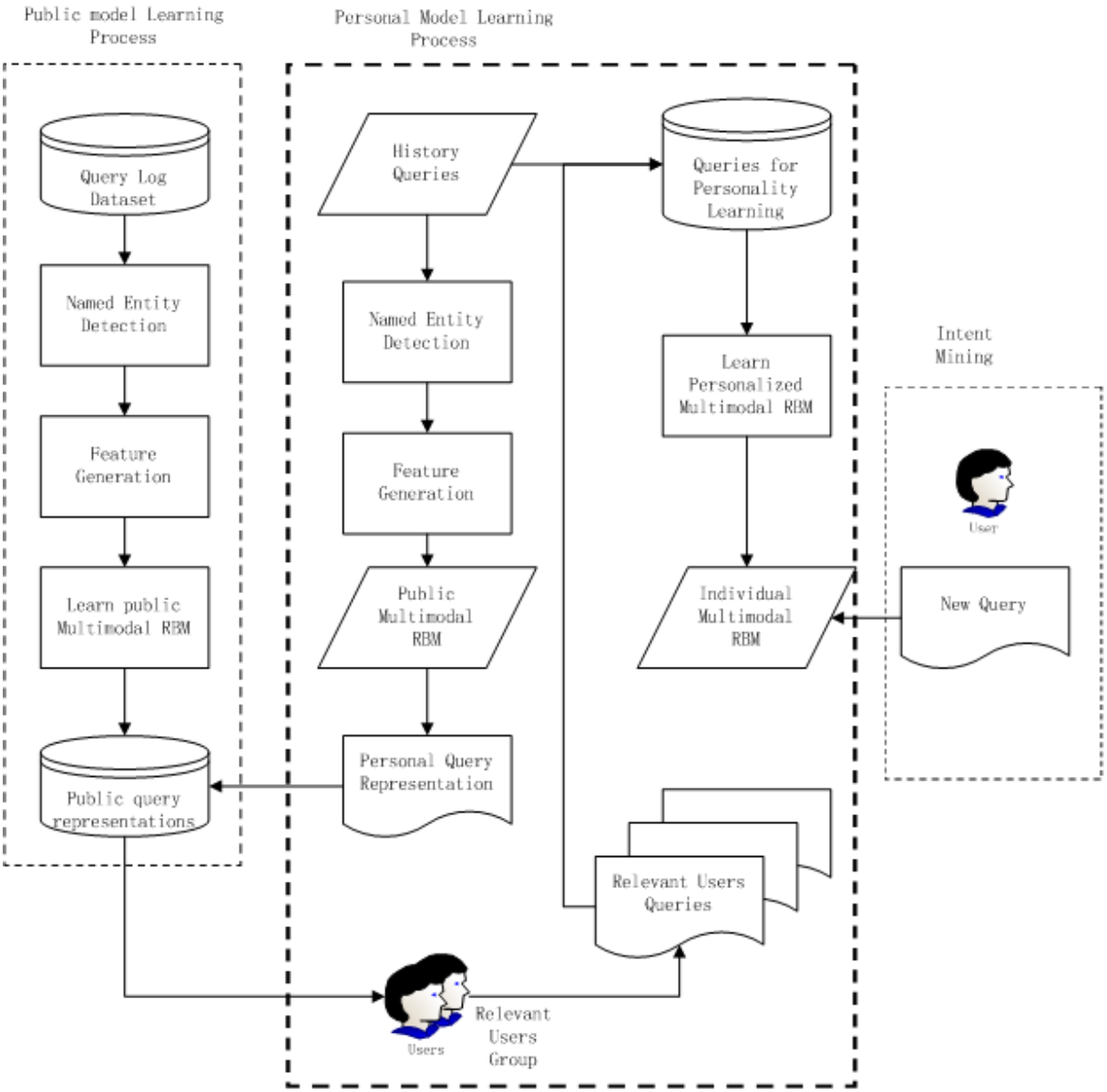


Figure 4.1: Framework of user intent understanding of query log using mRBM

In the framework, there are three major components: public model learning process, personal model learning process and user intent mining. By learning from public query log dataset, we can learn the intent relationship between users and websites, and general representations of the queries. Here're the overall steps for learning from public dataset. 1) We refine the n-grams from the whole dataset using PMI to build a candidate named entity set. 2) For each query with a candidate named entity, we represent it as a feature vector using some general text features. 3) We train a multimodal RBMs on both the query feature and URL. And to fulfill the personalization need, we use the individual log history to find users (user here refers to data examples with same section ID) that have similar interests. We calculate the representations of the personal data by model we learn from public dataset. Then it's time to compare the personal data with public to find the relevant users share similar interest with this user. Because the representation generated by our framework is binary and it also not that sparse as the raw data, calculation is more efficient. And finally, we use the data from all the relevant users to train a personalize model to predict what the user is looking for.

In term of problem, our work attempt to mine the user intent to find from query, which is an active research area in IR and text mining. Researchers have done many promising works on various applications, such as named entity mining[59], query suggestion[60], relevance feedback[61, 62]. In term of techniques, multimodal deep learning is a group of method use restricted Boltzmann machine[63, 64], auto-encoder[65] or recursive neural network[66] to generate embeddings for data with different source type, such as text and image. And our work is inspired by Xu's work, and we try to build a scalable learning model to introduce user's historical information to enhance the model's performance.

The novelty of the proposed unsupervised framework can be summarized as follows: (1) the scalability supported with multiple multimodal restricted Boltzmann machines, which can deal with situations with or without user history data. And more personal data will improve the accuracy by finding more relevant users. (2) the change-over-time trend detection model to reflect the user's interest change over time; (3) Sparsity reduction using a binary representation for each query log, a

high level feature to describe the query data.

4.2 Proposed Methods

As we have analyzed above, in this work, we design a scalable multimodal learning framework. The general workflow of the model is shown in 4.1. There're two learning processes, that is, public queries learning process and personal queries learning process.

						
Alice	1	1	1	0	0	0
Bob	1	0	1	0	0	0
Carol	1	1	1	0	0	0
David	0	0	1	1	1	0
Eric	0	0	1	1	0	0
Fred	0	0	1	1	1	0

Figure 4.2: Example of Movie classification using RBM

Public data learning process aims to learn general representations from large dataset and learn the parameters of model. This learning process only needs to run one time. While for personal queries learning process, when there's a user submit a query, and suppose there's history data for the user's online activity, the framework will try to learn a personal model.

First, making use of the model learned from public dataset, the personal data can be represented to the general representations. And calculate the similarities between representations of personal data and public data, and find the most similar queries. According to our model, the similar query representations mean similar queries and similar click information. Thus, we assume that users share similar query representations have similar tastes. Thus we collect the query data from these users and combine with this user's personal information to train a personal multimodal RBM model. And this model may reflect the user's preference better because it is trained from user's history data and users have similar sense. The more user history data, the less bias it will have when looking for similar users and the more accuracy the model will be. And the model will choose the latest M

queries of the user's history data, so it will also reflect the user's interest changing over time. And if there's no history data of user, the framework will learn the user intent from the model generated from the public dataset.

4.2.1 Multimodal Restricted Boltzmann Machine

Boltzmann Machines (BMs) are a particular form of log-linear Markov Random Field (MRF), i.e., for which the energy function is linear in its free parameters. To make them powerful enough to represent complicated distributions (i.e., go from the limited parametric setting to a non-parametric one), we consider that some of the variables are never observed (they are called hidden). By having more hidden variables (also called hidden units), we can increase the modeling capacity of the Boltzmann Machine (BM). Restricted Boltzmann Machines further restrict BMs to those without visible-visible and hidden-hidden connections. A graphical depiction of an RBM is shown below.

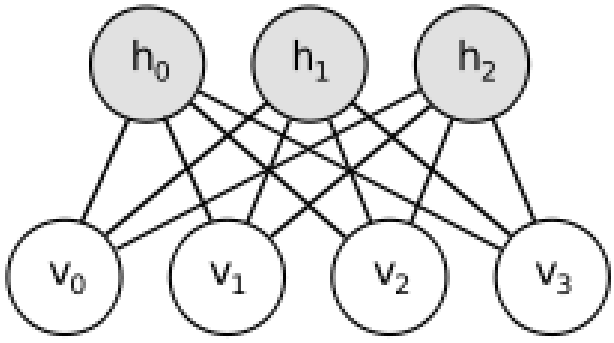


Figure 4.3: Illustration of Restricted Bolzmann Machine

The Restricted Boltzmann Machine (RBM) is an undirected model containing visible layer v and a hidden layer h [67]. Visible layers containing variables that represent observed data, while hidden layer containing variables, which could be features or representations of the visible layer. Connections between visible variables and hidden variables are symmetric, but theres no connection between hidden nodes or visible nodes.

Both visible nodes and hidden nodes are binary units ($v \in (0,1)^D, v \in (0,1)^D$) The RBM define

the joint distribution of v and h based on the following energy function:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F} v_i W_{ij} h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{D} a_j h_j \tag{4.1}$$

Parameters $\theta = a, b, W$ are what the model is going to learn. The joint distribution over hidden and visible units is defined as:

$$E(v,h;\theta) = -\frac{1}{Z(\theta)} exp(-E(v,h;\theta)) \tag{4.2}$$

and

$$Z(\theta) = \sum_{v,h} e^{-E(v,h;\theta)} \tag{4.3}$$

And RBM uses the contrastive divergence[68] to learn the model parameters $\theta = a, b, W$. And use the energy function, the conditional distribution given hidden units or visible units respectively are:

$$P(h_j = 1|v,\theta) = sigmoid(b_j + \sum_i v_i W_{ij}) \tag{4.4}$$

$$P(v_i = 1|h,\theta) = sigmoid(a_i + \sum_j W_{ij} h_j) \tag{4.5}$$

However, in real word task data examples are not always one dimension. For example, query logs, it is not a nature way to combine the URL and text together to learn a RBM model. In this work, we try to model the representations of different modalities of data separately and learn a unified representation by the outputs of the hidden units learned from them. By using different models like Gaussian RBM[69][70], replicated softmax[71], and etc. it can deal with various input type like real value or word count. And even though the model trains the data of different modalities separately and adds another layer to output the general representation, its still easy to be trained by contrastive divergence, and has fast approximate inference [63].

And there're several benefits from the multiple modalities settings. We can deal with data consist of multiple input types, and each input type provide different kind of information and using different

kind of representations. For example, query logs, the query text can be represented as word count vectors while the URL is a binary vector. Without a multimodal configuration, it's much difficult to learn the relationship from data with multiple input types than data with single input type. Moreover, it can be more robust than RBM to data with missing values.

In this work, we develop a multimodal learning framework for multiple input type data by training a RBM over the pre-trained layers for each input modality as shown in Figure 2. The output hidden layer for each input modality is used as input layer to train the higher output layer. Each dot square in Figure 2 is a single RBM trained separately. Parameters for this model is $\theta = (\theta_1, \theta_2, \theta_3)$ and we can learn the parameters separately.

4.2.2 Candidate Named Entity Detection

The "named entity" referred in our work is not like the standard NER (Named Entity Recognition) task. The traditional NER task only looks at phrases that describe specific classes of items like person's name, organization name, locations[72] and etc. However, things that people are interested in are various, and therefore it will make more sense to understand people's intent from a much larger range of named entities. So in this chapter, we design a general statistical based way to learn possible named entities mentioned by people in the search query log.

We first get the word count of all the n-grams in the dataset (n is no greater than 7). And we retain the n-grams whose word count is greater than 5. By doing this, we can get a candidate named entity set C. As you may have notice, the named entities in set C may have overlap. That is to say, for example, a named entity p with 2 words could be part of the named entity q with 4 words. Then here's the question, if p and q are both named entities that refer to two dependent items, or p is incomplete description of q? To solve this problem, we use mutual information in the information theory to measure it. Mutual information is a measure of the variables' mutual dependence. Here, we want to see if p and q are dependent items (like "Harry Potter" is a person's name and "Harry Potter and the Goblet of Fire" is a book name), or not(e.g., "The Lord of the Rings: The Return of the King" and "The Return of the King", both refer to the movie).

There are several ways to measure the mutual information, like Pointwise Mutual Information(PMI)[73],

Table 4.1: Query Text Feature Vectors

Features	**Examples**
bag-of-words	word count n-gram of the query text
punctuation	Abbreviation (end with period, or has internal period) Internal hyphen
named entity lookups	named entity length named entity bag-of-words
part-of-speech	POS for each words in text POS sequence for bi-gram POS sequence for detected named entity POS for the context of the named entity
position	Named entity position
category	Named entity possible categories

Google Similarity Distance(NGD)[74], Dice[75], Jaccard Distance and etc. In this work, we use PMI to measure the similarity of two candidate entities with overlaps. Pairs with larger PMI value will be regarded as one item and the shorter one will be removed from the candidate set C.

We use the basic text mining features as query input visible variables[64] as shown in Table 4.1.

4.2.3 User Intent Mining

We use the user's historical query logs to predict the future click-behaviors. However, sometimes user's query logs information is not enough for training the model. A practical solutions in previous studies is to develop group level personalization, which combines limited individual information with other related use. But to find similar users to enrich personalization is also challenging because of the data sparsity and have to compute the distance among each user. Here, we use the Restricted Boltzman machine to learn the activity representations for each user, and similarity calculation with such kind of representation is convenient. And then, we use the users activity logs to train a predicted M-RBM model for user behavior prediction.

The key idea is to learn a joint density model over the space of multimodal inputs. Missing modalities can then be filled-in by sampling from the conditional distributions over them given the observed ones[63]. And by drawing the samples from $P(v_{url}|v_t)$

The usage of the public dataset is two folds: 1)user model, and 2) query models. For user model,

treat each user's entire query log as one document, we use the count of occurrence for both terms and URLs as M-RBM visible layers and train a joint representation for each user. For query layer, we treat each query as a training sample and train a M-RBM for behavior prediction.

No History Record User Intent Mining For people with no history records, we dont do a personalized recommendation, but make use of the public model to predict the user intent, and showing the link the majority of people clicked on the top.

With History Records User Intent Mining When the history query logs of the user are available, the model learning process follows the following steps. 1) We use the user's existing entire query data to get the representation with the model learn from public dataset. 2) Find similar users with the representation, simply calculate the squared Euclidean distance. 3) Try a M-RBM for the user for prediction.

4.3 Experiment and Evaluation

We use AOL query log dataset to conduct our experiments. For each data example in this dataset, the format is a, starting with sectionID, followed by query text, query time, the URL user clicked and the ranking of URL in the entire return documents in one page. There are around 36M click-through data in the dataset. To verify the performance of user intent mining, we choose four classes of data to do the experiments. They are "movie", "actor", "book" and "digitals". We collect related items from websites such as Wikipedia, IMDB, and Amazon.

We conduct experiments to verify the performance of our method. In this section, we will show both the qualitative and quantitative evaluations by our method. There's mainly two parts in model training, which is query modeling and user modeling. To test the query modeling, we illustrate a qualitative result in Table 4.2. We show the top ranked context for each class. The class names are assigned by human judger and to illustrate the results, we omit the context with stopwords. In the table, # represents the position of named entities. And in Table 4.3 , we show the top ranked websites for each class.

To test the user modeling, we conduct the experiment for user interests prediction. We collect 30 users with more than 30 query logs for test. We use the whole dataset(not include this 30 users) to train the user models. Each user is represented as one sample. The input of M-RBM is the occurrence of query terms and clicked URLs. We can get the model parameters from the training process. Then for each user in testset, we use 5, 10, 15 history queries to learn the representation with learned parameters. And calculate the most similar users for query modeling. After that, a new query is used to evaluate the prediction. We run our prediction model to get a ranking list of websites. Then we calculate the precision of the result with P@3, P@5, P@10 and P@15. The result is shown in Table 4.4.

Table 4.2: Top ranked Context of Classes

Movie	**Actor**	**Book**	**Digitals**
# dvd	# movies	# by	games for #
# movie	# bio	book #	# games
# cast	pictures of #	the book #	codes for #
# trailer	# pictures	# spark notes	# game
# on dvd	# films	quotes from #	free #
# soundtrack	# biography	# summary	# cheat
# dvd release	# interview	# quotes	# ringtone
# wallpaper	# pics	# chapter	# cell phone
# quotes	# baby	notes #	# wireless
# imdb	# wallpaper	author of #	# reviews

We can see that, most of the results are reasonable context and websites for each class. And this result, can to some extends, shows that the model can learn the category information according to the both text feature of query itself as well as the URL information. For movie, the context contains information about movie dvds, movie trailers and movie wallpaper, and imdb.com and rottentomatoes.com are highly ranking are also consistent to people's searching behaviors.

The evaluate measures we used in this work is the *Correctness@topK*,short for *C@K*, which is defined as equation 4.6.It is illustrated how many users get correct prediction at top k return predictions.

$$Precision@K = \frac{\#users\_getCorrectPredictionAtTopKDocument}{\#users} \tag{4.6}$$

Table 4.3: Top ranked websites of Classes

Movie	Actor	Book	Digitals
www.imdb.com	www.imdb.com	www.amazon.com	www.amazon.com
www.rottentomatoes.com	www.starpulse.com	www.novelguide.com	www.gamespot.com
movies.yahoo.com	www.femalefirst.co.uk	www.sparknotes.com	www.gamezone.com
movies.about.com	www.msnbc.msn.com	www.cliffsnotes.com	www.cheatcc.com
www.allmoviephoto.com	www.thesuperficial.com	www.bookrags.com	www.gamewinners.com
www.youtube.com	people.aol.com	www.online-literature.com	www.boxcheats.com
www.amazon.com	en.wikipedia.org	www.enotes.com	reviews.cnet.com
en.wikipedia.org	www.celebritywonder.com	en.wikipedia.org	www.gamefaqs.com
www.movieweb.com	www.defamer.com	www.pinkmonkey.com	www.neoseeker.com
www.filmsite.org	abcnews.go.com	www.homework-online.com	www.mobiledia.com

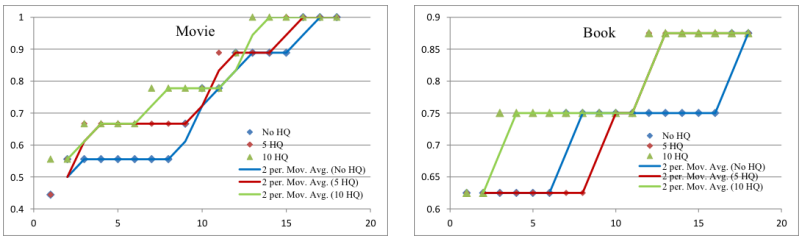
Table 4.4: Precision of User Intent

	P@3	P@5	P@10	P@15
No HQ	0.567	0.667	0.700	0.867
10 HQ	0.600(5.8%)	0.733(9.9%)	0.767(9.6%)	0.933(7.6%)

According to the results, firstly, with no historical data provided, 46% of query can get a proper URL on the first returned document. And two thirds of queries can get the websites that meet the users' intent on the top 5 documents. And as the user have more historical data, here we have 10, the performance of the prediction will improve the 7% of precision on P@1 and 4.9% of precision on P@5. And when the user's data increase to ten, the precision will increase 9.9% on P@5, 9.6% on P@10 and 8% on P@15 respectively. But as shown in our experiment, when only 5 history data provided, the performance is not as good as only use the public dataset. The result could be the few history data can not get accurate similar users group to complement the insufficient of training samples, however, it will bias the user understanding results with some not related users' data.

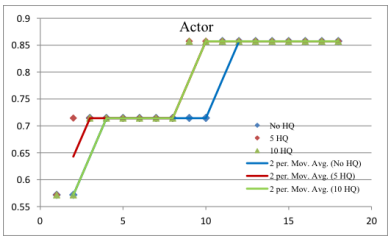
4.4 Conclusion

In this chapter, we propose a new framework to understand user's intent from search engine query log. The model can identify the category of named entity by learning from the massive amount of public click-through data as well as the user's history click through data. The model is a general framework for data with different modalities. Thus, it can be used to scenario with multiple input data types, like text and image, text and audio, etc. Moreover, the model is flexible to situations

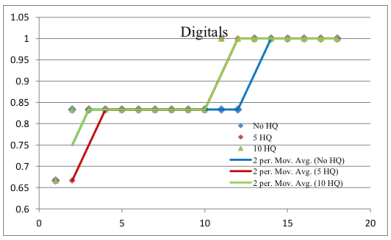


(a) Movie

(b) Book



(c) Actor



(d) Digitals

Figure 4.4: Impact of user's history data for different named entity categories. (Blue line for No HQ, red line for 5 HQ and green line for 10 HQ)

normal search or personalized search. It can get a 66% precision without personal data at P@5, and it will improve 5% to reach 70% if personal data is provided. As for future works, we will try to build a repository of named entity context and websites of different named entity categories, to facilitate further mining of the click through data. And we will also try to apply the model to different problems, like question and answering, image tagging and speech recognition.

Chapter 5: Dynamic User Intent Mining for Online Forum

Online forum or discussion board plays an important role in information sharing and spreading for many years. A critical issue for online community service is to keep active users and encourage users to create high-quality contents. Intuitively, one of the most important motivations for users to stay on an online community is the capability of efficiently obtaining target information. So for the good of both online community's prosperity and user's information needs, it's necessary to hold the ability to predict what information the users will respond to. This task has two close study domains. One is the recommendation on the social network, and the other is user profile construction. But sometimes social network relations are not available on some online communities, making many social relation-based models cannot be applied. And most of the user profile construction methods learn the user's interested topics from summarizing user's existing events, without considering the user interests evolution over time. In fact, user's interest topics are often changing over time, and sometimes they themselves cannot clarify what they are looking for.

In this work, we define the motivations of the user response processes on online forum as spontaneous action, self-exciting response and cross-exciting response. And we propose a categories-association enriched self-exciting point process framework to model the user's interest topics evolution over time by learning the latent evoking process of the events happened on user timeline. We use a branching structure to represent the evoking process and use EM to infer model parameters. We collect a Reddit corpus with user's historical event and evaluate our method on it. The experimental results show our methods outperform the several baseline methods.

5.1 Introduction

Online Forum always plays an important role in information generating, obtaining, spreading and sharing. From the platform's side, it's critical to maintain high daily active users. And for users, they hope to target their interested information efficiently. So for the good of both online community's

prosperity and user's information needs, it's necessary to understand the user's interests either explicitly or implicitly, to provide the highly related and interesting information to the target user.

However, user interests understanding and modeling is always a challenging task. Different people have very disparate interests. And even for one individual, his/her interest is always changing over time. In addition, one's interest is not exclusive; however, it's often true that individual has multiple interests at the same time. But the attention degree for each interest at a specific time is different. Suppose Andy is a Reddit user. He is a super sci-fi fan who never misses any super-hero movies; he like traveling with family, and he also has a dog. According to Andy's Reddit timeline(records for all his activities on the platform with time stamps), sometimes he thumbs up other's pet photos. When new Marvel movie is released, he will discuss the hero stories with others. Thus, Andy's interests may contain sci-fi, pet, and travel. But proportions of the three categories are not always the same. When "Captain American" released, his interests of sci-fi is greatly increased. Moreover, the interest persistent over different topics is also different. User's attention to news or event may vanish shortly after several days. But for a book fan of "Harry Potter", continuously following the related novels, games, movies, and events for years are highly possible. All the conditions make the user interests modeling and understanding a tough task.

The most common strategy is to rank the hottest and latest posts on top. This strategy works pretty well for users who are just hanging around to see what's happening but cannot serve the users with interests in specific domains well. Most users may read it, but the "hot topic" may not give enough motivations for users to reply to it, thus have no contribution to the platform's content. And another way is to do personalized recommendation. Content interesting to your friend will more likely be shown on your front page and people share interested in similar items will more probably become friends on social network. This collaborative filtering based strategy works on social networks. However, many online forums are content centroid, like Reddit, whose social information is not available or unreliable.

The motivations for users' response behaviors are complicated, and such a topic is out of the scope of this work. In this chapter, we regard the motivations relating to internal and external

factors. The internal factors are related to users, such as personalities. Some people are willing to share ideas and interact with others, while others are not. The external factors may be related to the content, topic, etc. Firstly, user response frequencies are various on different topics. It depends on user‘s interestedness to the topic and specialization level of the topic. Intuitively, the more interested to the topic the more likely the user may reply to it. As for specialization level, the high specialized topic needs user‘s specific experiences to the topic and even knowledge and study. For example, in Reddit, everyone may leave a comment on a picture posted under “funny” subforum. But specific knowledge is necessary for response a post about RNN under “deep learning” and only person with related interests and information need may respond to. Secondly, user’s interest persistent over different topics is also different. User’s attention on a news or event may vanish shortly after several days. But for a book fan of “Harry Potter”, continuously following the related novels, games, movies and events for years are highly possible. Thirdly, user’s interests can be evoked by his previous interests. Although there’re many categories, subforums in online community, the topics of posts under different categories are not exclusive to each other. A fan of Starwars may also be attracted by Marvel’s superheroes. The user’s different personalities, the time decaying phenomenon, and the cross evoking impact between topics make the response prediction problem even more challenging.

User’s interests are always changing over time. An exploration of a topic may result in an in-depth reading about related contents in a time period. And interests of a topic may also vanish as time cumulated. Moreover, in fact, user share several interests at the same time by different proportions or weights. And the percentage of each interest is also dynamically changing. A good representation considering these features can improve the performance of several tasks, such as user interest prediction, user profiling and modeling, temporal sensitive recommender system, etc. However, the existing method of representation can hardly capture the time dynamic nature of the user intent.

In this chapter, we present a statistical model based on the theory of multivariate Hawkes process to make the user response prediction in an online community. Self-exciting point processes were

naturally suitable to model continuous time events where the occurrence of one event can affect the likelihood of subsequent events in the future. The presented method models both of the internal and external factors of user's online forum response motivations, and also integrated the time decay fact of user's interests. Since the model can learn what previous user behaviors stimulate a current time point, it can also show the evolution of user interests over time. To summarize, the contributions of this paper are two-fold:

1. We propose a Multivariate Hawkes Process framework to predict what topic the user is most likely to respond in the near future, considering both internal and external factors of user online response behaviors.

2. We realized the branching structure of the self-exciting process, and it can reveal the evoking relation between responses in user's timeline data.

5.2 Related Works

The goal of user response prediction on online forum task is to recommend the information that the users are most likely interested in at this time. Thus, the problem is closely related to social network personalized recommendation. Many social networks based models[76, 77, 78] have been proposed to improve the recommendation performance. Yang et al[79]. proposed to use a domain obvious circle of friends on social network to recommend user items. Jamali et al. proposed a trust-network based method to solve the cold start problem of collaborative filtering. However, all these social network based recommendation methods rely on the existing of user relationship. It's still not clear if the social relationship submerges the user's personality, especially for the experienced users[79]. And for some content based online platforms, social network is not explicit, and we cannot rely on the relationship information to make the recommendation.

5.3 Problem Definition

Now let's consider a typical scenario of our model. There are K topics and M users in our dataset. Topics are represented as $C = C_i; i \in [1, K]$. Each user issues a sequence of responses and each of the issued response related to one topic. The response sequence of a given user is marked as $T = t_i k$

where $t_{i,k-1} < t_{i,k}, i \in [1, N]$ and t_{ik} is the time stamp of k-th response of user, belonging to i-th topic. Given the users categories of interest according to his response history, and the timestamps, our goal is to predict what topic in the future he may be interested in. In this section, we show how we predict the user's interest based on the response sequences t_{ik} .

Our task also related to user's profile construction or user context recognition. The output of our method could be a list of topics that the user may be interested at this time point. The user profile is a key issue for content-based recommendation[5]. Several studies have been done by summarizing the user event history and construct the user's profile[6, 7]. However, these methods didn't take the user's interest evolution into account, so as fail to capture the user's long-term and short-term interests' influences to their current information selection.

5.4 Multivariate Hawkes Process

A univariate Hawkes process is defined as a self-exciting temporal point process[80], which is suitable to model continuous time events where the occurrence of one event can affect the likelihood of subsequent events in the future. Originally, Hawkes Process is described based on intensity process[81]. A univariate Hawkes process N(t) is defined by formula 5.1.

$$\lambda(t|H_t) = \mu(t) + \int_{-\infty}^{t} \kappa(t-s)dN(s) \tag{5.1}$$

We denote the user's existing response sequences as $H_t = \{t_1, t_2, ..., t_{t-1}; t_{t-1} < t_t\}$, and the probability of an event happening at time t can be defined as the conditional intensity function as $\lambda(t|H_t)$ according to the point process definition. In formula 5.1, μ is the base intensity indicating a background rate, while κ is the kernel function modeling the influence of the past events on the current time stamp of the point process.

In our problem, user's responses in one topic can be seen as a temporal point process. But the happening of the current event is trigger by both previous event in this topic, and events happened in related topics as well. To solve such a more complicated situation, we use the framework of multivariate Hawkes process, which is the multi-dimension extension of univariate Hawkes Process.

The conditional intensity function for topic i is defined in formula 5.2[82].

$$\lambda_i(t) = \mu_i(t) + \sum_{t_{ik}<t} \phi_{ii}(t - t_{ik}) + \sum_{t_{jl}<t} \phi_{ij}(t - t_{jl}) \tag{5.2}$$

μ_i is the immigrant intensity of the Hawkes Process, which govern the frequency at which new immigrants arrive. ϕ_{ii} the self-response function, to model the influence of user's previous response in the same topic to the current event. ϕ_{ij} the cross response function, taking the influence from other related topics into account. The two subscripts denote the topic index, and the event index in topic respectively. We use the kernels defined in the previous part function. Figure 5.1 denotes an example from the multivariate Hawkes Process from a piece of the timeline of a user. The dots denote the user's historical events, and the arrows pointing to dots represent the dependencies between events.

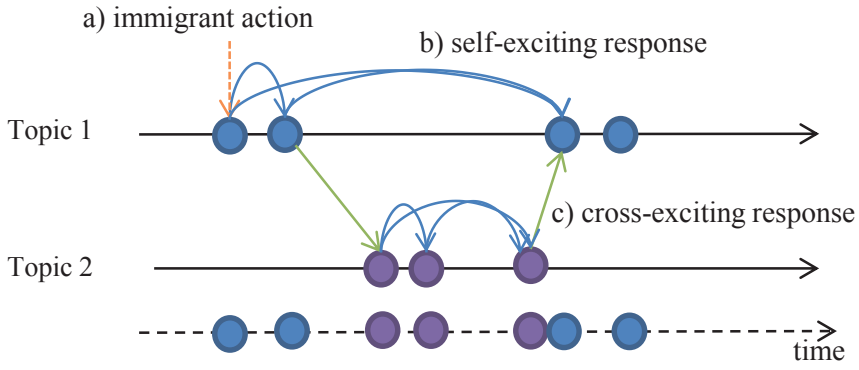


Figure 5.1: Illustration of the dependency among events in user timeline. Immigrant action, self-exciting and cross-exciting responses are denoted in different colors in the figure.

5.4.1 Kernel Selection

Immigrant Intensity

The immigrant function denotes an initial influence of a given topic on the response behaviors to the user. In Hawkes Process, it's the base rate of the intensity function. As illustrated in Figure 5.1, the immigrant function describes the probability of a user choosing to respond to a topic without any influences of his previous actions. In most related studies[5], immigrant intensity is treated as a constant. In our study, for each topic in user's timeline, we learn a constant value μ as its immigrant

intensity.

Response Function

In our problem, the non-spontaneous response of user in topic i at time t is determined by all the previous actions or responses that still have an impact on the user. And these previous responses include the response he made from topic i and the response from other related topics. Thus there are two kinds of response functions: Self-Response function and Cross-Response function. For example, Mike is very active in topic under skiing, and response a lot under posts about ski location and ski equipment. And if we learn a strong association between "ski" and "gopro", then we may predict he also has a high possibility want to buy "gopro" or share his experiences with others.

To simplify, we use the same framework to model both of them. And to consider the association between topics, we add the categories-association factor into the cross response function framework. According to Hawkes Process, the response function Φ_{ij} is defined as the product of the intensity parameter $\alpha_{ij}$and response kernel function $f(\mu;\xi_{ij})$, as shown in formula 5.3. is kernel parameter. is the topic association denotes the influence of topic j to the happening of event in topic i. When $i = j$, Φ_{ij} is the self-response function, and $R_{ij} = 1$.

$$\Phi_{ij}(\mu;\xi_{ij}) = R_{ij} * \alpha_{ij} * f(\mu;\xi_{ij}) \tag{5.3}$$

The response function is design to catch the time decay nature of the point process, and the kernel has parameters to control the function's scale and shape. Intuitively, when a topic attracts us, there's a process we cumulated our interest to the top, and then vanished as the time goes by. Thus, we use a log-normal distribution as the response kernel. Compare to the common used kernel function like exponential and power-law distributions, which have a very strict decreasing assumption, log-normal distribution is more suitable for our situation. The PDF of the log-normal function is written as formula 5.4:

$$f(x;\sigma,\nu) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln(x-\nu))^2}{2\sigma^2}} \tag{5.4}$$

σ and ν are kernel parameters, controlling location and scale of the distribution respectively.

5.4.2 Categories-Association

Because the association between categories is important information for users to find related topics to their interests, we integrate the categories-association information R into the cross response function. R_{ij} learned from temporal information from users entire timeline or many other users by mutual information(MI). The assumption is that temporally close response topics appeared many times by the same user or many other users is more likely to have a close semantic similarity.

$$R_{ij} = I(i,j) = \sum_{i,j} P(i,j) * \ln \frac{P(i,j)}{P(i)P(j)} \tag{5.5}$$

The whole response function of time t influenced previous response at time k in topic j can be listed as Equation 5.6 and the complete intensity function is listed in Equation 5.7.

$$\phi_{ij}(t - t_k; \xi_{ij}) = R_{ij} * \alpha_{ij} * \frac{1}{(t-t_k)\sqrt{2\pi\sigma_{ij}^2}} e^{\frac{(\ln((t-t_k)-\nu_{ij}))^2}{2\sigma_{ij}^2}} \tag{5.6}$$

The parameters to infer are $\theta_i = (\mu_i \alpha_{ij}, \sigma_{ij}, \nu_{ij})$, K+3K*K in total. In this chapter, we adopt an EM to do the parameter inference.

$$\lambda_i(t) = \mu_i + \sum_{t_{ik}<t} \alpha_{ii} * f(t - t_{ik}, \sigma_{ii}, \nu_{ii}) + \sum_{t_{jl}<t} R_{ij} * \alpha_{ij} * f(t - t_{jl}, \sigma_{ij}, \nu_{ij}) \tag{5.7}$$

5.4.3 Parameter Inferences

We use the branching structure representation[82] of the Hawkes Process to conduct the EM inference. In the branching structure assumption, each response has exactly one antecedent. And event with no antecedent is spontaneous action. The four statistical assumption can be another explanation of Formula 5.2.

1. The number of the spontaneous action in topic i over time [0,T] is a Poisson process with rate μ_i.

2. The number of self-response to event t_{ik} over time [t_{ik},T] is a Poisson process with rate ϕ_{ii} (t-t_{ij}).

3. The number of cross-response to event t_{ik} in other topic j over time [t_{ik},T] is a Poisson process with rate $\phi_{ji}(t - t_{ij})$.

4. All of the Poisson process in the above assumptions are independent.

In this scenario, we may first learn each event belonging to which of the three kinds of processes: spontaneous process, self-response process and cross-response process. To illustrate the idea, we may first define the latent variable, to represent the antecede assignment.

$$Z_i = (Z_{i00}, Z_{ii1}, ..., Z_{iin}, Z_{ij1}, ..., Z_{ijn}) \tag{5.8}$$

$X_{ijl} = 1$ indicates the associated event happening in i^{th} process is evoked by the l^{th} event in process j. And $Z_{i00} = 1$ indicates a spontaneous event in process i.

Firstly, initialize the parameter $\theta = (\mu, \alpha, \sigma, \nu)$. And in the E step, use θ to compute $Prob(Z_{ij} = z|X_i, \theta)$ according to Equation 5.9. i=1,..,K

$$Prob(Z_{ik} = z|X_i, \theta_i^{(n)}) = \frac{\lambda_z^{(n)}(t_{ik})}{\sum_{m \in Z_i}^{(n)}(t_{ik})} \tag{5.9}$$

$\lambda_z^{(n)}$ is the intensity function in branching structure representation, which is defined as:

$$\lambda_z^{(n)} = \begin{cases} \mu_i, z = z_{i00} \\ \phi_{ij}(t_{ij} - t_{jl}), z = z_{ijl} \end{cases} \tag{5.10}$$

In the M step, the expectation of log-likelihood function can be writen as Formula 5.11.

$$Q_i(\theta) = \sum_{Z \in Z_i} (\sum_{k=1}^{N_i} \ln(\lambda_z(t_{ij})) * Prob(Z_{ik} = z|X_i, \theta_i) - \Lambda_z(T)) \tag{5.11}$$

According to previous study[83], $\int_0^T f(t; \sigma, \nu) = 1$, then the expectation can be re-write as

Equation (11). And our goal is to maximize the value of it.

$$Q_i(\theta) = \sum_{Z \in Z_i} (\sum_{k=1}^{N_i} \ln(\lambda_z(t_{ij})) * Prob(Z_{ik} = z|X_i, \theta_i) - \alpha_{ij} * R_{ij}) \tag{5.12}$$

By taking the derivative of Equation 5.12, we can get each parameters optimized value, shown in Equation 5.13,5.14,5.15,5.16. $\hat{N_{i00}}$ is the number of events in $(0,t_{ik})$ that are evoked by Z_{i00}.

$$\mu_i = \frac{\sum_{k=1}^{N_i} Prob(z = z_{i00}|\theta_i, X_i)}{T} = \frac{\hat{N_{i00}}}{T} \tag{5.13}$$

$$\alpha_{ij} = \frac{\sum_{z \in Z_i} \sum_{k=1}^{N_i} Prob(Z_{ijl} = 1|\theta_i)}{\sum_{z \in Z_i R_{ij} * \hat{N}_j}} \tag{5.14}$$

$$\nu_{ij} = \frac{\sum_{z \in Z_i} \sum_{k=1}^{N_i} \ln(t_{ij} - t_{jl}) Prob(Z_{ijl} = 1|\theta_i)}{\sum_{z \in Z_i} \sum_{k=1}^{N_i} Prob(Z_{ijl}=1|\theta_i)} \tag{5.15}$$

$$\sigma_{ij}^2 = \frac{\sum_{z \in Z_i} \sum_{k=1}^{N_i} (\ln(t_{ij} - t_{jl}) - \nu_{ij})^2 Prob(Z_{ijl} = 1|\theta_i)}{\sum_{z \in Z_i} \sum_{k=1}^{N_i} Prob(Z_{ijl}=1|\theta_i)} \tag{5.16}$$

After obtaining the parameters θ, we can calculate the likelihood for each topic by Equation 5.12, and the top-ranked topic is the prediction according to our analysis.

5.5 Experimental Evaluation

5.5.1 DataSet

We examine our model on user comment data from Reddit. Reddit is an online community with news, information sharing network. Users can submit contents such as create posts, comments, votes to communicate with others. Contents are organized by topic of interest named "subreddits". Posts with highest user responses appear on the main page or the top of the subreddit. The subreddit topics are classified into numerous categories, including news, gaming, movies, music, books, fitness,

food, and art. As of Feb 2016, more than 800k subreddits are created by Reddit users. Thus, as we can imagine, a lot of information and content can hardly be reached by users. We have collected a corpus of 523 users timeline data, each of them is active users with more than 900 events on their timelines. This corpus contains 493k user responses under around 5k categories, related to 298k posts.

5.5.2 Baseline Methods

We compare the proposed method to the following baseline methods:

Reverse Chronological baseline (RC): This method ranks the user events on his timeline by most recent timestamp. Intuitively, users interested topics will be consistent in short time slot. Currently, many online communities are still using this method to sort new seeds[84].

Frequency Ranking baseline (FR): Intuitively, topics a user response frequently in the recent period of time may still attract the user to follow and respond. So we use $freq(c) = \frac{category_o ccurranc}{Total_o ccurrance}$ to rank the topics. And the top frequent topics are more likely to be a response by the user.

Hawkes Process Method (HP): To test the performance of our Multivariate Hawkes Process model, we take the framework from equation 5.2 and user the kernel function. Evaluating this method can show if the user interest evolution can help predict the user's interest in the future.

Categories-association Hawkes Process (CAHP): Add a categories-association factor to the cross-exciting function to the multivariate framework.

5.5.3 Experiments and Result Analysis

To evaluate the performance of our methods, for each run of the experiment, we use a random number to select a time stamp in a user's timeline as the future event we are going to predict. And we take the previous user event history before the selected time stamp to do the prediction. We can conduct two sets of experiments. One is to predict what is the next topic the user most likely to comment, vote or post. And the other one is to predict a possible topic he may be interested in within a short period of time in the future. Here, we set the time as one hour, and one day. The experiment results are shown in Table 5.1 and Figure 5.2.

The evaluate measures we used in this work is the *Precision@topK*,short for *P@K*, which is defined as equation 5.17. It is illustrated how many users get correct prediction at top k return predictions.

$$Precision@topK = \frac{\#users\_getCorrectPredictionAtTopKDocument}{\#users} \tag{5.17}$$

Table 5.1: Performance of P@K by 4 Methods, evaluated by difference predict time

Time	Method	P@1	P@2	P@3	P@4	P@5
NP	RC	0.3557	0.4171	0.4743	0.5014	0.5329
	FR	0.3686	0.4629	0.5014	0.5500	0.5886
	HP	0.4760	0.5813	0.6563	0.6896	0.7135
	CAHP	0.4698	0.5865	0.6479	0.6969	0.7198
HR	RC	0.4210	0.4814	0.5352	0.5657	0.5943
	FR	0.4371	0.5332	0.5871	0.6114	0.6386
	HP	0.6652	0.8000	0.8522	0.8696	0.8957
	CAHP	0.6955	0.8364	0.8591	0.8773	0.9000
DY	RC	0.5871	0.6686	0.7214	0.7533	0.7814
	FR	0.5614	0.6778	0.7286	0.7712	0.7986
	HP	0.7515	0.8576	0.9000	0.9273	0.9455
	CAHP	0.7824	0.8765	0.9118	0.9294	0.9714

According to the evaluation result, the proposed method outperforms the common used reverse chronological method and frequency ranking. Our experiments are conducted in three groups, to predict the next interesting topic (NP), to predict the topics interested in next hour (HR) and on the following day (DY) respectively. And for all the four methods, we evaluate the precision at top 1 to top 5 return results. Both the Multivariate Hawkes Process (HW) and Categories Association enriched Hawkes Process (CAHW) outperform the common used baselines. And we also show the increasing percentage of CAHP over the three other methods in the three groups of experiments in Figure 5.2.

As shown in Figure 2, both of the two proposed model based on exciting processes (HW and CAHW) increase the performance of the two baselines by 22%-40%. But the association enriched method does not significantly improve the performance compared to HW in NP experiments. One possible explanation is, when we explore information on an online community, we may first deeply immersed in the topics that we are interested in, without considering to find other related informa-

tion. So the category association information may not contribute to the prediction. But according to the result in the other two sets of experiments, which allow matching the predicted result in one hour and one day. The association information improves the predicting results of HW.

5.6 Conclusion

In this chapter, we proposed a dynamic user interest prediction model on online community platform. Compared to previous related works, our method mainly has 2 advantages. First, it implements Multivariate Hawkes Process as the framework, modeling the user response process on online forum directly, combining the immigrant action, and self-exciting and cross-exciting process. And as shown in evaluation section, it outperforms the baseline methods. Second, the proposed method can also show correlations between topics and reveal the underlying factor of user's response motivation.

For future work, there are mainly two directions. First, the current model works on data with timestamps and response categories to predict the user's future behaviors, without considering the content of responses. We may integrate the content information, such as entities, key phrases to the model and it may reveal more underlying user's interests. Second, we may apply the model to social networks with social relations, such as Twitter or Facebook, to compare the user interests' evolution patterns between different online communities or platforms.

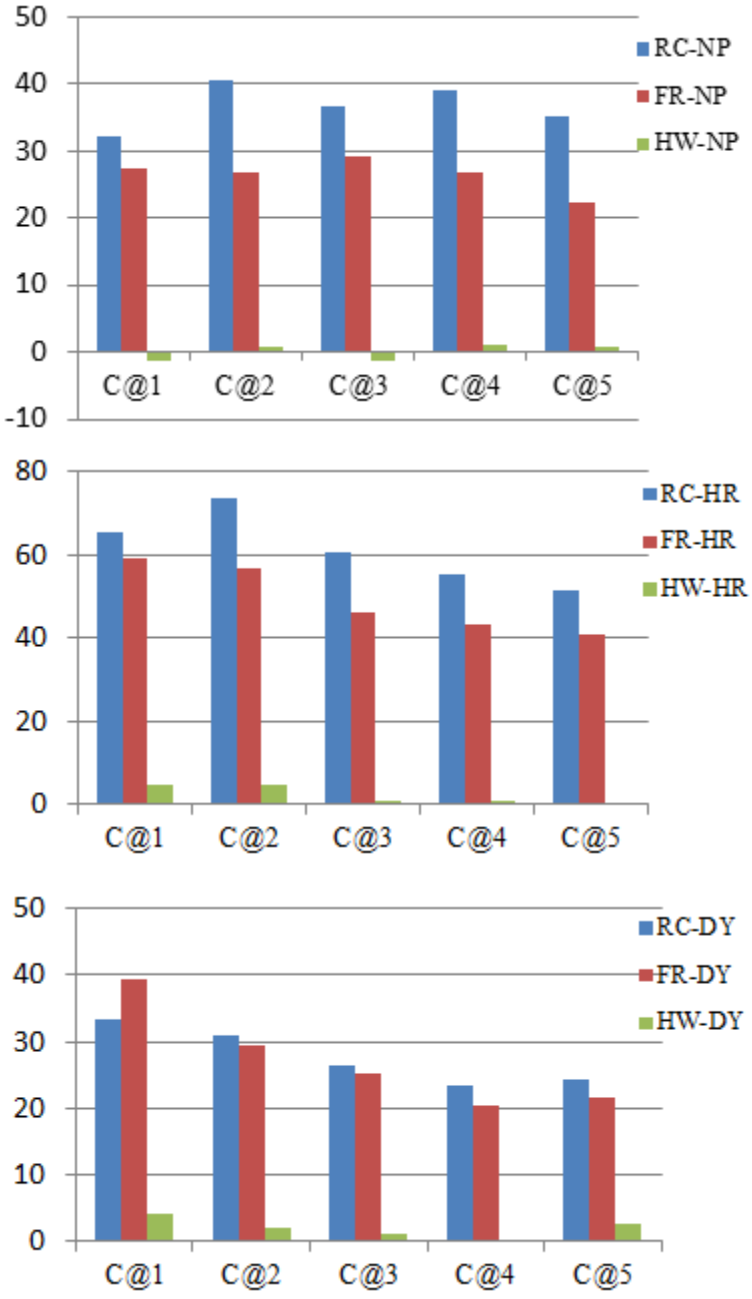


Figure 5.2: Increasing percentage of CAHP over the other three methods in NP, HR and DY experiments

Chapter 6: Intent Visualization using Enriched Domain Ontology

Ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that exist in a particular domain. Using domain-specific ontologies to annotate a dataset for exploring, analyzing and integrating the entities and relations within the corpus is an important step for further study of the data. In this chapter, we present an ontology-based entity annotation system to annotate entities in neuroscience documents. To improve the annotation performance, we propose an ontology entity expansion method based on web service and ontology structure. We evaluate the proposed entity annotation method on real data obtained from Elsevier's BrainNavigator. The results show that using web service and ontology structure to expand ontology entities can improve the annotation result.

6.1 Introduction

The study of the human brain has received a tremendous boost in recent years. With the exponential explosion of neuroscience journals, articles and data, it is necessary to develop tools to explore, compare, combine and integrate the extensive and growing neuroscience data. Information visualization is a good way to quickly acquire knowledge in a specific domain by showing the hot topics and their relationships. However, accurately annotating the entities within the corpus is a crucial step for achieving a good performance. In this chapter, we propose an ontology based entity annotation method. The key idea to improve annotation performance is populating ontology entities from unstructured text using both web service and ontology hierarchical information.

In the biomedical domain, ontologies are often used to support semantic queries, name entities recognition and data integration [85, 86, 87]. However, a large number of new entities and concepts continue to emerge due to the exponentially increment of biomedical literature. Keeping ontologies up-to-date with extensive coverage is important for ontology-based applications. Manually building and expanding ontologies is a time-consuming task, which requires considerable human efforts. Thus,

developing an automatic way to expanding the ontology instance is an important research topic. If candidate entities could be automatically recommended for ontology expansion, we can save manpower and time of ontology maintenance. Another consideration for ontology expansion is that the entities within the ontology used for annotation are professional and domain specific that they are not often used in writings. For example, "grey matter" is an important component of the central nervous system, including regions of the brain. But this entity does not involve in NIF (Neuroscience Information Framework) gross anatomy OWL[88] data. However, "grey matter" is often shown in the article discussing human memory and speech function of brain[89]. Without entity population, it is hard to annotate "grey matter" in documents. Thus, it is necessary to expand the ontologies in order to recall more entities to improve the performance of name entity recognition. Moreover, because of the massive amount of documents and the rapid growth rate of knowledge, it is difficult to access, process and analyze each document directly. Web service like online repository and search engine provides an effective and efficient interface to acquire and utilize existing documents. Thus, web service is a feasible way to facilitate the ontology expansion process by bridging the semantic gap between ontology entities and candidates. In this chapter, we present an entity annotation method on neuroscience data using web service and ontology hierarchical information to expand the ontology entity set in order to find more useful entities from unstructured documents. The rest of the chapter is organized as follows. Section 2 presents the related works of ontology expansion and entity based annotation. Section 3 describes the method we use for ontology expansion and entity annotation. Section 4 shows the evaluation and result analysis for annotation and ontology expansion respectively. The last section presents the conclusion of this work.

6.2 Related Works

With the evolution of semantic web, many studies have been carried out to solve the annotation problem of Web raw text in an ontology-based manner. Most of them use word-net and Wikipedia as key mean for entities annotation task[90]. Many tools are developed to extract entities and relations from web pages, like KnowItAll[91], TextRunner[92, 93], SEAL and Text2Onto. KnowItAll and TextRunner adopt an open information extraction method, which use redundant information from

web documents to perform a bootstrapping information extraction process.

Using named entities to populate ontology has attracted a lot of attentions recently, and there are two main kinds of approaches: pattern-based and distributional[94]. The pattern based method leverages lexical or syntactic patterns to extract patterns like “is -a”. Weakly supervision is often applied in form of a small set of human made seed patterns or seed instances[95, 96]. The distributional method uses context as evidence to find features for ontology population. A small seed set of entities are also required for new entities exploration. Our method aims to deal with entity annotation problem within biomedical domain. To obtain a better annotation performance, ontology is expanded using web services and ontology structure information.

6.3 Proposed Methods

To implement the ontology-based entity annotation, two main components consist in our method: ontology entity expansion and entity annotation. Professional and specialized entity terms in ontologies may limit the annotation performance. It is necessary to expand the ontology entities with the phrases with similar semantic meaning that are frequently used in writings as well. Thus, we proposed an ontology entity expansion method here using web services and ontology structure. And then, an entity annotation process is implemented with the expanded entity set. The whole process of our method is shown in Figure 6.1.

6.3.1 Ontology Entity Expansion

The goal of ontology entity expansion is to recognize more named entities that are semantically related to the existing ontology entities in order to improve the performance of entity annotation for unstructured corpus. To fulfill this goal, new entities need to be detected from unstructured text. Ontology entity expansion can be regarded as two sub-tasks: candidate entities detection and candidate ranking.

NIF is a Neuroscience information framework, which contains web-accessible neuroscience resources, an expanded and integrated terminology, and a framework for concept-based queries [97]. NIF ontology is composed of a collection of OWL modules covering distinct domains of bio-medical

reality, such as anatomy, molecule, disease, and organism, etc. However, NIF ontology is not broad enough to cover most of the entities used in the neuroscience publications. Thus, to obtain a better performance for entity annotation, an ontology entity expansion method is applied to broaden the thesaurus of NIF ontology. We present a two-step method here. Firstly, candidates are selected according to patterns of part-of-speech tagger results. Then by calculating the Web service based similarity between entities, a collection of entities with high similarities with ontology entities is obtained to expend the thesaurus of ontology. We rank the candidate entities according to the similarity as well as the ontology hierarchical information.

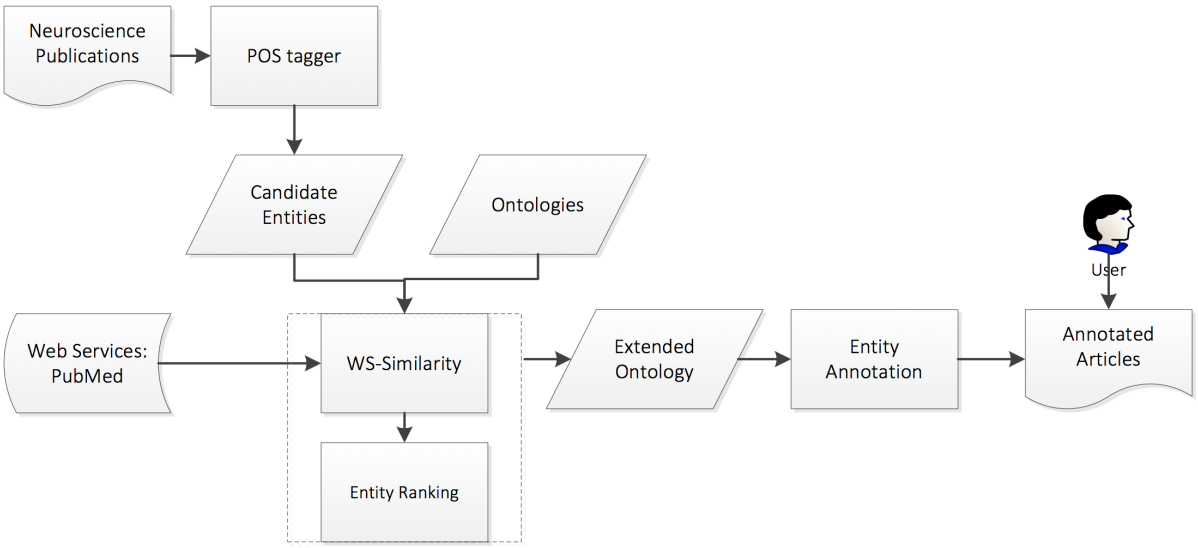


Figure 6.1: General Flowchart for Neuroscience Entities Annotation

To obtain the entities within NIF ontology, protg API is used to parse the owl file of ontology. Protg is an open source toolkit that allows researcher to build, alter and search ontologies[98].

Candidate Entity Detection

To obtain candidates, we use the POS tagger in OpenNLP toolkit[1], whose English POS model uses the Penn Treebank tag set. After the text has been tagged, expressions with patterns shown in Table 6.1 are used to detect the noun phrases, which may contain the full name of an ontology entity. Three patterns for candidate entity detection are used in this chapter as shown in Table 6.1. The first pattern is about sequential noun phrase ("Brain Regions"). The second pattern is

[1]OpenNLP: http://opennlp.apache.org/

Table 6.1: Candidate Entity Detection

Name	Pattern
Noun Phrase	`(NN|NNS)+`
Proper Noun Phrase	`(NN|NNS)*(NNP|NNPS)+(NN|NNS)*`
Adj None Phrase	`(JJ|JJR|JJS)+(<Noun Phrase>|<Proper Noun Phrase>)`

composed by a central component of one or several Proper Nouns, with no or several Nouns followed or leaded ("insula cortex"). The third pattern is composed by noun phrase or a Proper noun phrase leaded by one or more adjective ("White Matter", "Premotor Cortex").

Web service based Similarity

We propose a method that integrates the search result from PubMed with co-occurrence measurement methods like PMI, Dice to calculate the semantic similarity of the candidate entities with existing ontology entities. Calculating semantic similarity between entities according to search engine results has been explored since the pioneering work of Turney[73]. Turney defined point wise mutual information (PMI) measurement using the number of hit result returned by a Web search engine to recognize synonyms. Recently, using Web service as a live corpus has become an active research topic. Google and Wikipedia become popular choices to conduct the similarity measures. Normalized Google Distance (NGD)[74], make use of the number of hits returned by Google to calculate the semantic similarity between terms. Many following studies about word semantic similarity measuring[99], ontology matching[100], tag ranking[101], and others are followed up using the count of hits from Google and Wikipedia.

In this chapter, we focus on dealing with neuroscience entity annotation problem so most of the entities are domain specific. Thus, we use PubMed as the online repository to get the co-occurrence hit count for calculating semantic similarity. Our goal is to extract biomedical entities from unstructured text. PubMed is a web repository for biomedical literature. Firstly, the candidate terms obtained from the previous section will be scored according to semantic similarity measures with the existing ontology terms. Four measurement approaches are used here: PMI[73], Dice[75], Jaccard, and NGD[74]. Since we have received a great number of candidate terms extracted by

the patterns defined in the previous section, it would be time consuming if we run the similarity comparison with ontology entities with all of them. Thus, we will filter the candidate phrases by the number of hits from web service to take the inappropriate ones out. Here we make an assumption that if a phrase is not a real world entity, web service will not return or only return few hits exactly match the phrase. Then, we used the candidates remained after filtering as reasonable entities and calculate the similarities using the following formulas.

$$PMI(X,Y) = \log \frac{P(X,Y)}{P(X) * P(Y)} = \log \frac{CO(X,Y)}{O(X) * O(Y)} \tag{6.1}$$

$$DICE(X,Y) = \log \frac{2 * CO(X,Y)}{O(X) + O(Y)} \tag{6.2}$$

$$Jaccard(X,Y) = \log \frac{CO(X,Y)}{O(X) + O(Y) - CO(X,Y)} \tag{6.3}$$

$$NGD(X,Y) = \log \frac{max(\log C(X), \log C(Y)) - CO(X,Y)}{O(X) + O(Y) - CO(X,Y)} \tag{6.4}$$

X and Y represent the query entities. O(X) represents the number of hit count from search engine. CO(X,Y)represents the hit count for the query "X AND Y". The count of hits for "X AND Y" returned by PubMed can be regarded as an estimation of the semantic relations of two terms[73].

Entity Ranking

For each term, we use the average co-occurrence score with all the ontology terms as its score and rank the terms for all these four kinds of scores respectively. We define as the measurement of relatedness for candidate term M with the ontology. N is the number of total entities within the ontology and O_i is an ontology entity. $S(M, O_i)$ represents one of the co-occurrence similarity measures in the previous section. $height(O_i)$ represents the path length from the root of ontology to O_i. The ontology is created in a tree structure. The nodes near the root of ontology tree are more general entities while the more specific nodes are at the bottom. As a result, entities on the top

of ontology are more likely to return more hits from the search engine. For example, "retrosplenial cortex" is a term we found has high similarities with NIF brain structure ontology terms and it has a high score using the following formula.

$$Score(M) = \frac{\sum_{i=1}^{n} S(M, O_i) + \log(1 + height(O_i))}{N} \tag{6.5}$$

Then we rank the candidate terms according to using PMI, Dice, Jaccard, and NGD respectively and four lists of terms are obtained. Finally candidate terms are ranked by the average rank in four list and the top N candidate terms are chosen as new entities to expend the ontology we have to annotate the text.

6.3.2 Entity Annotation

Levenshtein distance is used here to calculate the similarity of two strings. The Levenshtein distance between two strings is the minimum number of single-character operation (three kinds of operation: insertion, deletion and substitution) that can change on string to the other[102]. Lingpipe[2] is used in our application to implement Levenshtein distance. We design the ontology based entities annotation system, called "SemIntegrator", which is an open toolkit that allow users to explore, compare, and annotate documents using ontologies. Figure 6.2 show the interface for our system.

We design the ontology based entities annotation system, called "SemIntegrator", which is an open toolkit that allow users to explore, compare, and annotate documents using ontologies. Figure 6.3 show the interface for our system. This system is a Protg [3] plugin that can create, modify, combine and compare ontologies and use the user input ontologies to annotate documents.

6.4 Experimental Results

We conduct two sets of experiments to evaluate the method proposed in this chapter. In the first part of this section, we evaluate the overall performance of entity annotation by comparing with an online application from Elsevier for brain structure annotation about Neuroscience publication.

[2]http://alias-i.com/lingpipe
[3]http://protege.stanford.edu

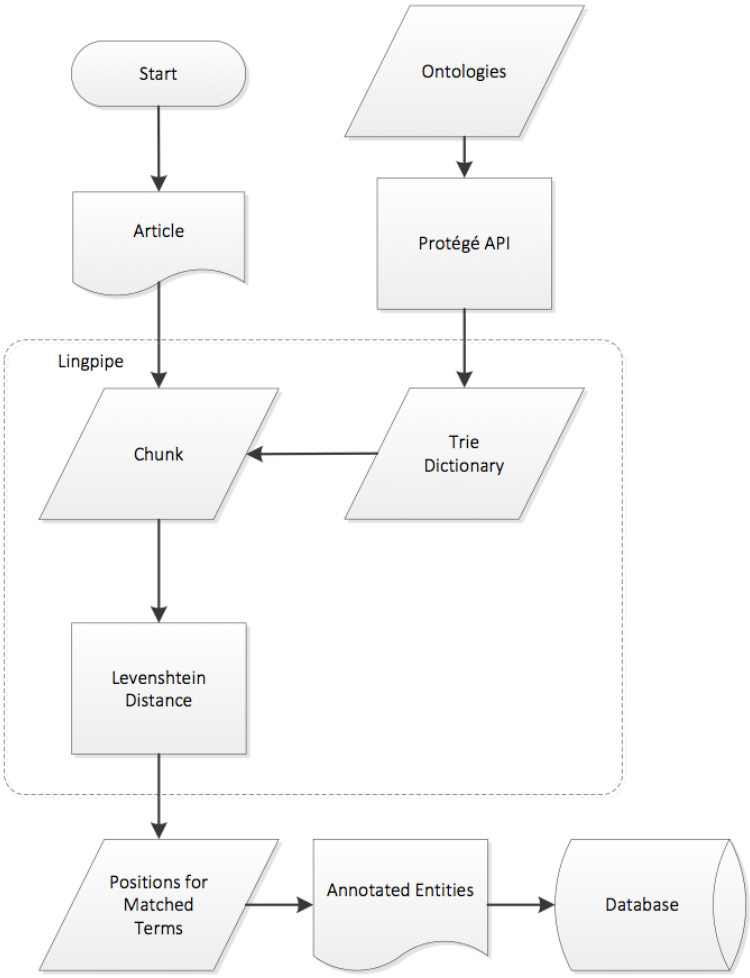


Figure 6.2: Flowchart for Entity Annotation

Next, to evaluate the performance of ontology expansion, we also conduct an experiment on brain structure related ontology.

6.4.1 Experiment for Entity Annotation

As far as we known, there's no dataset for brain related concept annotation. To evaluate entity annotation on real world data, we apply our method on Elsevier Neuroscience documents. The goal of our experiment is to annotate the brain structure entities from the Neuroscience publications with brain structure ontology input by user. The gold standard we use is BrainNavigator, which is an application built and maintained by Elsevier to recognize brain structure in publications. We use the results from BrainNavigator as golden standard for our experiments. Precision, recall and

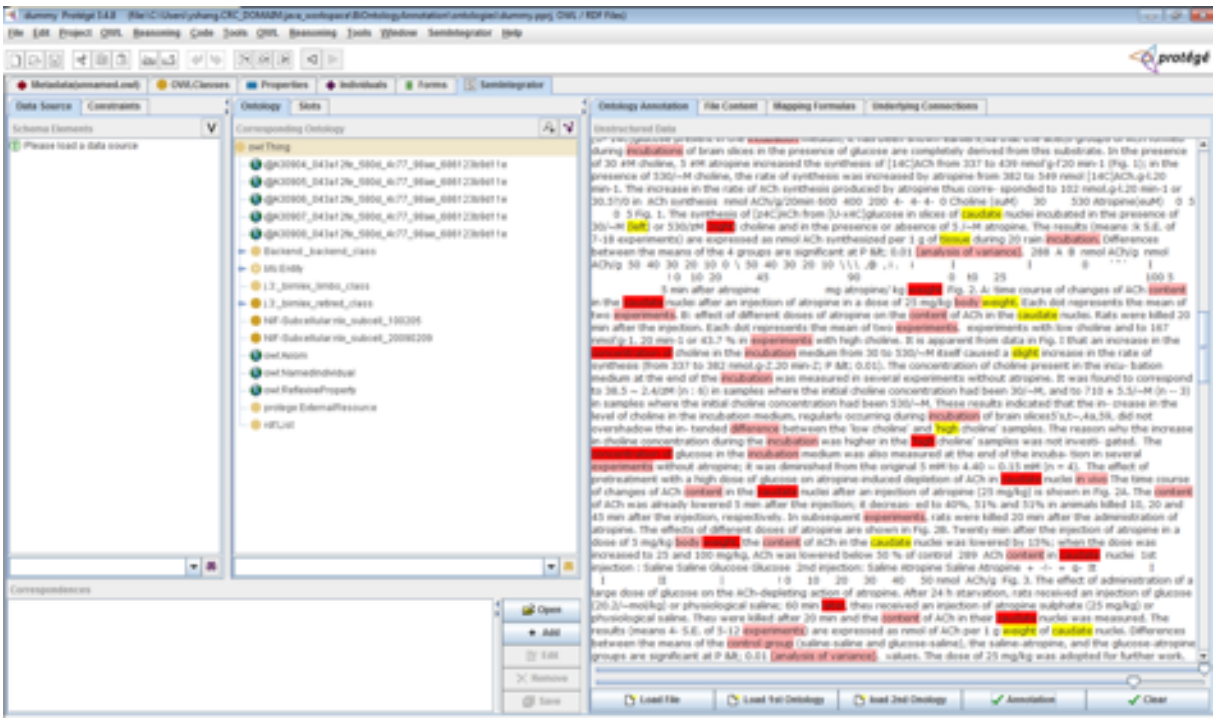


Figure 6.3: User Interface of SemIntegrator for the Entity Annotation System

F-score are used to evaluate the performance of our annotation performance.

Precision, recall and F-score are used to evaluate the performance of our annotation performance.

$$Precision = \frac{TP}{TP + FP} \tag{6.6}$$

$$Precision = \frac{TP}{TP + FN} \tag{6.7}$$

$$FScore = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.8}$$

To annotate the Elsevier articles with our method, firstly we need to select ontology with knowledge of brain structure. There are several ontologies containing brain structure knowledge, like NIF (Neuroscience Information Framework), and Allen Brain Atlas project[103] brain structure ontology

Table 6.2: Results comparison from different methods for Entity Annotation using Elsevier data

Matric	Precision	Recall	Fscore
NIF only	0.7243	0.575045	0.6411
PMI	0.6382	0.6189	0.6284
Jaccard	0.7192	0.6682	0.6928
Dice	0.6842	0.6438	0.6634
NGD	0.7176	0.6678	0.6918
PJDN	0.711	0.6524	0.6806
Proposed method	0.7164	0.6855	0.7006

[4]. In our experiment, we use the brain structure part of NIF ontology, which are the sub-classes of "Regional part of Brain" in NIF gross anatomy OWL. By using Protg API, we obtained the brain structure concepts by reserving all the concepts that are subclasses of "Regional part of Brain" in NIF ontology. Protg is a free, open source ontology editor and knowledge-based framework. It allows the users to operate OWL data and design their own application under protg framework. There are 2567 entities after trimming the ontology file. This is the ontology we use to populate. The unstructured text we used to learn candidate entities are journals of Elsevier NeuroImaging journals. There are 15096 documents from the journal of Neuroimage in the dataset we got from Elsevier and we use these documents to learn new entities.

Six benchmark methods are used to compare the result with our method. The first benchmark only uses the ontology entities to annotate documents. PMI, Jaccard, Dice and NGD are methods we use to calculate the similarity of two entities. PMI, for example, indicates using PMI method to calculate the semantic similarity of ontology entities with candidate entities. And without combining the ranking results of four methods, we only use PMI and get the top N entities as the new entities. PJDN (PMI, Jaccard, Dice and NGD) indicates using the four similarity calculation methods, but without adding ontology hierarchical information. That is to say, in the term ranking step, for each candidate entity, we only calculate its average semantic similarity score with ontology entities without considering the hierarchy information. By comparing this, we can clarify whether ontology hierarchy information contribute to the final result. The results are shown in Table 6.2.

[4]http://human.brain-map.org/ontology.html

Table 6.3: Entities discovered by our proposed method

frontal areas	left temporal cortex
hippocampal areas	middle occipital gyrus
inferior parietal lobe	orbitofrontal gyrus
left frontal cortex	temporal lobes
left occipital cortex	white matter

Table 6.4: Result for Evaluation Ontology Entity Expansion

Removed amount	Number of Entities	NGD	PJDN	Proposed Method
10%	79	43	41	48
20%	158	74	58	73
30%	237	88	78	96
40%	316	126	125	139
50%	395	171	158	182

The proposed method shows the best performance. Comparing with PJDN, the proposed method improves the recall by 2.6%, which indicates that using ontology hierarchy information can improve the performance of entity annotation. For the four kinds of similarity measures, NGD have a highest performance and Jaccard also have a relatively high experimental result. All these methods improve the result of the method using ontology only, indicating that use a web service as a context to populate the ontology entity can find more entities that are related to the ontology and thus improve the annotation performance.

6.4.2 Experiment for Ontology Entity Expansion

We also test the performance of ontology expansion using the previous brain structure ontology from NIF.

Firstly, we randomly remove certain percentage of terms in ontology, in our experiment 10% to 50% respectively. Then, we run our method on the remaining ontology entities to see if our method can expand the ontology by learning from unstructured documents. Then we would like to evaluate the correctness of the new entities our method found. We return the same number of terms as removed from ontology. For time consuming problem, we filter the NIF ontology by PubMed hit count. If a term receives the hit count less than 10, it is removed from the ontology entity set. And after filtered by PubMed, the size of ontology is reduced to 790.

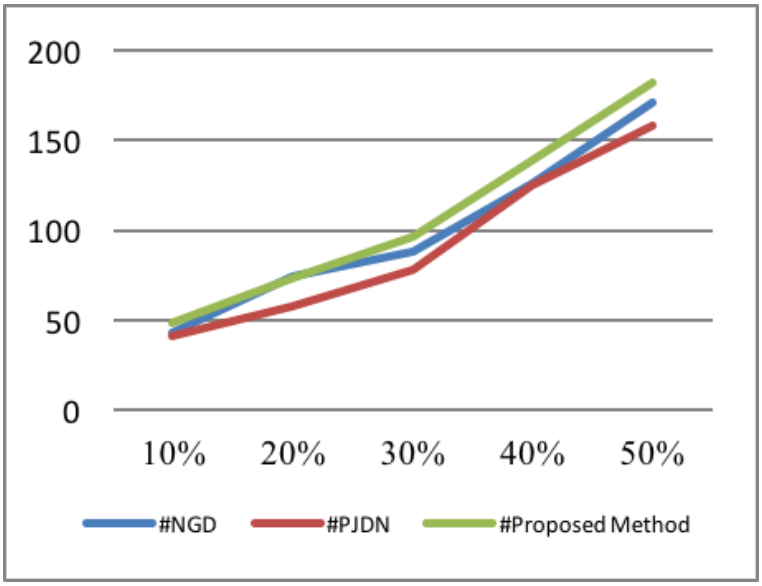


Figure 6.4: Results comparison for correct predict entities from NGD, PJDN and proposed method

The experimental results for our method as well as NGD and PJDN over the NIF brain structure ontology are shown in Table 4. We show the number of correctly learned entities for NGD, PJDN and our method, according to the original ontology entity. From the result shown in Table 4 and Fig 3 we can see that our method can find more correct entities than the other two methods. Both NGD and our method perform better than PJDN, indicating that the ontology structure information can improve the performance. One of the reasons is that if a candidate entity has a high similarity with the entities at the bottom of ontology it would be more related to the ontology. For example, "vivo studies" is a candidate entity we get and it has high similarity with many NIF ontology entities. "Grey Matter" is a central nervous system entity involved in many brain functions. Both two entities do not exist in NIF ontology. We calculate the average height of the related ontology entities for each of them. The average height of the 46 entities that have a similarity with "vivo studies" is 1.36, while for the 125 entities related to "grey matter" the average height is 1.62. This, to some extent, indicates that "grey matter" has high similarity with more specific entity than "vivo studies" does.

6.4.3 Application for Entity Annotation

We apply the ontology based entity annotation method on Elsevier dataset and use the annotated entities to visualize the relationship between entities in Figure 6.5.

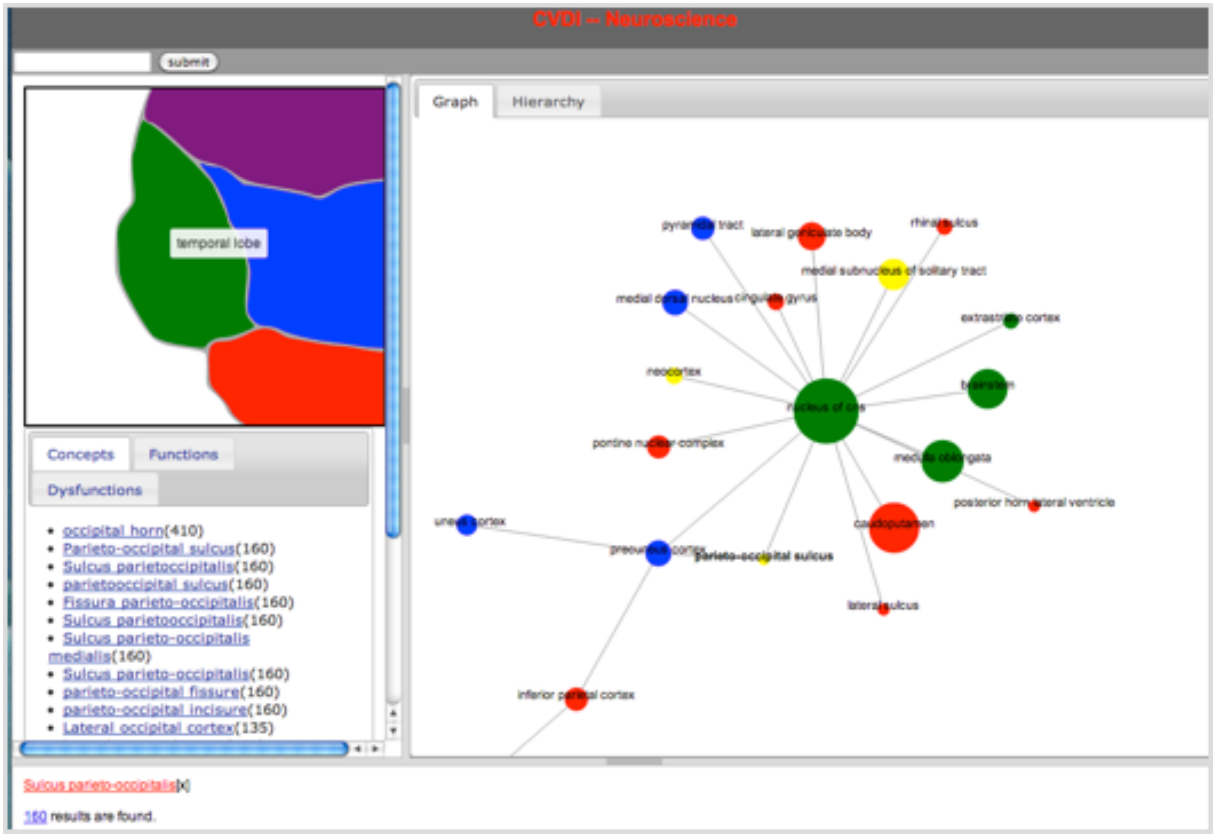


Figure 6.5: Application for Semantic Relation Visualization based on Ontology-based Entity Annotation

6.5 Conclusion

In this chapter, an ontology based entity annotation method is presented using web service as well as ontology structure information. Our goal is to design an ontology based entity annotation tool for users to tag the entities within unstructured text using their own ontologies. The challenge of the work is that ontology entities are domain specific, professional and sometimes hard to cover the entities used by researchers in writings. As a result, synonyms, phrases that have similar semantic content with ontology entities are unable to be detected. To solve this problem, we make use of the web service as an external context to calculate the semantic similarity between entities. And ontology

structure information aids the entities ranking process by adding weights for candidates that related to specific ontology entities. Furthermore, an annotation tool "SemIntegrator" is implemented for entity annotation using ontologies. And based on the proposed method, an entity relation visualization system is designed to illustrate the semantic relationship between entities.

Chapter 7: Conclusions

User intent understanding is a key problem for the applications such as information retrieval, text mining, e-commerce, and recommender system. User intent is a task-specific, predefined or latent concept, topic or knowledge-base, which could bridge the user input, such as a query in search, or status published in the social network, with user's goal. And there's few systematical analysis of user intent mining. In this thesis, we review the previous user intent mining studies in various domains, including how these works define and represent the user intent, what dataset they use and how they model their problems.

From our analysis, the challenges of user intent mining fall into three folds. Firstly, user intent could be express explicitly or implicitly. Implicit user intents do not contain the intent keywords, which is more challenging to classify and recognize users' real ideas. But such kind of implicit express broadly exists in various kind of user-generated content. Secondly, research of user intent in many domains is lacking. As the improving impact of smartphones to our daily life, the resource of our information seeking and the way we express our information need is also changing dramatically. Thus, it's necessary to study the intent of user using cellphone and how they express their information need. Thirdly, we also observed that user intent is not stable but changing over time. Intentions could interact with each other and have a time decaying phenomenon. Then how to model this dynamic nature of intention is also important to predict user's interests and information needs. Based on the challenges we analyze about user intention, we raise four research questions and try to solve them in this work.

In Chapter 3, we study user intentions when they use smartphones while driving. We give a detailed definition of the intention categories and the intention attributes for each category. And a domain dataset is built using crowdsourcing platform and carefully handcraft. We proposed a user intention mining pipeline with general two components, which are intention class classification and intention attribute recognition.

In Chapter 4, we try to model the implicit intent using query clickthrough data. Implicit intents are modeled as the hidden layers of Restricted Boltzmann Machines through an unsupervised manner. And queries context of similar user intent is shown from model output.

In Chapter 5, we learn the dynamic user intent modeling problem using online discussion forum data. Users are modeled as a distribution over multiple intents. A Multivariate Hawkes Process models the evolving of intention and their interactions. And we evaluate the proposed model by predicting user intent distribution given his previous activity timeline. Experiment results show a better performance compare to several baselines.

In Chapter 6, we develop a system to highlight concepts and keywords of a user input article using a domain-specific ontology in an article. And we proposed an algorithm to find ontology-related terms to enrich the ontology's taxonomy, by leveraging the term co-occurrence information and ontology hierarchy.

Bibliography

[1] Andrei Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM, 2002.

[2] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[3] Jinpeng Wang, Gao Cong, Wayne Xin Zhao, and Xiaoming Li. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In *AAAI*, pages 318–324, 2015.

[4] Carol A Taylor, Ona Anicello, Scott Somohano, Nancy Samuels, Lori Whitaker, and Judith A Ramey. *A framework for understanding mobile internet motivations and behaviors*. ACM, 2008.

[5] Daniel E Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM, 2004.

[6] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 68–76. ACM, 2013.

[7] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*, pages 471–480. ACM, 2009.

[8] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[9] Patrick Pantel, Thomas Lin, and Michael Gamon. Mining entity types from query logs via user intent modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 563–571. Association for Computational Linguistics, 2012.

[10] Jim Jansen. *Understanding sponsored search: Core elements of keyword advertising*. Cambridge University Press, 2011.

[11] Nicholas J Belkin et al. Interaction with texts: Information retrieval as information seeking behavior. *Information retrieval*, 93:55–66, 1993.

[12] Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, and Bilal Chebaro. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1732–1736. ACM, 2009.

[13] Bernd Hollerit, Mark Kröll, and Markus Strohmaier. Towards linking buyers and sellers: detecting commercial intent on twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 629–632. ACM, 2013.

[14] Bonnie A Nardi, Diane J Schiano, Michelle Gumbrecht, and Luke Swartz. Why we blog. *Communications of the ACM*, 47(12):41–46, 2004.

[15] Liliana Calderón-Benavides. *Unsupervised Identification of the User's Query Intent in Web Search*. Universitat Pompeu Fabra, 2011.

[16] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592, 2011.

[17] Liliana Calderón-Benavides, Cristina González-Caro, and Ricardo Baeza-Yates. Towards a deeper understanding of the users query intent. In *SIGIR 2010 Workshop on Query Representation and Understanding*, pages 21–24, 2010.

[18] Long Chen, Dell Zhang, and Levene Mark. Understanding user intent in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*, pages 823–828. ACM, 2012.

[19] Guangyu Feng, Kun Xiong, Yang Tang, Anqi Cui, Jing Bai, Hang Li, Qiang Yang, and Ming Li. Question classification by approximating semantics. In *Proceedings of the 24th International Conference on World Wide Web*, pages 407–417. ACM, 2015.

[20] Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1935–1944. ACM, 2014.

[21] Ricardo Baeza-Yates, Georges Dupret, and Javier Velasco. A study of mobile search queries in japan. In *Proceedings of the International World Wide Web Conference*, 2007.

[22] Karen Church, Barry Smyth, Keith Bradley, and Paul Cotter. A large scale study of european mobile search behaviour. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 13–22. ACM, 2008.

[23] Maryam Kamvar and Shumeet Baluja. Deciphering trends in mobile search. *Computer*, 40(8), 2007.

[24] Jeonghee Yi, Farzin Maghoul, and Jan Pedersen. Deciphering mobile search patterns: a study of yahoo! mobile search queries. In *Proceedings of the 17th international conference on World Wide Web*, pages 257–266. ACM, 2008.

[25] Karen Church and Barry Smyth. Understanding the intent behind mobile information needs. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 247–256. ACM, 2009.

[26] Bernard J Jansen, Danielle L Booth, and Amanda Spink. Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web*, pages 1149–1150. ACM, 2007.

[27] Ryen W White, Paul N Bennett, and Susan T Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1009–1018. ACM, 2010.

[28] Justin Cheng, Caroline Lo, and Jure Leskovec. Predicting intent using activity logs: How goal specificity and temporal range affect user behavior. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 593–601. International World Wide Web Conferences Steering Committee, 2017.

[29] Honghua Kathy Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, and Ying Li. Detecting online commercial intention (oci). In *Proceedings of the 15th international conference on World Wide Web*, pages 829–837. ACM, 2006.

[30] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 616–623, 2003.

[31] Zhiyuan Chen, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Identifying intention posts in discussion forums. In *HLT-NAACL*, pages 1041–1050, 2013.

[32] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009.

[33] Jiafeng Guo, Xueqi Cheng, Gu Xu, and Xiaofei Zhu. Intent-aware query similarity. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 259–268. ACM, 2011.

[34] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[35] Ivan Koychev and Ingo Schwab. Adaptation to drifting users interests. In *Proceedings of ECML2000 Workshop: Machine Learning in New Information Age*, pages 39–46, 2000.

[36] I Barry Crabtree and Stuart J Soltysiak. Identifying and tracking changing interests. *International Journal on Digital Libraries*, 2(1):38–53, 1998.

[37] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[38] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization*, pages 1–12. Springer, 2011.

[39] Amr Ahmed, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 114–122. ACM, 2011.

[40] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, and Hongyuan Zha. Identifying and labeling search tasks via query-based hawkes processes. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–740. ACM, 2014.

[41] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[42] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[43] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[44] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[45] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014.

[46] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

[47] Andrew McCallum and Fang-Fang Feng. Chinese word segmentation with conditional random fields and integrated domain knowledge. *Unpublished Manuscript*, 2003.

[48] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics, 2002.

[49] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1):41–75, 2011.

[50] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[51] Jerome H Friedman et al. Flexible metric nearest neighbor classification. Technical report, Technical report, Department of Statistics, Stanford University, 1994.

[52] Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

[53] Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(Aug):975–1005, 2004.

[54] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[55] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[56] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[57] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[58] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *KDD*, 2012.

[59] Gu Xu, Shuang-Hong Yang, and Hang Li. Named entity mining from click-through data using weakly supervised latent dirichlet allocation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1365–1374. ACM, 2009.

[60] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883. ACM, 2008.

[61] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[62] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006.

[63] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[64] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[65] Pengcheng Wu, Steven CH Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao. Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 153–162. ACM, 2013.

[66] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[67] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[68] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[69] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*, pages 599–619. Springer, 2012.

[70] Nan Wang, Jan Melchior, and Laurenz Wiskott. Gaussian-binary restricted boltzmann machines on modeling natural image statistics. *arXiv preprint arXiv:1401.5900*, 2014.

[71] Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[72] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[73] Peter Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *Machine Learning: ECML 2001*, pages 491–502, 2001.

[74] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, 19(3), 2007.

[75] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[76] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[77] Hao Ma, Irwin King, and Michael R Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[78] Mohsen Jamali and Martin Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406. ACM, 2009.

[79] Xiwang Yang, Harald Steck, and Yong Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1267–1275. ACM, 2012.

[80] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[81] Paul Embrechts, Thomas Liniger, Lu Lin, et al. Multivariate hawkes processes: an application to financial data. *Journal of Applied Probability*, 48:367–378, 2011.

[82] Peter F Halpin and Paul De Boeck. Modelling dyadic interaction with hawkes processes. *Psychometrika*, 78(4):793–814, 2013.

[83] Jamie F Olson and Kathleen M Carley. Exact and approximate em estimation of mutually exciting hawkes processes. *Statistical Inference for Stochastic Processes*, 16(1):63–80, 2013.

[84] Mehrdad Farajtabar, Safoora Yousefi, Long Q Tran, Le Song, and Hongyuan Zha. A continuous-time mutually-exciting point process framework for prioritizing events in social media. *arXiv preprint arXiv:1511.04145*, 2015.

[85] Shuai Zheng, Fusheng Wang, and James Lu. Enabling ontology based semantic queries in biomedical database systems. *International journal of semantic computing*, 8(01):67–83, 2014.

[86] James J Cimino, X Zhu, et al. The practical impact of ontologies on biomedical informatics. *Yearb Med Inform*, 2006:124–35, 2006.

[87] Soner Kara, Özgür Alan, Orkunt Sabuncu, Samet Akpınar, Nihan K Cicekli, and Ferda N Alpaslan. An ontology-based retrieval system using semantic indexing. *Information Systems*, 37(4):294–305, 2012.

[88] Stephen D Larson and Maryann E Martone. Ontologies for neuroscience: what are they and what are they good for? *Frontiers in neuroscience*, 3(1):60, 2009.

[89] AKH Miller, RL Alston, and JAN Corsellis. Variation with age in the volumes of grey and white matter in the cerebral hemispheres of man: measurements with an image analyser. *Neuropathology and applied neurobiology*, 6(2):119–132, 1980.

[90] Gerhard Weikum and Martin Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 65–76. ACM, 2010.

[91] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM, 2004.

[92] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[93] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics, 2007.

[94] Marco Pennacchiotti and Patrick Pantel. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 238–247. Association for Computational Linguistics, 2009.

[95] Philipp Cimiano and Steffen Staab. Learning by googling. *ACM SIGKDD explorations newsletter*, 6(2):24–33, 2004.

[96] Michael J Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. Knowitnow: Fast, scalable information extraction from the web. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 563–570. Association for Computational Linguistics, 2005.

[97] Daniel Gardner, Huda Akil, Giorgio A Ascoli, Douglas M Bowden, William Bug, Duncan E Donohue, David H Goldberg, Bernice Grafstein, Jeffrey S Grethe, Amarnath Gupta, et al. The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatics*, 6(3):149–160, 2008.

[98] Daniel L Rubin, Natalya F Noy, and Mark A Musen. Protege: a tool for managing and using terminology in radiology applications. *Journal of digital imaging*, 20(1):34–46, 2007.

[99] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. *www*, 7:757–766, 2007.

[100] Risto Gligorov, Warner ten Kate, Zharko Aleksovski, and Frank Van Harmelen. Using google distance to weight approximate ontology matches. In *Proceedings of the 16th international conference on World Wide Web*, pages 767–776. ACM, 2007.

[101] Dong Liu, Xian-Sheng Hua, Linjun Yang, Meng Wang, and Hong-Jiang Zhang. Tag ranking. In *Proceedings of the 18th international conference on World wide web*, pages 351–360. ACM, 2009.

[102] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

[103] Allan R Jones, Caroline C Overly, and Susan M Sunkin. The allen brain atlas: 5 years and beyond. *Nature Reviews Neuroscience*, 10(11):821–828, 2009.

Vita

Yue Shang

Education

- Drexel University, Philadelphia, Pennsylvania USA
  - Ph.D., Information Science 2017
- Dalian University of Technology, China
  - M.S., Computer Science, 2012
- Dalian University of Technology, China
  - B.S., Computer Science, 2009

Publications

- Shang, Yue, et al. "Scalable user intent mining using a multimodal Restricted Boltzmann Machine." Computing, Networking and Communications (ICNC), 2015 International Conference on. IEEE, 2015.
- Shang, Yue, et al. "Enhancing entity annotation using web service and ontology hierarchy in biomedical domains." Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on. IEEE, 2013.
- Shang, Yue, et al. "Learning to rank-based gene summary extraction." BMC bioinformatics 15.12 (2014): S10.
- Ding, Wanying, Yue Shang, et al. "Video popularity prediction by sentiment propagation via implicit network." Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015.
- Ding, Wanying, Yue Shang, et al. "ProbitUCB: A Novel Method for Review Ranking." Trends and Applications in Knowledge Discovery and Data Mining. Springer, Cham, 2015. 3-15.
- Song, Xiaoli, Yue Shang, et al. "Pairwise Topic Model via relation extraction." Big Data (Big Data), 2014 IEEE International Conference on. IEEE, 2014.
- Ding, Wanying, Yue Shang, et al. "Method and system for ranking media contents." U.S. Patent Application No. 14/445,220.
- Liu, Mengwen, Yue Shang, et al. "Method and system for multimodal clue based personalized app function recommendation." U.S. Patent Application No. 14/805,830.

Teaching Experience

- **Teaching Assistant & Fellow** *June 2015 to June 2016*

  Autumn 2009 to Spring 2010 – Software System Construction
  Sept 2015 to June 2016 –Computing and Informatics Design I-III