IMPROVING KNOWLEDGE BASE POPULATION WITH INFORMATION EXTRACTION

by

Xiang Li

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May, 2016

Professor Ralph Grishman

Dedication

To my wife and parents.

Acknowledgments

I am indebted to many excellent people who have advised, supported, and inspired me in my journey of PhD study. I will remember all your wisdom, your help, and your friendship, without which I would not be the person I am today.

Firstly and most importantly, I am eternally grateful to my advisor, Prof. Ralph Grishman, for his continuous guidance, patience, and support throughout my graduate study. Ralph gave me consistently invaluable advice, tremendous freedom and numerous support to explore my research interests and seize my career opportunities. I will never forget the moments that he patiently analyzed the data and results with me, guided me the right directions, and helped me with proofreading my draft papers even got it at the last minute until the late night. His professional expertise with a sense of humor, perceptiveness and passion will always be an inspiration to me. I will be forever grateful for his help and feel extremely proud of being referred as "Ralph Grishman's student". There is no doubt that without his advising, support, and encouragement, I could not have accomplished this thesis. Furthermore, I could not have imagined having a more excellent, supportive advisor and mentor.

I am greatly honored to have the wonderful researchers and mentors in my doctoral committee: Prof. Ralph Grishman, Prof. Heng Ji, Prof. Ernest Davis, Prof. Adam

ACKNOWLEDGMENTS

Meyers and Dr. Sungjin Lee, who took the most valuable time in reading my thesis and providing me with extensive comments. I would like to thank all of them for their insightful suggestions for my thesis.

I would like to say many thanks to Prof. Adam Meyers for providing me many linguistic guidance and resources, and Prof. Satoshi Sekine for giving me helpful advice on my research. I am indebted to many present and former Proteus members for providing such an amazing environment for me to learn and grow. I especially cherish the fun and unforgettable days when I study together with Dr. Ang Sun, Dr. Shasha Liao, Dr. Bonan Min, Dr. Yifan He, Dr. Wei Xu, Dr. Maria Pershina, Kai Cao, Thien Huu Nguyen, Lisheng Fu, Miao Fan, and Angus Grieve-Smith. I benefit tremendously from the discussion with them, and I couldn't have asked for a more interesting and engaging group of colleagues.

It was my former advisor Prof. Heng Ji who introduced me, led me, and inspired me to the fascinating world of Natural Language Processing, when I was still a undergraduate student several years ago. I deeply appreciate her tremendous and generous help, encouragement, warmth, patience, and guidance. Without her encouragement, I would not even be brave enough to pursue my dream of being a PhD since childhood. I thank her for making me realize that NLP is such a wonderful field which is worth committing my whole life to explore. I am immensely fortunate that I could get to know Heng in my life, and her advice on both my research and life is always invaluable.

I also feel very lucky to be a member of Blender group during my undergraduate NLP research. I will never forget those bright days when I worked with my colleagues to solve interesting NLP problems. I would like to give special thanks to Dr. Zheng

ACKNOWLEDGMENTS

Chen, Dr. Qi Li, Dr. Hao Li, Dr. Taylor Cassidy, Dr. Haibo Li, Dr. Matt Snover, Dr. Javier Artiles, Dr. Yu Chen, Dr. Suzanne Tamang, Daniel Lin, Adam Lee, Sam Anzaroot, Mingmin Ge, and Jun Zheng for their productive collaborations with me. I learned tremendous knowledge and skills from all of them.

I feel extremely lucky to have had three very fruitful summer internships. I would like to thank my awesome internship mentors Dr. Ang Sun and Dr. Hakan Kardes from Inome Research, Dr. Gökhan Tür and Dr. Dilek Z. Hakkani-Tür from Microsoft Research, and Dr. Sungjin Lee, Dr. Aasish Pappu and Dr. Amanda Stent from Yahoo! Labs. A large part of this dissertation is the result of collaboration with them. I also would like to thank other researchers at Inome Research, especially Dr. Lin Chen, Sriram Krishnan, Dr. Xin Wang, and Deepak Konidena. I would like to thank following researchers in MSR: Dr. Larry Heck, Dr. Malcolm Slaney, Dr. Andreas Stolcke, and Dr. Geoffrey Zweig. I would also like to thank everyone in the NLP team of Yahoo! Labs and others: Dr. Joel Tetreault, Dr. Kapil Thadani, Dr. Yashar Mehdad, Dr. Liangliang Cao, Dr. Meizhu Liu, Dr. Dragomir Radev, Dr. Steven Skiena, and Seth Tropper. I also want to thank my co-interns Dr. Sujatha Das Gollapalli at Inome Research, Dr. Qi Li, Dr. Yun-Nung (Vivian) Chen, and TJ Tsai at MSR, Dr. William Wang Yang, Junyi (Jessy) Li, and Ellie Pavlick at Yahoo! Labs. I thank all of them for their wonderful mentoring, help, and sharing the fascinating moments with me throughout my internships.

Last, but certainly not least, I would like thank my parents and my parents-in-law for their selfless support of my PhD study and unfailing faith in me. I am most grateful for the support from my beloved wife and best friend, Xiaochao Jin, for her infinite love, trust, understanding, encouragement, patience and care, which keep me moving forward

and become a better person. She has always believed in me and I couldn't have made this happen without her. Finally I have reached the finish line of this PhD journey because of her constant source and reflection of love.

To conclude, I could not have succeeded without the advice, support and influence of my advisors, colleagues, friends and family. I cannot express my gratitude enough for their contributions to my progress in both study and life.

Abstract

Knowledge Bases (KBs) are data resources that encode world knowledge in machine-readable formats. Knowledge Base Population (KBP) aims at understanding this knowledge and extending KBs with more semantic information, which is a fundamental problem in Artificial Intelligence. It can benefit a wide range of tasks, such as semantic search and question answering. Information Extraction (IE), the task of discovering important types of facts (entities, relations and events) in unstructured text, is necessary and crucial for successfully populating knowledge bases. This dissertation focuses on four essential aspects of knowledge base population by leveraging IE techniques: extracting facts from unstructured data, validating the extracted information, accelerating and enhancing systems with less annotation effort, and utilizing knowledge bases to improve real-world applications.

First, we investigate the Slot Filling task, which is a key component for knowledge base population. Slot filling aims to collect information from a large collection of news, web, or other sources of documents to determine a set of predefined attributes ("slots") for given person and organization entities. We introduce a statistical language understanding approach to automatically construct personal (user-centric) knowledge bases from conversational dialogs.

Second, we consider how to probabilistically estimate the correctness of the extracted slot values. Despite the significant progress of KBP research and systems in recent years, slot filling approaches are still far from completely reliable. Using the NIST KBP Slot Filling task as a case study, we propose a confidence estimation model based on the Maximum Entropy framework, and demonstrate the effectiveness of this model in both precision and the capability to improve the slot filling aggregation through a weighted voting strategy.

Third, we study rich annotation guided learning to fill the gap between an expert annotator and a feature engineer. We develop an algorithm to enrich features with the guidance of all levels of rich annotations from human annotators. We also evaluate the comparative efficacy, generality and scalability of this framework by conducting a case study on Knowledge Base Population domain, facilitating slot filling systems. Empirical studies demonstrate that with little additional annotation time, we can significantly improve the performance.

Finally, we explore utilizing knowledge bases in a real-world application – personalized content recommendation. Traditional systems infer user interests from surface-level features derived from online activity logs and user demographic profiles, rather than deeply understanding the context semantics. We conduct a systematic study to show the effectiveness of incorporating deep semantic knowledge encoded in the entities on modeling user interests, by utilizing the abundance of entity information from knowledge bases.

Table of contents

TABLE OF CONTENTS

List of Figures

List of Figures

List of Tables

List of Tables

Chapter 1

Introduction

"I am convinced that the crux of the problem of learning is recognizing relationships and being able to use them."

- Christopher Strachey, *a letter to Alan Turing, 1954*

1.1 Knowledge Base

A Knowledge Base (KB) is a special-purpose structured resource used for the collection and management of knowledge in the form of logical statements. KBs store factual information in form of relationships between entities, and most of them are in a graph structure. This kind of relational knowledge representation has a long history in logic and artificial intelligence (Davis et al., 1993), for example, in semantic networks (Sowa, 2008) and frames (Minsky, 1974).

Most of the KBs follow the Resource Description Framework (RDF) standard or similar format to represent facts in the form of binary relationships, in particular *(entity,*

predicate, value) triples, where *entity* represent person, organization, and other types of entities (e.g., *Leonardo DiCaprio*), *predicate* indicates the relationship, and *value* can be another entity (e.g., *Jack Dawson*), a type (e.g., *Actor*), an attribute (e.g., *41*), and other factual information. The existence of a particular triple indicates an existing fact. For example, the facts extracted from the following context can be expressed via the triples shown in Table 1.1:

> *Leonardo DiCaprio, 41, was an actor who starred in James Cameron's romantic disaster movie Titanic (1997) as Jack Dawson.*

entity	**predicate**	**value**
(*LeonardoDiCaprio*,	age,	*41*)
(*LeonardoDiCaprio*,	profession,	*Actor*)
(*LeonardoDiCaprio*,	starredIn,	*Titanic*)
(*LeonardoDiCaprio*,	played,	*JackDawson*)
(*JamesCameron*,	directed,	*Titanic*)
(*Titanic*,	genre,	*RomanticDisaster*)
(*Titanic*,	releaseYear,	*1997*)
(*JackDawson*,	characterIn,	*Titanic*)

Table 1.1: Example Triples in Knowledge Base

Triple stores covering various domains have already emerged, such as freebase.org. We can aggregate all the triples in a KB to form a graph, where nodes represent entities and values, and directed edges represent relationships. The direction of an edge can

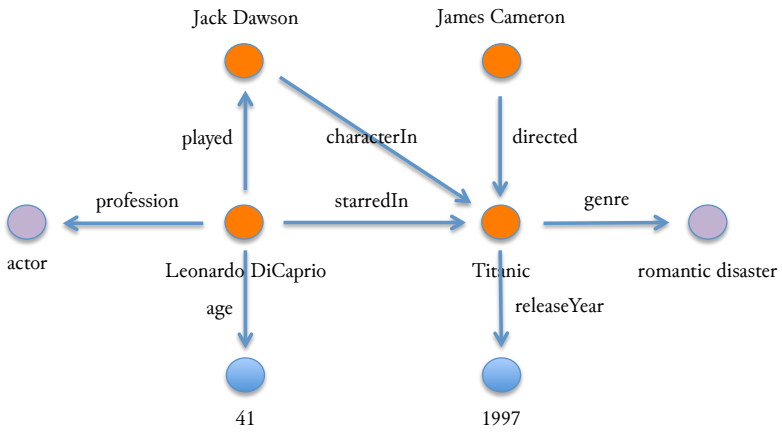

Figure 1.1: Sample Knowledge Graph, where nodes represent entities, types, or attributes, and edges represent types of relations.

reflect which entities are the subject entities in that triple if there are two entities, i.e., an edge points from the subject entity to the object entity. Different relations are represented using different types of edges. This construction is called a Knowledge Graph (KG). As a KG is useful for understanding and visualizing the structure of knowledge bases, many KBs are also represented as a KG. Figure 1.1 shows an example knowledge graph based on the facts in Table 1.1. Figure 1.2 illustrates part of the structure of the DBpedia KG, where knowledge from various domains gets stored and linked [1].

Recent years have seen tremendous research and engineering efforts in constructing large knowledge bases (KBs). Examples of these knowledge bases constructed by research communities include DBpedia (Auer et al., 2007), DeepDive (Niu et al., 2012a), Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010), OpenIE (Banko et al., 2007; Etzioni et al., 2011), ProBase (Wu et al., 2012), and YAGO (Biega et al., 2013; Hoffart et al., 2013; Mahdisoltani et al., 2015). At the same time, many technology

1. This was produced by the Linked Open Data Cloud project (http://lod-cloud.net).

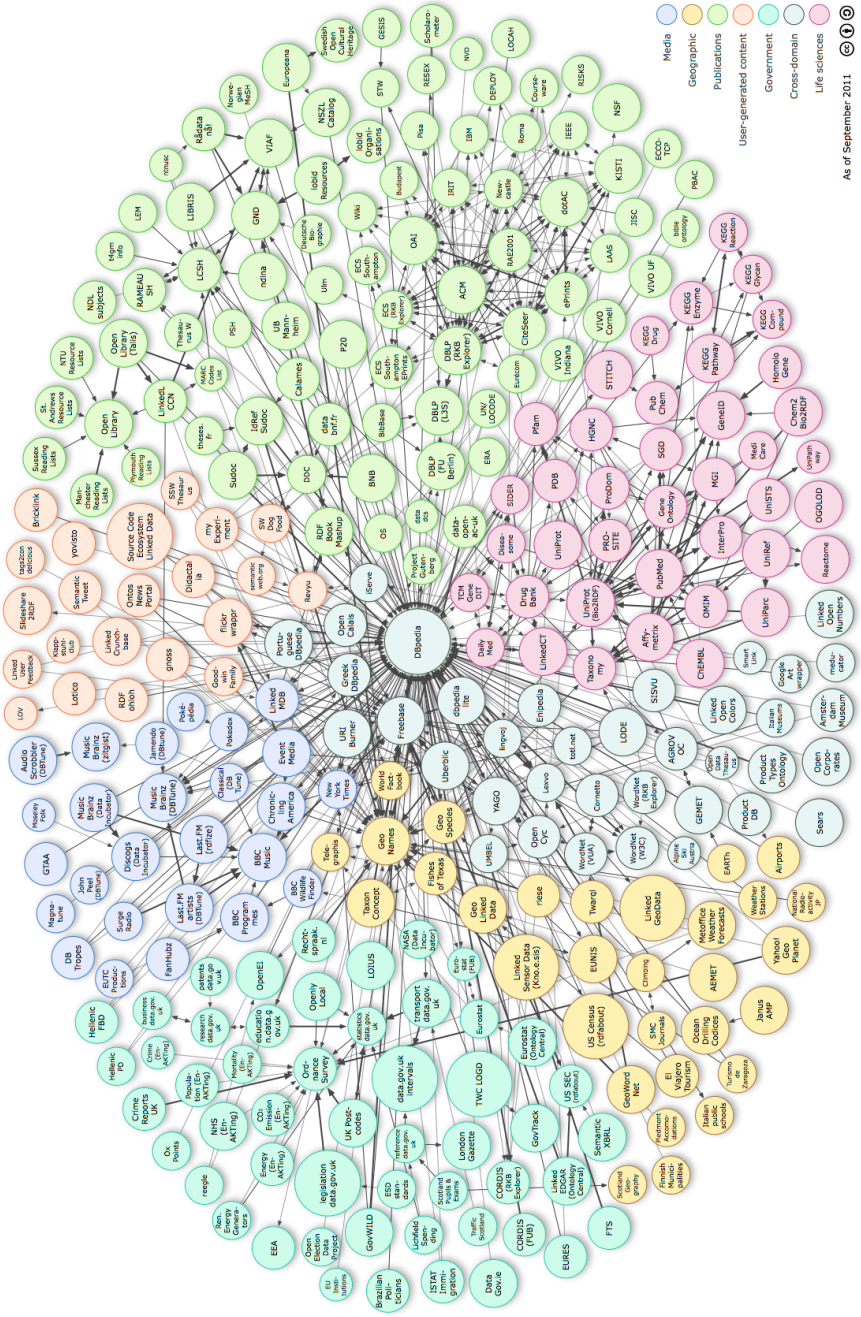

Figure 1.2: Visualization of DBpedia Knowledge Base Structure (Schmachtenberg et al., 2014)

Knowledge Base	Entities	Relation Types	Facts
Wikidata	18 M	1,632	66 M
YAGO2	9.8 M	114	447 M
DBpedia	4.6 M	1,367	539 M
Freebase	40 M	35,000	637 M
Yahoo! Knowledge Graph	3.4 M	800	1,391 M
Google Knowledge Graph	570 M	35,000	18,000 M

Table 1.2: Size of Some Schema-based Knowledge Bases

companies also build and maintain their own knowledge bases to advance applications, such as Google Knowledge Graph (Singhal, 2012), Microsoft Knowledge Graph Satori, Yahoo Knowledge Graph, and Facebook Entity Graph. These knowledge bases can be utilized for various purposes, such as search and question answering. Table 1.2 shows a selection of such KBs and their number of entities, relation types, and facts.

Up to this point, we may bring up a question like "*How does it differ from traditional information management?*" Actually traditional DBs or data warehouses are centered around "records" and "tables". This is certainly efficient when a domain is well known in advance and discovery of new information is not expected. However, in domains where one has the need to flexibly connect all sorts of information, some of it unexpected, the knowledge base technology has distinctive advantages:

- **Entity centric** All bits of information are cataloged with respect to an entity or entities it is relevant for.

- **Schemaless** Free knowledge structure is allowed, no schema is required a priori.
- **Metadata Rich, Self Describing** Streams of metadata rich knowledge are now easier to integrate and scale across organizations and domains.

Uses of Knowledge Bases

Knowledge Bases provide semantically structured information that is interpretable by computers — a property that is regarded as an important ingredient to build more intelligent machines (Lenat and Feigenbaum, 1991). Consequently, knowledge bases are already powering multiple "Big Data" applications in a variety of commercial and scientific domains. A good example is the integration of Google's Knowledge Graph, which currently stores 18 billion facts about 570 million entities, into the results of Google's search engine (Singhal, 2012). The Google Knowledge Graph is used to identify and disambiguate entities in text, to enrich search results with semantically structured summaries, and to provide links to related entities in exploratory search (Nickel et al., 2016). Similarly, Microsoft integrated Satori with its Bing search engine, and Yahoo utilized Yahoo Knowledge Graph in its Yahoo search engine.

In an enterprise the typical goal of a knowledge base is to collect information about every entity of interest in a domain (and their relationships) and make it "*maximally easy*" to reuse for any application, current, future, foreseen or unforeseen. Typical usages include searching and displaying information about entities, recognizing entities in context, connecting entities to content and data sources, discovering and recommending related information, question answering, semantic parsing, connecting people, places and things in social networks, virtual personal assistants, and so on.

CHAPTER 1. INTRODUCTION

Enhancing search results with semantic information from knowledge graphs can be seen as an important step to transform text-based search engines into semantically aware question answering services. Another prominent example demonstrating the value of knowledge graphs is IBM's question answering system *Watson*, which was able to beat human experts in the game of *Jeopardy!*. Among others, this system used YAGO, DBpedia, and Freebase as its sources of information (Ferrucci et al., 2010). Repositories of structured knowledge are also an indispensable component of digital assistants, such as Apple Siri, Microsoft Cortana, Google Now, Amazon Echo, and Facebook M.

There are knowledge bases that store general information, such as *Freebase*, which is a large collaborative knowledge base consisting of data composed mainly by its community members. It is an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions. At the same time, knowledge bases are also utilized in various specialized domains for different usages. For instance, the Internet Movie Database (abbreviated IMDb) is an online knowledge base of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews. Furthermore, Bio2RDF (Belleau et al., 2008), Neurocommons (Ruttenberg et al., 2009), and LinkedLifeData (Momtchev et al., 2009) are knowledge bases that integrate multiple sources of biomedical information. These have been used for question answering and decision support in the life sciences.

1.2 Knowledge Base Population

Even the largest knowledge bases are far from complete, since new knowledge is always emerging rapidly. For instance, in those KBs, entities which are popular usually contain more knowledge facts, e.g., the basketball player *Michael Jordan* and the actor *Leonardo DiCaprio*, while most other entities often have fewer facts. In addition, facts should be updated as entities develop, such as changes in the cabinet, a marriage event, or an acquisition between two companies. Most of the missing knowledge is available on web pages in the form of free text now. To access that knowledge, information extraction and information integration methods are necessary. Recent advances in natural language processing and information extraction have made it possible to construct structured KBs from online encyclopedia resources, at an unprecedented scale and much more efficiently than traditional manual editing.

Information Extraction (IE) is the process of extracting structured information from unstructured or semi-structured machine readable documents. Traditional IE systems can extract information from individual documents in isolation quite efficiently. To meet the real life requirement of building large-scale knowledge bases, the current IE systems utilize Information Retrieval techniques to collect information (scattered among multiple document collection), identify relevant documents, and integrate facts involving redundant, complementary or conflicting entities.

To be specific, IE systems begin with gathering all known information about a given query entity. For instance, given the query "*Barack Obama*", the goal of slot filling systems is to collect *Barack Obama*'s birthplace, birthdate, occupation, spouse, and other predefined attributes (or slots). This can be thought of as "*filling*". A key aspect of this is

relation extraction – the classification of a sentence and two entities in the sentence to a relation of interest. For example, reading "*Barack Obama was born in Hawaii*" and extracting the relation born_in*(Barack Obama, Hawaii)*. The slot filling systems are required to automatically distill information from the document collection which fills missing KB attributes for focus entities. The slot filling task is a hybrid of traditional IE (a fixed set of relations) and QA (responding to a query, generating a unified response from a large collection). Then IE systems link entities and information about these entities to the entries in the data/knowledge base.

Approaches

Completeness, accuracy, and data quality are important parameters that determine the usefulness of knowledge bases and are influenced by the way knowledge bases are constructed. We can classify KB construction methods into four main groups:

- In *curated* approaches, triples are created manually by a closed group of experts.
- In *collaborative* approaches, triples are created manually by an open group of volunteers.
- In *automated semi-structured* approaches, triples are extracted automatically from semi-structured text (e.g., infoboxes in Wikipedia) via hand-crafted rules, learned rules, or regular expressions.
- In *automated unstructured* approaches, triples are extracted automatically from unstructured text via machine learning and natural language processing techniques.

Table 1.3 lists current knowledge base population projects classified by their creation method and data schema. In this dissertation, we will only focus on schema-based KBs. Construction of curated knowledge bases typically leads to highly accurate results, but this technique does not scale well due to its dependence on human experts. Collaborative knowledge base construction, which was used to build Wikipedia and Freebase, scales better but still has some limitations. For instance, the place of birth attribute is missing for 71% of all people included in Freebase, even though this is a mandatory property of the schema (West et al., 2014). Also a recent study (Suh et al., 2009) found that the growth of Wikipedia has been slowing down. Consequently, automatic knowledge base population methods have been gaining more attention. Nickel et al. (2016) provided a review of state-of-the-art statistical relational learning (SRL) methods applied to very large knowledge graphs, and also demonstrated how SRL can be used in conjunction with machine reading and information extraction methods to automatically build knowledge repositories.

Knowledge intensive enterprises across many sectors can immensely benefit from the ability to keep the data and the structure of any piece of information they can collect about any entity of interest to their business. Typically an enterprise knowledge base is created with data ranging from relational databases to public open data including public knowledge graphs, to unstructured data processed via machine learning API, to a customer's own datasets.

For instance, Yahoo acquires and extracts information about entities from multiple complementary sources using information extraction techniques, and leverage open data sources such as Wikipedia as well as closed data sources from paid providers. The mined

Method	Schema	Examples
curated	yes	Cyc/OpenCyc (Lenat, 1995) WordNet (Miller, 1995) UMLS (Bodenreider, 2004)
collaborative	yes	Wikidata (Vrandecic and Krötzsch, 2014) Freebase (Bollacker et al., 2008)
auto. semi-structured	yes	YAGO (Hoffart et al., 2013; Suchanek et al., 2007) DBpedia (Auer et al., 2007) Freebase (Bollacker et al., 2008)
auto. unstructured	yes	Knowledge Vault (Dong et al., 2014) NELL (Carlson et al., 2010) PATTY (Nakashole et al., 2012) PROSPERA (Nakashole et al., 2011) DeepDive/Elementary (Niu et al., 2012b)
auto. unstructured	no	ReVerb (Fader et al., 2011) OLLIE (Mausam et al., 2012) PRISMATIC (Fan et al., 2010)

Table 1.3: Knowledge Base Population Projects (Nickel et al., 2016)

facts are stored uniformly in a central knowledge repository where entities and their attributes and relationships are categorized, normalized, and validated against a common ontology using a generalized and scalable framework. Then machine learning techniques are applied to disambiguate and blend together entities that co-refer to the same real-world objects, eventually turning siloed, incomplete, inconsistent, and possibly inaccurate informations into a rich, unified, disambiguated knowledge graph. A plugin system can be used to enrich the graph with inferred information useful for the Yahoo applications. Meanwhile, editorial curation can also be leveraged for hot fixes. The current Yahoo Knowledge Graph manages millions of interconnected entities and relationships, and runs on top of distributed storage and data processing systems. (Blanco et al., 2013)

The Knowledge Base Population (KBP) track, organized by U.S. National Institute of Standards and Technology (NIST)'s Text Analysis Conference (TAC), is an active but challenging research task, aiming to promote research in discovering information about entities and augmenting a Knowledge Base (KB) with this information (Ji et al., 2010). TAC KBP mainly consists of four typical knowledge base population tasks, and more information can be found in the task definition (e.g., Ji et al., 2010; Ji and Grishman, 2011; Ji et al., 2011, 2014, 2015; Surdeanu and Ji, 2014; "Slot Filler Validation/Ensembling at TAC 2015 Task Guidelines" 2015; "TAC KBP 2015 Slot Descriptions" 2015):

- **Entity Linking (EL)** aims to link names in a provided document to entities in the KB or NIL.
- **Slot Filling (SF)** aims to extract information about an entity in the KB to automatically populate a new or existing KB.

- **Cold Start KBP (CSKBP)** aims to build a knowledge base from scratch using a given document collection and a predefined schema for the entities and relations that compose the KB.

- **Slot Filler Validation (SFV)** aims to refine output from SF systems by either combining information from multiple slot filling systems, or apply more intensive linguistic processing to validate individual candidate slot fillers.

1.3 Dissertation Overview

The rest of the dissertation is organized as follows: Chapter §2 reviews background knowledge and summarizes related works. The chapter also discusses current challenges of the task and describes several knowledge resources that may benefit understanding problems. Then we present four concrete studies to answer the following four questions in order to achieve the above goal:

- **Question 1: "*How can we build knowledge bases?*"**

  Knowledge graphs provide a powerful representation of entities and the relationships between them, but automatically constructing such graphs from spoken language utterances presents novelty and numerous challenges. We introduce a statistical language understanding approach to automatically construct personal (user-centric) knowledge graphs in conversational dialogs. Such information has the potential to better understand the users' requests, fulfilling them, and enabling other technologies such as developing better inferences or proactive interactions. Three key language understanding components are built: (1) *Personal Assertion Clas-*

sification identifies the user utterances that are relevant with personal facts, e.g., *"my mother's name is Rosa"*; (2) *Relation Detection* classifies the personal assertion utterance into one of the predefined relation classes, e.g., *"parents"*; and (3) *Slot Filling* labels the attributes or arguments of relations, e.g., *"name(parents):Rosa"*. Our experiments using the Microsoft conversational understanding system demonstrate the performance of this proposed approach on the population of personal knowledge graphs. More will be described in Chapter §3.

- **Question 2: "*How can we validate the correctness of information in knowledge bases?*"**
  As we know knowledge base population systems automatically extract structured information from machine-readable documents, such as newswire, web, and multimedia. Despite significant improvement, the performance is far from perfect. Hence, it is useful to accurately estimate confidence in the correctness of the extracted information. Using the Knowledge Base Population Slot Filling task as a case study, we propose a confidence estimation model based on the Maximum Entropy framework, and the effectiveness of this model is demonstrated in both precision and the capability to improve slot filling task through a weighted voting strategy. More details will be discussed in Chapter §4.

- **Question 3: "*How can we build knowledge bases more efficiently?*"**
  As an inter-disciplinary area, statistical natural language processing (NLP) requires two crucial aspects: (1) good choice of machine learning algorithms; (2) good feature engineering. In particular, (2) significantly affects the performance of systems. Linguistic annotation is a fundamental and crucial step of supervised learning. However, feature engineering remains a challenging task. Moreover, annotated

corpora are usually prepared by a separate group of human annotators before system development, such as LDC annotated corpora. As a result, almost all previous NLP systems only utilized direct manual labels for training, while ignoring the valuable knowledge that human annotators have learned and summarized from corpora preparation. In fact, compared to system developers who normally design features based on partial data analysis, human annotators are usually more knowledgeable because they need to go through the entire data set and restrictively follow annotation guidelines. We have applied rich annotation guided learning to help improve the performance of knowledge base population systems and related tasks, and Chapter §5 will present more about this framework.

- **Question 4: "*How can we use these better knowledge bases to advance other tasks?*"** Nowadays the web plays an important role in the distribution of information from different sources to the users. The main problem that comes into play is called *Information Overload*, which necessitates the content recommendation techniques to help choose the best items matching users' interests. Thus modeling user interests is a key and challenging component for personalized content recommendation. Traditional systems usually infer users' interests from surface-level features derived from online activity logs and user demographic profile, rather than deeply understanding the semantics behind the users' requests. Named entities that appear in the search queries, contents, stream news articles, and other content forms enable interpreting what these contents are really about. We have conducted a systematic study to show the effectiveness of incorporating deep semantic knowledge encoded in the entities for modeling users' interests, by utilizing the abundance of knowl-

edge populated in knowledge bases, which will be discussed more in Chapter §6.

Finally, Chapter §7 concludes the main contributions and discusses a number of interesting directions that can be explored in the future.

Chapter 2

Prior Work

In this chapter, we review prior work that is closely related to the solutions (Chapter §3, §4, §5, and §6) described in this dissertation. Our goal is to introduce the background for our work, but not to present a comprehensive survey of research in knowledge base population. Therefore, certain interesting research such as *Entity Linking* is not described here.

2.1 Knowledge Graph Population in SLU

Conventional Spoken Language Understanding (SLU) approaches typically focus on user intent determination and slot filling tasks. Intent determination systems have roots in call routing systems used in call centers (e.g., *Billing* vs. *Sales*), such as the AT&T *How May I Help You system* (Gorin et al., 1997). They are usually modeled as an utterance classification task aiming at classifying a given speech utterance S_i into one of M semantic classes, $\hat{C}_r \in \mathcal{C} = \{C_1, ..., C_M\}$ (where r is the utterance index). To this end, researchers

have tried various classification methods such as Boosting (; Schapire and Singer, 2000; Zitouni et al., 2003), support vector machines (SVMs) (Haffner et al., 2003), and more recently deep learning (Dauphin et al., 2014; Sarikaya et al., 2011).

On the other hand, slot filling systems have flourished after DARPA sponsored the Airline Travel Information System (ATIS) (Price, 1990) project. These systems attempted to convert the user utterance into an SQL query. The approaches ranged from generative models such as hidden Markov models (He and Young, 2003; Pieraccini et al., 1992), discriminative classification methods (Kuhn and Mori, 1995; Tür et al., 2010; Wang and Acero, 2006), knowledge-based methods, probabilistic context free grammars (Seneff, 1992; Ward and S.Issar, 1994), and more recently deep learning methods (Deng et al., 2012; Xu and Sarikaya, 2013; Yao et al., 2014). Recently, the state of the art approach for slot filling is framing the task as a sequence classification problem, similar to part of speech tagging or named entity extraction, in order to find both the boundaries and labels of phrases which are used to fill the semantic template. The non-slot filler words are assigned to a special null state.

Similar to the slot filling task defined in SLU, another Slot Filling task is constructed in the Knowledge Base Population (KBP) track, organized by U.S. NIST's Text Analysis Conference (TAC) (Ji and Grishman, 2011). The KBP Slot Filling (SF) task aims at collecting from a large-scale multi-source corpus the values ("slot fillers") for certain attributes ("slot types") of a query entity, which is a person or some type of organization. KBP2013 has defined 25 slot types for persons (per) (e.g., age, spouse, employing organization) and 16 slot types for organizations (org) (e.g., founder, headquarters-location, and subsidiaries). Some slot types take only a single slot filler (e.g., *per:birth_place*),

whereas others take multiple slot fillers (e.g., *org:top_employees*). More information can be found in the task definition (Ji et al., 2010). Various approaches have been proposed to perform the task, including pattern matching (Chen et al., 2010b; Min et al., 2012), question answering (Byrne and Dunnion, 2010; Chen et al., 2010b), hand-coded heuristic rules (Gao et al., 2010; Yu et al., 2010), distant supervision (Chrupala et al., 2010; Intxaurrondo et al., 2010; Min et al., 2012; Nemeskey et al., 2010; Surdeanu et al., 2010), hybrid (Chen et al., 2010b; Min et al., 2012), knowledge graph based (Yu et al., 2014a), etc.

As we know, knowledge graphs have been demonstrated to be useful and powerful in many conversational understanding research tasks. Hakkani-Tür et al. (2013) and Hakkani-Tür et al. (2014) compute entity type weights to enrich semantic knowledge graph entities with probabilistic weights for the SLU relation detection task. El-Kahky et al. (2014) proposes a technique to enable SLU systems to handle user queries beyond their original semantic schemas defined by intents and slots. Wang et al. (2014) presents a full pipeline to leverage semantic web search and browse sessions for a semantic parsing problem in multi-turn spoken dialog systems. Tür et al. (2012) and Heck et al. (2013) present studies towards bringing together the semantic web experience and unsupervised statistical natural language semantic parsing modeling. Heck and Hakkani-Tür (2012) proposes an unsupervised training approach for SLU systems on the intent detection task, which exploits the structure of semantic knowledge graphs from the web.

2.2 Confidence Estimation for Information Extraction

Confidence estimation is a generic machine learning approach for measuring confidence of a given output, and many different CE methods have been used extensively in various Natural Language Processing (NLP) fields (Gandrabur et al., 2006). Gandrabur and Foster (2003) and Bach et al. (2011) investigated the use of machine learning approaches for confidence estimation in machine translation. Agichtein (2006) used Expectation-Maximization algorithms to estimate the confidence for partially supervised relation extraction. White et al. (2007) described how a maximum entropy model can be used to generate confidence scores for a speech recognition engine. Louis and Nenkova (2009) presented a study of predicting the confidence of automatic summarization outputs. Many approaches for confidence estimation have also been explored and implemented in other NLP research areas.

There are also many previous confidence estimation studies in IE, and most of these have been in the Active Learning literature. Thompson et al. (1999) proposed a rule-based extraction method to compute confidence. Scheffer et al. (2001) utilized hidden Markov models to measure the confidence in an IE system, but they only estimated the confidence of singleton tokens. Culotta and McCallum (2004)'s work is the most relevant to our work, since they also utilized a machine learning model to estimate the confidence values for IE outputs. They estimated the confidence of both extracted fields and entire multi-field records mainly through a linear-chain Conditional Random Field (CRF) model, but their case studies on contact information extraction from web pages are not as complicated and challenging as slot filling, since SF systems need to handle difficult cross-document coreference resolution, sophisticated inference, and also other

challenges (Min and Grishman, 2012). For the TAC KBP Slot Filling task, recent research approaches of filtering incorrect values from multiple systems include heuristic rules, weighted voting (Li and Grishman, 2013), supervised learning to rank algorithms (Tamang and Ji, 2011), unsupervised multi-dimensional truth finding (Yu et al., 2014a), and more.

2.3 Rich Annotations

In some NLP tasks such as information retrieval, it has proven effective to incorporate user feedback to customize or tune a system, such as personalized search (e.g., Lv et al., 2006; Tyler and Teevan, 2010). However, such user feedback is not always available. Nevertheless most supervised learning methods rely on the labels by human annotators. Therefore there is great potential to fully utilize the deep knowledge from human annotators. Vapnik (2009) proposed to incorporate more of "*teacher's role*" (i.e., privileged knowledge) into traditional machine learning paradigm. We follow this basic idea and incorporate additional feedback from annotators into system development.

Recent work has pointed out the problem that human annotators are "underutilized" and incorporated rich annotations into many classification problems (Yu et al., 2011; Zaidan et al., 2007; Zaidan and Eisner, 2008). Some other work (Druck et al., 2008; Haghighi and Klein, 2006; Raghavan et al., 2006) asked human annotators to label or select features. In this dissertation we shall generalize all kinds of annotator rationales into multiple levels and conduct a systematic study.

Castro et al. (2008) investigated a series of human active learning experiments. Our experiment of using Rich Annotation Guided Learning to speed up human assessment

exploited assistance from multiple systems. Our idea of learning from error corrections is also similar to Transformation-based Error-Driven Learning, which has been successfully applied in many NLP tasks such as part-of-speech tagging (Bril, 1995), chunking (Milidiu et al., 2008), word sense disambiguation (Dini et al., 1998) and semantic role labeling (Williams et al., 2004). In these applications the transformation rules are automatically learned based on sentence contexts at each iteration. However, our applications require global knowledge which may be derived from diverse linguistic levels and vary from one system to the other, and thus it's not straightforward to design and encode transformation templates. Therefore in this dissertation we choose a more modest way of exploiting the comments encoded by human annotators.

There are many other alternative automatic assessment approaches for slot filling. Besides the RTE-KBP validation (Bentivogli et al., 2011) discussed in the dissertation, some slot filling systems also conducted filtering and cross-slot reasoning (e.g., Castelli et al., 2010; Chen et al., 2010a) to improve results.

2.4 User Profiling for Content Recommendation

Our user profiling for content recommendation work consists of three areas of research, including user profiling, recommendation systems, and factorization machines. Some of the related work and publications are listed below.

User profiling aims to represent users' interests in the same feature space as that of the items being recommended (Chen and Pu, 2004). One of the popular tasks that needs user profiling is content recommendation (Middleton et al., 2004; Webb et al., 2001; Zukerman and Albrecht, 2001), and user profiling can also be applied on personal-

ized web search and ads push to enhance the user experience (Sieg et al., 2007; Sugiyama et al., 2004). Recommendation system is an information filtering system that attempts to present information items that are likely of interest to the user. In general there are two approaches to recommendation, collaborative filtering and content-based recommendation. Collaborative filtering is the process of filtering information using the techniques involving collaboration among users, such as nearest neighborhood models (Sarwar et al., 2001) and matrix factorization methods (Koren et al., 2009). Content-based recommendation mainly explores explicit features of items and users (Adomavicius and Tuzhilin, 2005), and this approach is essential for the applications where a lot of *cold start* items appear, which is typical in content recommendation. Various approaches have been proposed to be effective for new recommendation, including spatio-temporal model (Agarwal et al., 2009), probabilistic models (Liu et al., 2010), click shaping (Agarwal et al., 2012), hyper graph learning (Li and Li, 2013), activity ranking (Agarwal et al., 2014), and latent factor models (Zhong et al., 2015).

The integration of hierarchical knowledge repository in recommendation and user preference profiling is becoming an emerging area of interest. For instance, Yu et al. (2014b) studied personalized entity recommendation for search engine users by utilizing user click log and the knowledge extracted from Freebase. Cheekula et al. (2015) proposed a content-based recommendation approaches that adapts a spreading activation algorithm over the DBpedia category structure to identify entities of interest to the user. Passant (2010) studied to recommend music entities based on its Linked Data Semantic Distance (LDSD) from other explicitly rated entities of the user with DBpedia. Di Noia et al. (2012) have harnessed DBpedia in order to recommend movies based on the content

of the user. And Kapanipathi et al. (2014) leveraged hierarchical relationships present in knowledge-bases to infer user interests expressed as a hierarchical interest graph.

However, user interests are traditionally modeled using different sources of profile information (e.g., explicit demographic or interest profiles, or implicit profiles based on previous queries, search result clicks, general browsing activity, or even richer desktop indices). And user preference is usually inferred from their activities (e.g., clicking on a hyperlink, viewing/saving/bookmarking a page), rather than trying to understand the semantics of the queries and the content of visited pages. But the use of deep semantic knowledge allows us to furnish rich contextual information. For example, Yahoo Knowledge Graph entities extracted either from the search queries or the contents of the webpages the user has visited make it possible to get connected with knowledge bases where plenty of deep semantic knowledge about the entities exists. Hence, in this work, we model users' interests from a deeper semantic aspect by investigating the information network based on the entities that the users are really interested in.

Our work is closely related with the work of Zhong et al., 2015. The general goal of our proposed framework is consistent with the one of Zhong et al., 2015, which is to model user's interests and provide personalized content recommendation within a large-scale framework. We have also utilized a similar framework, Factorization Machines (Rendle, 2010), to overcome both data sparsity and cold-start problems and infer user interest vectors. One of the most significant differences between their work and our work is that they only utilized the entities that had appeared in each user's previous click history and content categories as features, but our idea is to exploit more semantic knowledge and entities to enrich the feature space, by utilizing the related entities extracted from

knowledge base and augment seen entities with similar unseen entities. Thus our work can take the advantages of the abundant knowledge stored in the Yahoo Knowledge Graph to capture users' preferences more from the semantic side. This also enables the user profiling model to predict users' future interests from a long-term concern. Hence, our work shares the similar goals and framework with Zhong et al., 2015, but our methodology and focus are substantially different.

Chapter 3

Personal Knowledge Graph Population ¶

3.1 Introduction

With the rapid proliferation of smart phones aligned with advances in automatic speech recognition (ASR) and machine learning technologies, virtual personal assistant (VPA) systems, such as Apple Siri and Microsoft Cortana, have started to emerge. These systems are typically more complex than applications like voice search or voice messaging, and require advanced spoken language understanding (SLU) capabilities, which are robust to variability in natural language, ASR noise, and spontaneous ungrammatical spoken input.

In VPA systems, at each turn, a user's speech is recognized, and then the SLU com-

. ¶ This work has been published in "*Personal Knowledge Graph Population from User Utterances in Conversational Understanding*". Xiang Li, Gokhan Tur, Dilek Hakkani-Tur, and Qi Li. *Proceedings of 2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014.

ponent semantically parses that into a task-specific semantic representation of the user's intention (e.g., *play music* or *check weather*) with associated arguments (e.g., *name of the artist* or *location*) (Tür and DeMori, 2011). Since SLU is not a single stand-alone technology like speech recognition or synthesis, there is no established definition of a semantic parse; it depends on the task, domain, or application. The dialog manager then interprets and decides on the most appropriate system action exploiting semantic context, user specific meta-information, such as geo-location and personal preferences, and other contextual information. For example, if the user clicks on a map on the screen and says "*How much is the cheapest gas around here?*", the system should be able to interpret the domain, intent, and the associated arguments (Tür et al., 2014), like:

> *Domain*: Local Business; *Intent*: Get_Price
> *Slots*: *good*: gas; *cost_relative*: cheapest; *location*: (lat,long)

Typically, spoken dialog queries to a dialog system may be classified as *informational*, *transactional*, and *navigational* in a similar way to the taxonomy for web search (Broder, 2002). Informational queries seek an answer to a question, such as "*find the movies of a certain genre and director*", transactional queries aim to perform an operation, such as "*play a movie*", or "*reserve a table at a restaurant*", and navigational queries aim to navigate in the dialog, such as "*go back to the previous results*". However, in VPA systems, in addition to these three main categories, more and more *personal assertion* utterances are conveyed from the users, where users are talking about themselves (e.g., "*I am vegetarian*" or "*My daughter is getting married*". In such utterances, instead of instructing the VPA to perform some unambiguous specific user intent, users interact with the VPA in a more intimate

way. This is an uncharted area of research in the SLU literature, since the users express no overt intention.

More formally, an assertion is a declarative sentence (instead of imperative, interrogative, or any other type). The personal assertion sentences are more focused on describing personal facts, where the subject of the sentence is either the user (i.e., "*i*") or somebody/something related to the user (i.e., "*my wife*", "*my birthday*", etc.). While such personal information may vary greatly, as a first step towards processing such personal assertions, we exploit the semantic knowledge graphs of the semantic web (McIlraith et al., 2001; Shadbolt et al., 2006) and semantic search (Guha et al., 2003). A commonly used ontology is provided in schema.org, with consensus from academia and major search companies like Microsoft, Google, and Yahoo. In this ontology, the personal relation types, such as education or family are also defined for individuals.

In this study, more specifically, we follow the Freebase semantic knowledge graph schema[1], including 18 types of relations about the *people.person* entity, such as nationality (the country (or countries) that the person is a citizen of), profession (the name of the person's primary occupation(s), during their working life), parents (the biological parents and adoptive parents), and so on. A list of the personal factual relations that are encountered in the spoken utterance evaluation dataset is shown in Section 3.4. For illustration, example utterances with defined semantic space are shown in Table 3.1, and a sample user-centered knowledge graph based on these utterances is shown in Figure 3.1.

For each relation, we leverage the complete set of entities in the Freebase knowledge graph that are connected to each other with the specific relation, and search for these entity pairs on the web using the Microsoft Bing search engine (www.bing.com). We use

1. http://www.freebase.com/schema

Number	Utterance	Relation	Slot
1	*my mother's name is Rosa*	parents	parents : *Rosa*
2	*my wife her name is Amy*	spouse_s	spouse_s : *Amy*
3	*my children are Alex and Eileen*	children	children : *Alex* ; children : *Eileen*
4	*I was born on November 17 1991 in New York City*	date_of_birth place_of_birth	date_of_birth : *November 17 1991* place_of_birth : *New York City*
5	*I work for Microsoft as a software engineer*	profession employment_history	profession : *software engineer* employment_history : *Microsoft*

Table 3.1: Example Utterances with Semantic Space

the snippets that the search engine returns to create natural language examples that can be used as the training data for each relation, based on the earlier work (Hakkani-Tür et al., 2013). We further refine and augment the annotations of these examples, such as there are more than one relation instances in a snippet, which is similar to Heck and Hakkani-Tür (2012) and Tür et al. (2012).

This paradigm of constructing personal knowledge graphs in SLU can enhance the user experiences, since the SLU component knows more about the user's relationships and behaviors. In addition to customizing knowledge about users, it can also help enhance the performance of SLU systems from many aspects. For example, the SLU component may not appropriately respond to an utterance like "*show me the direction to my daughter's school*" previously. But once the SLU has built a user-centered knowledge graph, where "*my daughter's school*" has been associated with the address of the user's daughter's school, the SLU is able to interpret more utterances and act accordingly by taking advantage of more knowledge about the user. Moreover, once the VPA constructs a user-centric knowledge graph for each user, a global knowledge network may be populated by aggregating and integrating personal knowledge graphs through entity linking.

3.2 Framework

In this work, we align our SLU semantic space with the back-end semantic knowledge repositories such as Freebase and aim to identify knowledge graph relations invoked in user's utterances. To achieve this goal, we propose the statistical language understanding framework, as shown in Figure 3.2, with three key language understanding components: *Personal Assertion Detection*, *Relation Detection*, and *Slot Filling*. Each of these components

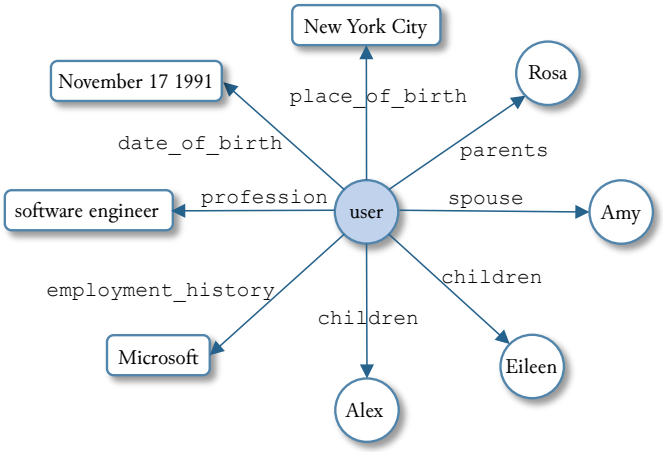

Figure 3.1: Example Personal Knowledge Graph

will be introduced in detail in the following subsections.

3.2.1 Personal Assertion Classification

This component aims to classify the spoken utterances into binary classes according to whether the utterance depicts personal facts. For example, one positive case could be "*i was born in 1999*", and, on the other hand, a negative instance could be "*how is the weather today?*". We formulate this problem as a binary classification task and apply Support Vector Machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) framework to perform the classification.

We use linear kernels as provided in the SVM^{light} (Joachims, 1999) package, since they are extremely efficient. The features include the ngrams (n = 1,2,3), stems, part-of-speech tags, and their combinations. The outputs of this stage provide us with coarse-grained information on whether we could further extract fine-grained personal factual relations from next two levels.

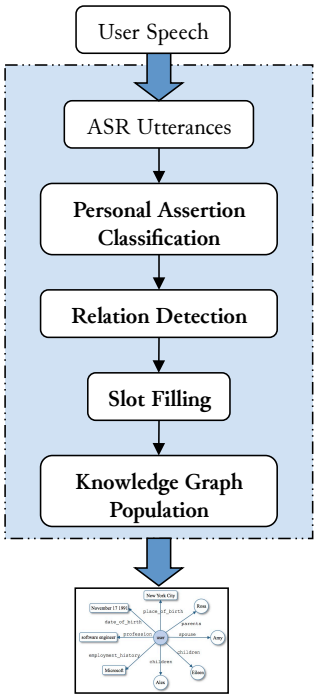

Figure 3.2: Framework of Personal Knowledge Graph Construction

3.2.2 Relation Detection

Relation detection aims to determine which relations in the part of knowledge graph related to the utterance have been invoked in the user utterances. For example, Table 3.1 shows example utterances that invoke various relations in the knowledge graph, and one utterance can also invoke more than one relation. Hence, the detection of the relation being invoked in the utterance is necessary for formulating the query to the back-end. We frame this subtask as a multi-class classification problem, and we also apply the SVM$^{\text{light}}$ package to classify each utterance into one or more relation classes. But instead of directly using the extended algorithm, SVM$^{\text{multiclass}}$, for multi-class scenarios, we still apply the binary, linear kernels in the SVM$^{\text{light}}$ package through a *one-vs-rest* approach.

We construct k SVM models where k is the number of relation classes. The ith SVM is trained with all the examples in the ith class with positive labels, and all other examples with negative labels. Then we apply all k SVM models to each utterance to determine which relations are invoked in it. The features also include the ngrams (n = 1,2,3), stems, part-of-speech tags, and their combinations. Depending on whether in-domain annotated Cortana utterances are available or not, the models can be trained in two ways:

- Case 1: (Supervised Baseline) Use only the in-domain annotated data for training and testing;
- Case 2: (Unsupervised) In cases where there is absolutely no in-domain annotated data, the distantly mined data can be used to build relation detection SVM models;

The formulation of the complete query to the back-end requires detection of the invoked entities in the user's utterance, in addition to detecting the graph relations that are invoked. We will extract the specific entities or arguments of detected relations with the following Slot Filling component.

3.2.3 Slot Filling

The semantic structure of an application domain is defined in terms of the semantic frames. The semantic frame contains several typed components called "slots". The task of slot filling is then to instantiate the semantic frames. Check Table 3.1 for slot filling in the example utterances. In this case, the semantic frame is represented as a flat list of attribute-value pairs, similar to Pieraccini and Levin, 1995. In this study, we follow the

popular IOB (in-out-begin) format in representing the data and use CRF++[2], an open source implementation of CRFs. Similarly, the features include the ngrams (n = 1,2,3), stems, part-of-speech tags, and their combinations.

3.2.4 Knowledge Graph Population

Once the relations and the associated entities or arguments are identified from the utterances, the user-centered personal knowledge graph would be populated with the newly extracted information and get updated. Then if the user intends to talk more about himself/herself, the system will repeat the above procedures to integrate more personal facts into the current knowledge graphs.

3.3 Data Collection

In this study, we utilize the semantic space that is defined in a knowledge base, or a triple store, such as *people.person* related facts in Freebase, for the SLU model to be built.

These semantic ontologies are not only used by search engines, which try to semantically parse them, but also by the authors of the in-domain web pages (such as imdb.com) for better visibility. While the details of the semantic web literature is beyond the scope of this chapter, it is clear that these kinds of semantic ontologies are very close to the semantic ontologies used in goal-oriented natural dialog systems and there is a very tight connection between the predicate/argument relations and intents, as explained below.

To create a training data set for our framework, we mine training examples by searching on the web for entity pairs that are related to each other in the knowledge graph.

2. http://crfpp.googlecode.com

As in the earlier work (Hakkani-Tür et al., 2013; Heck and Hakkani-Tür, 2012), we extract a set of entity pairs in a given domain that are connected with a specific relation from the knowledge base[3]. Our approach for mining examples guided by relations in the knowledge base is similar to (Krishnamurthy and Mitchell, 2012), but we directly detect relations invoked in user utterances, instead of parsing utterances with a combinatory categorical grammar (Steedman, 1996). Furthermore, we enhance our data with web search queries which are inquiring similar information as dialog system users.

Assume AS is the set of all snippets returned for the pair of entities a and b via web search[4]. We choose a subset of AS, SAS, that include snippets with both entities: $SAS = \{s : s \in AS \wedge includes(s, a) \wedge includes(s, b)\}$, where $includes(x, y)$ is true if string x contains y as a substring. One approach is using the complete strings of the snippets for each relation as training examples. However, the snippets can contain more than one correct relation tuples. In order to capture more relations in the mined snippet sentences, we post-process these sentences to augment the relation tags from Freebase, since many crawled instances actually contain more than one relation. (Even though we cannot guarantee that the augmented relations are "complete", because the Freebase is not complete as well as our collected data.) For example, we extract two relations regarding "Jacques Berthier", which are date_of_birth(*February 10, 1916*) and place_of_birth(*Paris, France*). This newly added step would generate the following two instances with all corresponding tags rather than two instances with incomplete tags: *Jacques Berthier was born on <date_of_birth>February 10, 1916</date_of_birth> in <place_of_birth>Paris, France</place_of_birth>*.

3. http://www.freebase.com

4. In this work, we use the Microsoft Bing search engine and download the top 10 results for each entity pair.

Category	Data	Number
Positive	web mined snippets	72,820
	pattern mined utterances	12,989
Negative	Cortana domain data	150,915

Table 3.2: Training Data for Personal Assertion Classification

3.4 Experiments

3.4.1 Evaluation Dataset

We first create a set of test examples to evaluate each key component of the proposed framework. To extract a set of testing instances, we have collected a total of 10 million utterances from Microsoft conversational understanding, Cortana, query logs. In order to mine real cases that are personal assertions and contain personal factual relations, we use 7 simple yet general patterns to extract a candidate pool, where the patterns are "*i am a **", "*i am from **", "*i have a **", "*i live **", "*i was born **", "*i work **", and "*my **". Then we randomly sample a subset of the pooled candidate utterances, and manually annotated each utterance with three levels of annotations, corresponding to the three main components of our proposed framework: (1) whether the utterance is a personal assertion; (2) the relations invoked in the utterance; and (3) the entities or argument of the invoked relations in the utterance. The final set of annotated data consists of 12,989 examples of personal assertions, among which 1,811 utterances contain at least one of the predefined relations, while the remaining 11,178 instances do not. We then experimentally investigate the performance of each key component based on this evaluation data set.

3.4.2 Personal Assertion Classification

To evaluate the performance of the Personal Assertion Classification component, a 10-fold cross-validation approach is applied on a combined data set, which contains the automatically mined snippets from the web, the annotated utterances from Cortana query logs, and a subset of Cortana related in-domain data. The Cortana related in-domain data consists of utterances in 7 distinct domains such as "weather" or "calendar". We use this data as negative assertion examples, while we label both snippets and annotated utterances as positive training data. Table 3.2 shows the number of examples from each data source. Then the data set is randomly split into 10 equal size subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds are then combined to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. Among total $236,724$ data samples that are semi-automatically collected, $234,650$ instances are correctly classified while only $2,074$ are classified with wrong class labels, which achieves 99.12% accuracy. This demonstrates the reliable performance of this SVM-based Personal Assertion classifier.

3.4.3 Relation Detection

In order to measure the quality and effectiveness of Relation Detection component, the models have been trained using the snippets mined from the web and the annotated

Cortana utterances in two scenarios, depending on whether in-domain annotated data is available or not:

- Case 1: (Supervised Baseline) Only use the in-domain annotated Cortana utterances for both training and testing, where a 2-fold cross-validation approach is applied. For each fold, annotated utterances are randomly assigned to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). Then the model is trained on d_0 and tested on d_1, following by being trained on d_1 and tested on d_0. This has the advantage that our training and test sets are both large, and each data point is used for both training and validation;
- Case 2: (Unsupervised) To mimic the cases where there is absolutely no in-domain annotated spoken data, the snippets crawled from the web are used to build models, and gauge the model performance on the annotated Cortana utterances;

For evaluation, we used Precision@N (P@N), where N is the number of positive examples for that relation in the test set. Table 3.3 shows the detailed results in each above case, where only n-gram features are used. The supervised method provides the upper bound of 84.32% P@N, based on manual annotations. Using the proposed unsupervised approach results in a bootstrap model achieving 42.85% P@N overall. However for certain classes such as sibling or spouse, the model has performed on par with the supervised approach. For relations, requiring a named entity such as location for place_of_birth or date for date_of_birth, using a generic named entity tagger should help improve the performance. This is left as future research. Another promising direction is adapting this bootstrap model with supervised data, using an online learning mechanism, drawing

learning curves for each relation. We suspect that with a few manually tagged examples, some relation types may improve significantly, such as employment_history.

3.4.4 Slot Filling

The Slot Filling results in each above case are also shown in Table 3.3. For slot filling we only used the supervised approach, since the semantic annotation mechanisms of the snippets and the evaluation set are different, as they belong to different genre (e.g., *Jacques Berthier is the son of <parents>Paul Berthier<parents>* vs. *my <parents>father<parents> is old*). For evaluation, the slot F-measure is used, following the literature (Raymond and Riccardi, 2007) using the CoNLL evaluation script[5]. We can see that the supervised approach can achieve 68.34% F-measure in the overall performance. For most relation types, where the context is obvious, the system achieves reasonable performance levels with minimal annotations. There are a few relation types, where the task is nontrivial such as profession relation, since profession may get invoked with a much larger pool of expressions, such as "computer research scientist", "helicopter trainer", "international standard ballroom dancer", and so on, which cannot easily get trained from a small in-domain data. As part of future research, it is interesting to extract these patterns from the automatic annotations we mined from the snippets. Similarly, the named entity features would help improving the overall performance.

5. http://www.cnts.ua.ac.be/conll2000/chunking/output.html

3.5 Conclusion

In this chapter, we have presented a novel SLU framework aiming to construct personal (user-centric) knowledge graphs in spoken utterances. This approach contains three main language understanding components: *Personal Assertion Classification*, *Relation Detection*, and *Slot Filling*. Our experimental results have proven the effectiveness of the proposed scheme on all three levels. While relation detection and slot filling have been studied in many SLU tasks, to the best of our knowledge, this is a pioneering study for systematically building personal knowledge graphs in human/machine conversational systems.

Relation Type	**Count**	**Relation Detection**		**Slot Filling**		
		unsupervised	supervised	supervised		
		Precision@Count (%)	Precision@Count (%)	Precision (%)	Recall (%)	F-Measure (%)
place_of_birth	8	0.00	0.00	0.00	0.00	0.00
religion	8	0.00	50.00	0.00	0.00	0.00
ethnicity	17	0.00	70.59	100.0	17.65	30.00
employment_history	40	7.50	52.50	50.00	12.50	20.00
nationality	47	0.00	63.83	75.00	82.98	78.79
profession	61	0.00	54.10	50.00	1.64	3.72
gender	63	6.35	82.54	90.91	47.62	62.50
date_of_birth	73	46.58	75.34	56.25	36.99	44.63
places_lived	121	2.48	68.59	69.91	65.29	67.52
sibling_s	248	86.29	90.32	85.92	71.08	77.80
children	260	23.08	87.31	80.92	47.31	59.71
parents	401	19.95	86.78	83.97	65.17	73.39
spouse_s	464	82.11	94.39	86.81	68.10	76.33
Total	1811	42.85	84.32	82.01	58.58	68.34

Table 3.3: Performance of Relation Detection and Slot Filling

Chapter 4

Confidence Estimation for Knowledge Base Population ♣

4.1 Introduction

Despite significant progress in recent years, Information Extraction (IE) technologies are still far from completely reliable. Errors result from the fact that language itself is ambiguous as well as methodological and technical limitations (Gandrabur et al., 2006). Therefore, evaluating the probability that the extracted information is correct can contribute to improve IE system performance. Confidence Estimation (CE) is a generic machine learning approach for estimating the probability of correctness of the

. ♣ This work has been published in "*Confidence Estimation for Knowledge Base Population*". Xiang Li and Ralph Grishman. *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, 2013.

outputs, and usually adds a layer on top of the baseline system to analyze the outputs using additional information or models (Gandrabur et al., 2006). There is previous work in IE using probabilistic and heuristic methods to estimate confidence for extracting fields using a sequential model, but to the best of our knowledge, this work is the first probabilistic CE model for the multi-stage systems employed for the Knowledge Base Population (KBP) Slot Filling task.

The goal of Slot Filling (SF) is to collect information from a corpus of news and web documents to determine a set of predefined attributes ("slots") for given person and organization entities (Ji et al., 2011) (Section 4.2). Many methodologies have been used to address the SF task, such as Distant Supervision (Min et al., 2012) and Question Answering (Chen et al., 2010b), and each method has its own strengths and weaknesses. Many current KBP SF systems actually consist of several independent SF pipelines. The system combines intermediate responses generated from different pipelines into final slot fills. Since these intermediate outputs may be highly redundant, if confidence values can be associated with the outputs, it will definitely help re-ranking and aggregation. For this purpose, we require comparable confidence values from disparate machine learning models or different slot filling strategies.

Robust probabilistic machine learning models are capable of accurate confidence estimation because of their intelligent handling of uncertainty information. In this chapter, we use the Maximum Entropy (MaxEnt) framework (Berger et al., 1996) to automatically predict the correctness of KBP SF intermediate responses (Section 4.3). Results achieve an average precision of 83.5%, Pearson's r of 54.2%, and 2.3% absolute improvement in final F-measure score through a weighted voting system (Section 4.4).

4.2 KBP Slot Filling

4.2.1 Task Definition

The Knowledge Base Population (KBP) track, organized by U.S. National Institute of Standards and Technology (NIST)'s Text Analysis Conference (TAC), aims to promote research in discovering information about entities and augmenting a Knowledge Base (KB) with this information (Ji et al., 2010). KBP mainly consists of two tasks: Entity Linking, linking names in a provided document to entities in the KB or NIL; and Slot Filling (SF), extracting information about an entity in the KB to automatically populate a new or existing KB. As a new but influential IE evaluation, Slot Filling is a challenging and practical task (Min and Grishman, 2012).

The Slot Filling task at *KBP2012* provides a collection of 3.7 million newswire articles and web texts as the source corpus, and an initial KB derived from the Wikipedia infoboxes. In such a large corpus, some information can be highly redundant. Given a list of person (PER) and organization (ORG) entity names ("queries"), SF systems retrieve the documents about these entities in the corpus and then fill the required slots with correct, non-redundant values. Each query consists of the name of the entity, its type (PER or ORG), a document (from the corpus) in which the name appears, its node ID if the entity appears in the provided KB, and the slots which need not be filled. Along with each slot fill, the system should also provide the ID of the document that justifies this fill. If the system does not extract any information for a given slot, the system just outputs "NIL" without any document ID. The task defines a total of 42 slots, 26 for person entities and 16 for organization entities. Some slots are single-valued, like

"per:date_of_birth", which can only accept at most a single value, while the other slots, for example "org:subsidiaries", are list-valued, which can take a list of values. Since the overall goal is to augment an existing KB, the redundancy in list-valued slots must be detected and avoided, requiring a system to identify different but equivalent strings such as, "United States" and "U.S.". More information can be found in the task definition (Ji et al., 2010).

4.2.2 Baseline System Description

We use a slot filling system that has achieved highly competitive results (ranked top 2) at the *KBP2012* evaluation as our baseline. Like most SF systems, our system has three basic components: Document Retrieval, Answer Extraction, and Response Combination. Our SF system starts by retrieving relevant documents based on a match to the query name or the results of query expansion. Then our system applies a two-stage process to generate final slot fills: Answer Extraction, which produces intermediate responses from different pipelines, and Response Combination, which merges all intermediate responses into final slot fills. Answer extraction begins with document pre-processing, such as part-of-speech tagging, name tagging, and coreference resolution. Then it uses a set of 6 SF pipelines operating in parallel on the retrieved documents to extract answers. Our pipelines consist of two that use hand-coded patterns, two pattern-based slot fillers in which the patterns are generated semi-automatically from a bootstrapping procedure, one based on name coreference, and one distant-supervision based pipeline. The result of this stage is a set of intermediate slot responses, potentially highly redundant. Next, Response Combination validates answers and eliminates redundant answers to aggregate

all intermediate responses into final slot fills, where the best answer is selected for each single-valued slot and non-redundant fills are generated for list-valued slots. More details about our KBP Slot Filling system can be found in the system description paper (Min et al., 2012).

4.3 Confidence Estimation Model

Our confidence estimation model is based on the Maximum Entropy (MaxEnt) framework, a probabilistic model able to incorporate all features into a uniform model by assigning weights automatically. We implement a mix of binary and real-valued features from different aspects to estimate confidence of each intermediate slot filling response under a consistent and uniform standard, incorporating four categories of features:

- **Response Features** extract features from the slot and the *Response* context.
- **Pipeline Features** indicate how well each pipeline performed previously.
- **Local Features** explore how *Query* and *Response* are correlated in the supporting context *Sentence*.
- **Global Features** detect how closely *Query* correlates with *Response* in the global context.

Each specific feature in the above categories is listed in Table 4.4, where *Q* refers to a person or organization *Query*; *R* indicates the pipeline-generated *Response* for a particular slot of a query; and *S* represents the *Sentence* that supports the correctness of the *Response*. It is worth noting that the features *cond_prob_givenQ*, *cond_prob_givenR*, and *mutual_info*

are calculated based on the number of documents that can be retrieved by *Query*, *Response*, or both of them.

4.4 Experiments

	PER#	**ORG#**	**Total#**	**Response#**
KBP2010	50	50	100	7917
KBP2011	50	50	100	14976
KBP2012	40	40	80	8989
total	140	140	280	31878

Table 4.1: Number of Queries and Number of Intermediate Responses from Each Year Data

We have collected and merged the three years' KBP SF evaluation data, which consists of a total of 280 queries, and Table 4.1 lists the number of person and organization queries as well as the number of intermediate responses from each year. There are in total 31878 intermediate responses generated by 6 different pipelines from our SF system. We trained our CE model and measured the confidence values through a 10-fold cross-validation, so that each fold randomly contains 14 person queries and 14 organization queries with their associated intermediate responses. Then for each iteration, the CE model is trained on 9 folds and approximates the confidence values in the remaining fold; it assigns the probability of each intermediate response being correct as confidence.

4.4.1 Voting Systems

To evaluate the reliability of confidence values generated by this model, we used the weighted voting method to investigate the relationship between the confidence values and performance.

4.4.1.1 Baseline Voting System

Our baseline SF system applies a basic plurality voting to combine all intermediate responses to generate the final response submission. This voting system simply counts the frequencies of each response entity, which is a unique response tuple in the form <Query_ID, Slot_Name, Response_Fill>. For a single-valued slot of a query, the response with the highest count is returned as the final response fill. For the list-valued slots, all non-redundant responses are returned as the final response fills. In this basic voting system, each intermediate response contributes equally.

4.4.1.2 Weighted Voting System

Weighted voting is based on the idea that not all the voters contribute equally. Instead, voters have different weights concerning the outcome of an election. In our experiment, voters are all of the intermediate responses generated by all pipelines, and the voters' weights are their confidence values. We set a threshold τ in this weighted voting system, where those intermediate responses with confidence lower than τ would be eliminated. For each response entity, this weighted voting system simply sums all the weights of the intermediate responses that support this response entity as its weight. Then for a single-valued slot of a query, it returns the response with the highest weight as the final slot fill,

while it returns all non-redundant responses as the final slot fills for the list-valued slots. The maximum confidence ψ of supporting intermediate responses is used as the final confidence for that slot fill. We also set a threshold η (optimized on a validation data set), where the final slot fills with confidence ψ lower than η would not be submitted.

4.4.1.3 Results

Table 4.2 compares the results of this weighted voting system (with τ = 0, η = 0.17) and the baseline voting system, where the responses were judged based only on the answer string, ignoring the document ID. As we can see, the weighted voting system achieves 2.3% absolute improvement in F-measure over the baseline, at a 99.8% confidence level according to the Wilcoxon Matched-Pairs Signed-Ranks Significance Test. Precision obtains 9.0% absolute improvement with only a small loss of 0.5% in Recall.

	Precision	**Recall**	**F-measure**
Baseline	0.351	**0.246**	0.289
Weighted	**0.441**	0.241	**0.312**

Table 4.2: Results Comparison between Baseline Voting System and Weighted Voting System

Figure 4.1 summarizes the results of this weighted voting system with different threshold τ settings. When τ is raised, Precision continuously increases to around 1, while Recall gradually decreases to 0.

In addition to improving overall performance, the confidence estimates can be used to convey to the user of slot filling output our confidence in final individual slot fills. After

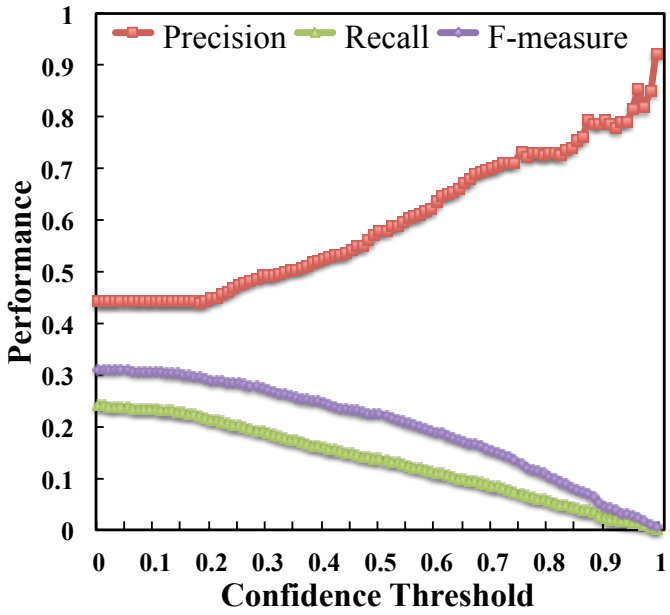

Figure 4.1: Impact of Threshold Settings

the intermediate responses are combined by the above weighted voting system (setting τ and η as 0), we divide the range of confidence values (0 to 1) into 10 equal intervals (0 to 0.1, 0.1 to 0.2, and so on) and categorize these final slot fills by their confidence values. Then for each category, the final slot fills are scored in Precision. Figure 4.2 strongly demonstrates that the slot fills with higher confidence consistently generate more precise answers, indirectly validating the reliability of the confidence estimates.

4.4.2 Evaluation

We use two further methods to evaluate the quality of confidence estimation in a more direct way. The first method is *Pearson's r*, a correlation coefficient ranging from -1 to 1 that measures the correlation between a confidence value and whether or not

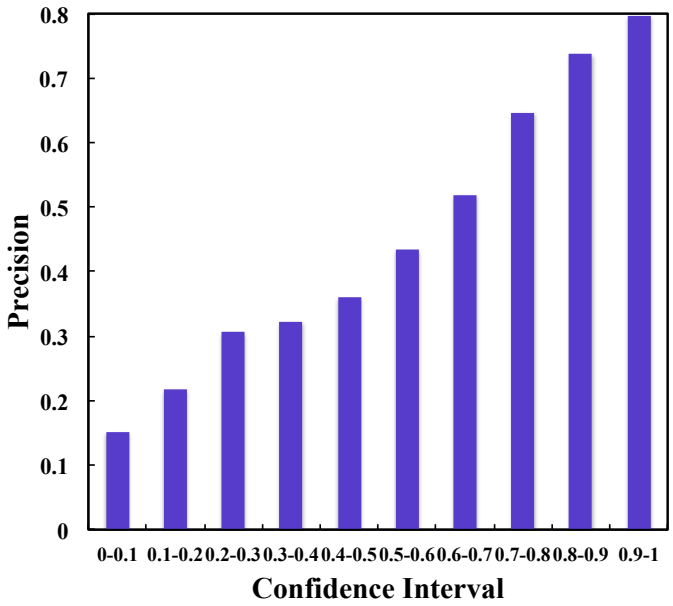

Figure 4.2: Performance of Confidence Intervals

the instance is correct. It is widely used in the sciences as a measure of linear dependence between two variables. The second method is *average precision*, used in the Information Retrieval community to evaluate a ranked list. It calculates the precision at each point in the ranked list where a relevant document is found and then averages these values. Instead of ranking documents by their relevance scores, the intermediate responses are ranked by their confidence values.

	Avg. Prec	**Pearson's r**
RANKED	**0.835**	**0.542**
RANDOM	0.525	0.001
WORSTCASE	0.330	-

Table 4.3: Evaluation of Confidence Estimates

Table 4.3 shows the Pearson's r and average precision results for all intermediate responses, where RANKED ranks the responses based on their confidence values; RANDOM assigns confidence values uniformly at random between 0 and 1; WORSTCASE ranks all incorrect responses above all correct ones.

Applying the features separately, we find that *slot_response_length* and *response_doc_num* are the best predictors of correctness. *dpath_length* (the length of the shortest dependency path between query and response) is also a significant contributor. Among the features, only *NE_margin* seeks to directly estimate the confidence of a pipeline component, and it makes only a minimal contribution to the result. Overall this shows that confidence can be predicted quite well from features of the query and response, their appearance in the corpus, and prior IE system performance, without modeling the confidence of individual pipeline components.

4.5 Conclusion

We have presented our Maximum Entropy based confidence estimation model for information extraction systems. The effectiveness of this model has been demonstrated in the challenging Knowledge Base Population Slot Filling task, where a weighted voting system achieves 2.3% absolute improvement in F-measure score based on the confidence estimates. A strong correlation between the confidence estimates in KBP slot fills and the correctness has also been proved by obtaining an average precision of 83.5% and Pearson's r of 54.2%.

Category	Feature	Description
Response Features	slot_name	The slot name
	slot_response_length	The conjunction of the length of *R* and the slot name
	name_response_slot	The slot requires a name as the response
Pipeline Features	pipeline_name	The name of pipeline which generates *R*
	pipeline_precision	The Precision of the pipeline which generates *R*
	pipeline_recall	The Recall of the pipeline which generates *R*
	pipeline_fmeasure	The F-measure of the pipeline which generates *R*
Local Features	sent_contain_QR	*S* contains both original *Q* and *R*
	sent_contain_ExQR	*S* contains both co-referred *Q* or expanded *Q* and *R*
	dpath_length	The length of shortest dependency path between *Q* and *R* in *S*
	shortest_dpath	The shortest dependency path between *Q* and *R* in *S*
	NE_boolean	*R* is a person or organization name in *S*
	NE_margin	The difference between the log probabilities of this name *R* and the second most likely name
	n-gram	Tri-gram context window associated with part-of-speech tags containing *Q* or *R*
	genre	The supporting document is a newswire or web document
Global Features	query_doc_num	The number of documents retrieved by *Q*
	response_doc_num	The number of documents retrieved by *R*
	co-occur_doc_num	The number of documents retrieved by the co-occurrences of *Q* and *R*
	cond_prob_givenQ	The conditional probability of *R* given *Q*
	cond_prob_givenR	The conditional probability of *Q* given *R*
	mutual_info	The Point-wise Mutual Information (PMI) of *Q* and *R*

Table 4.4: Features of Confidence Estimation Model

Chapter 5

Rich Annotation Guided Learning †

5.1 Introduction

Statistical Natural Language Processing (NLP) has two crucial aspects: (1) good choice of machine learning algorithms; (2) good feature engineering. In particular, (2) significantly affects the performance of systems. Linguistic annotation is a fundamental and crucial step of supervised learning. However, feature engineering remains a challenging task because it encompasses feature design, feature selection, feature induction and studies of feature impact, all of which are very time-consuming, especially when there are a lot of data or errors to analyze. As a result, in a typical feature engineering process, the system developer is only able to select a representative data set as the development

. † This work has been published in "*Rich Annotation Guided Learning*". Xiang Li, Heng Ji, Faisal Farooq, Hao Li, Wen-Pin Lin, and Shipeng Yu. *Invited Paper for International Journal On Advances in Intelligent Systems, v 5 n 3&4*, 2012.

set and analyze partial errors. Moreover, annotated corpora are usually prepared by a separate group of human annotators before system development. As a result, almost all previous NLP systems only utilized direct manual labels for training, while ignoring the valuable knowledge that human annotators have learned and summarized from corpora preparation. In fact, compared to system developers who normally design features based on partial data analysis, human annotators are usually more knowledgeable because they need to go through the entire data set and restrictively follow annotation guidelines.

If we consider an NLP system as a "*student*" while a human annotator as a "*teacher*", then the homework answer keys or grades (i.e., basic annotations) are just a small part of the teacher's role. Besides grading, a teacher also provides explanations about why an answer is wrong, comments about what kind of further knowledge the student can benefit from, and so on. Similarly, besides the textbook, a teacher can also highlight part of the content to compose lecture notes. All of this additional evidence and comments can be considered as "*rich annotations*". When human annotators produce some certain labels, they must also have certain evidence for the annotation they provide for each instance. Therefore, it would not cost them much extra time to highlight the evidence in contexts, or generalize enough knowledge to suggest what kind of linguistic features might be helpful for system development.

In this chapter, we propose a new and general Rich Annotation Guided Learning (RAGL) framework in order to fill the gap between an expert annotator and a feature engineer. As an extension of the comment-guided learning framework proposed in our previous work (Li et al., 2011), this new framework aims to enrich features with the guidance of all levels of *rich annotations* from human annotators. In order to verify

the efficacy of this approach, we conducted case studies on four distinct applications in various domains in our previous work (Li et al., 2012): medical concept extraction, name translation, slot filling and event modality detection. Empirical studies demonstrate that with slightly additional annotation time, we can significantly improve the performance for all tasks. For example, the case study on event modality detection demonstrated that the system trained from rich annotations can save 65% annotation cost in order to obtain the same performance as using basic annotations. (Li et al., 2012)

The rest of this chapter is structured as follows. Section 5.2 presents an overview of our new learning framework incorporating rich annotations from human annotators. Section 5.3 presents the detailed algorithms to incorporate rich annotations from Level 3 and a Knowledge Base Population Slot Filling case study. Section 5.4 then concludes the chapter.

5.2 Rich Annotation Guided Learning

In this section we present the general framework of incorporating rich human annotations into the learning process. In Table 5.1, we aim to formalize the mapping of some essential elements in human learning and machine learning for NLP.

In a regular annotation interface, a human annotator is only asked to provide the final labels (e.g., 0/F or 1/T in binary settings). We call this basic annotation '**Level 0**'. We can see that among these elements, little study has been conducted on incorporating rich annotations from human annotators. In most cases it was not the obligation of the human annotators to write down their evidence or comments during annotation. In contrast, the human learning scenario involves more interactions. However, we can assume

Human Learning	Machine Learning for NLP	Approaches
student	system	baseline NLP system
teacher / teaching assistant	human annotator / human assessor	
textbook / homework answer key	training data with basic annotations	
graded homework		
lecture notes / graded homework with comments	**training data with rich annotations**	**our proposed approach**
errorneous homework set	negative samples / errors	transformation based learning
homework review against lecture	system output with background documents	recognizing textual entailment
group study	pooled system responses	voting, learning-to-rank

Table 5.1: Some Elements in Human Learning and Machine Learning for NLP

that any annotator is able to verify and comment on his/her judgment. We propose to unleash the powerful knowledge based on rich annotations from human annotators on various deeper levels:

- **Level 1**: Ask an annotator to verify a label by providing surface evidence (e.g., highlighting indicative contexts) (Yu et al., 2011);
- **Level 2**: Ask an annotator to verify a label by providing deep evidence (e.g., generalizing indicative contexts) (Li et al., 2012);
- **Level 3**: Ask an annotator to provide comments about linguistic features or resources that might be helpful for system development (Li et al., 2011).

Based on this intuition we propose a new Rich Annotation Guided Learning (RAGL) paradigm as shown in Figure 5.1.

5.3 Level 3: Expensive Rich Annotations

5.3.1 Algorithm Overview

Recently many NLP tasks have moved from processing hundreds of documents to large-scale or even web-scale data. Once the collection grows beyond a certain size, it is not feasible to prepare a comprehensive answer key in advance. Because of the difficulty in finding information from a large corpus, any manually-prepared key is likely to be quite incomplete. Instead, we can pool the responses from various systems and have human annotators manually review and judge the responses. Assessing pooled

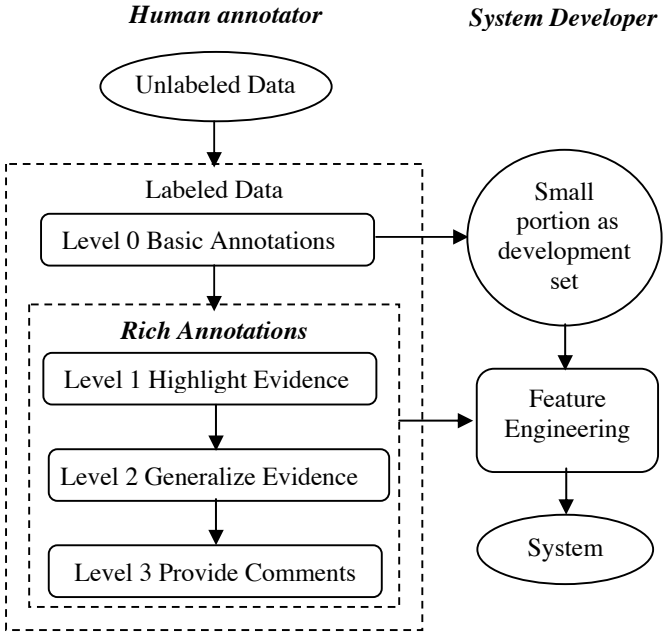

Figure 5.1: Rich Annotation Guided Learning Framework

system responses as opposed to identifying correct answers from scratch has provided a promising way to generate training data for NLP systems. Usually such tasks require deep knowledge beyond surface information provided by Level 0 (basic annotations), Level 1 (highlighting the part of the text that leads the annotator to the conclusion), and Level 2 (generalizing the indicative context such as providing some categories of words and contexts). In contrast, the comments from Level 3 can be exploited as features for automatic assessment.

This algorithm aims to extensively incorporate all comments from an old development data set (i.e., "*old homework*" in human learning) into an automatic correction component. This assessor can be applied to improve the results for a new test data set (i.e., "*new homework*" in human learning).

The algorithm can be summarized as follows.

1. The pipeline starts by running the baseline system to generate results. In this step we can also add the outputs from other systems (i.e., classmates in human learning) or even human annotators (i.e., Teaching Assistant (TA) in human learning). We will present one case study on slot filling which incorporates these two additional elements.

2. We obtain comments from human annotators on a small development set D^i. Each time we ask a human annotator to pick N[1] random results and provide a new comment on each result. One could impose some pre-defined format or template restrictions for the comments, such as marking the indicative words as rich annotations and encoding them as features. Nonetheless, we found that most of the expert comments are rather implicit and even require global knowledge. Nonetheless these comments represent general solutions to reduce the common errors from the baseline system.

3. We encode these comments into features. We then train a Maximum Entropy (MaxEnt) based automatic assessor A^i using these features. For each response generated from the baseline system, A^i can classify it as correct or incorrect. We choose a statistical model instead of rules because heuristic rules may overfit a small sample set and highly dependent on the order. In contrast, a MaxEnt model has the power of incorporating all comments into a uniform model by assigning weights automatically. In this way we can integrate assessment results tightly with comments during MaxEnt model training.

4. Finally, A^i is applied as a post-processing step to any new data set D^{i+1}, and filters out those results judged as incorrect.

The algorithm can be conducted in an iterative fashion. For example, human an-

1. $N = 3$ in this chapter, the value of 3 was arbitrarily chosen; variations in this number of clusters produce only small changes in performance

notators can continue to judge and provide comments for D^{i+1} and we can update the automatic assessor to A^{i+1} and apply it to a new data set D^{i+2}, and so on. We conduct a case study on a challenging residence slot filling task (5.3.2).

5.3.2 Slot Filling

In this section, we shall apply Level 3 annotations to a more challenging task of slot filling and investigate the detailed aspects of human-comment-guided learning by comparing it with alternative methods. In the TAC Slot Filling task as described in Section §4.2.1, we choose three residence slots for person entities ("*countries_of_residence*", "*stateorprovinces_of_residence*" and "*cities_of_residence*") for our case study, because they are one group of the most challenging slot types for which almost all systems perform poorly (less than 20% F-measure).

5.3.2.1 Baseline Systems

We use a slot filling system (Chen et al., 2010b) which achieved highly competitive results (ranked at top 3 among 31 submissions from 15 teams) at the KBP2010 evaluation as our baseline. This system includes multiple pipelines in two categories: two bottom-up IE based approaches (pattern matching and supervised classification) and a top-down Question Answering (QA) based approach that searches for answers constructed from target entities and slot types. The overall system begins with an initial query processing stage where query expansion techniques are used to improve recall. The best answer candidate sets are generated from each of the individual pipelines and are combined in a statistical re-ranker. The resulting answer set, along with confidence values are then

processed by a cross-slot reasoning step based on Markov Logic Networks (Richardson and Domingos, 2006), resulting in the final system outputs. In addition, the system also exploited external knowledge bases such as Freebase (Bollacker et al., 2007) and Wikipedia text mining for answer validation.

In order to check how robust the RAGL assessor is, we also run it on some other anonymous systems in KBP2010 with representative performance (high, medium and low).

5.3.2.2 Comments and Feature Encoding

The detailed comments used for our slot filling experiment are as follows.

- **Comment 1: "*this answer is not a geo-political name*"**

  This comment is intended to address some obvious errors which could not be Geo-Political (GPE) names in any contexts. In order to address this comment, we apply a very large gazetteer of GPE hierarchy (countries, states and cities) from the geonames website [2] for answer validation.

- **Comment 2: "*this answer is not supported by this document*"**

  Some answers obtained from Freebase may be incorrect because they are not supported by the source document. Answer validation was mostly conducted on the document basis, but for the residence slots we need to use sentence-level validation. In addition, some sentence segmentation errors occur in web documents. To address this comment, we apply a coreference resolution system (Ji et al., 2005) to the source document, and

2. http://www.geonames.org/statistics/

check whether any mention of the query entity and any mention of the candidate answer entity appear in the same sentence.

- **Comment 3: "*this answer is not a geo-political name in this sentence*"**

  Some ambiguous answers are not GPE names in certain contexts, such as "*European Union*". To address this comment, we extract the context sentences including the query and answer mentions, and run a name tagger (Grishman et al., 2005) to verify the candidate answer is a GPE name.

- **Comment 4: "*this answer conflicts with this system/other system's output*"**

  When an answer from our system is not consistent with another answer which appears often in the pooled system responses, this comment suggests us to remove our answer. In order to address this comment, we implemented a feature based on hierarchical spatial reasoning. We conduct majority voting on all the available system responses, and collect the answers with global confidence values (voting weights) into a separate answer set *ha*. Then for any candidate answer *a*, we check the consistency between *a* and any member of *ha* by name coreference resolution and part-whole relation detection based on the gazetteer of GPE hierarchy as described in Comment 1. For example, if "*U.S.*" appears often in *ha* we can infer "*Paris*" is unlikely to be a correct answer for the same query; on the other hand if "*New York*" appears often in *ha* we can confirm "*U.S.*" as a correct answer.

The detailed features converted from the above comments are summarized in Table 5.2.

Comments	Features
1	whether the answer is in the geo-political gazetteer
2	whether any mention of the query entity and any mention of the answer entity appear in the same sentence using coreference resolution
3	whether the answer is a GPE name by running name tagging on the context sentence
4	whether the answer conflicts with the other answers which received high votes accross systems using inferences through the GPE hierarchy

Table 5.2: Validation Features for Slot Filling

5.3.2.3 Data and Scoring Metric

During KBP2010, an initial answer key annotation was created by the Linguistic Data Consortium (LDC) through a manual search of the corpus, and then an independent adjudication pass was applied by LDC human annotators to assess these annotations together with pooled system responses to form the final gold-standard answer key. We incorporated the assessment comments for our system output on a separate development set (182 unique non-NIL answers in total) from KBP2010 training data set to train the automatic assessor. Then we conduct a blind test on the KBP2010 evaluation data set which includes 1.7 million newswire and web documents. The final answer key for the blind test set includes 81 unique non-NIL answers for 49 queries.

The number of features we can exploit is limited by the unknown restrictions of individual systems. For example, some other systems used distant learning based answer validation and so could not provide specific context sentences. Since comment 2 and comment 3 require context sentences, we trained one assessor using all features and tested it on our own system. Then we trained another assessor using only comment 1

and 4 and tested it on three other systems representing different levels of performance.

Equivalent answers (such as "*the United States*" and "*USA*") are grouped into equivalence classes. Each system answer is rated as correct, wrong, or redundant (an answer which is equivalent to another answer for the same slot or an entry already in the knowledge base). Given these judgments, we calculate the precision, recall and F-measure of each system, as defined by Ji et al. (2010, 2011).

5.3.2.4 Overall Performance

Table 5.3 shows the slot filling scores before and after applying the RAGL assessors (because of the KBP Track requirements and policies, we could not mention the specific names of other systems). The Wilcoxon Matched-Pairs Signed-Ranks Test show we can reject the hypothesis that the improvements using RAGL over our system were random at a 99.8% confidence level. It also indicates that the features encoded from comment 2 and comment 3 which require intermediate results such as context sentences helped boost the performance about 3.4%. We can see that although the other high-performing system may have used very different algorithms and resources from ours, our assessor still provided significant gains. Our approach improved the precision on each system (more than 200% relative gains) with some loss in recall. Since most comments focused on improving precision, F-measure gains for moderate-performing and low-performing systems were limited by their recall scores. This is similar to the human learning scenario where students from the same grade can learn more from each other than from different grades. In addition, the errors removed by our approach were distributed equally in newswire (48.9%) and web data (51.1%), which indicates the comments from human

annotators reached a good degree of generalization across genres.

Slot Filling Systems		Annotation Category	P (%)	R (%)	F (%)
Our system		Level 0	17.1	30.9	22.0
		Level 3 (f1+f4)	26.2	27.2	**26.7**
		Level 3 (full)	38.5	24.7	**30.1**
Other systems	High-Performing	Level 0	13.7	29.6	18.8
		Level 3 (f1+f4)	40.9	22.2	**28.8**
	Moderate-Performing	Level 0	12.2	7.4	9.2
		Level 3 (f1+f4)	35.7	6.2	**10.5**
	Low-Performing	Level 0	6.7	3.7	4.8
		Level 3 (f1+f4)	50.0	3.7	**6.9**

Table 5.3: Overall Performance of Slot Filling

5.3.2.5 Cost and Contribution of Each Comment

The comments from the RAGL assessor may reflect different aspects of the system. Therefore it will be interesting to investigate what types of comments are most useful and not costly. We did another experiment by applying one comment at a time into the assessor. Table 5.4 shows the results along with the cost of generating and encoding each comment (i.e., knowledge transferring to its corresponding feature), which was carefully recorded by the human annotators.

Table 5.4 indicates that every feature made contributions to precision improvement. Comment 1 (gazetteer-based filtering) only provided limited gains mainly because our

Annotations		Level 0	Level 3			
			f1	f2	f3	f4
Performance	P (%)	17.1	17.6	26.4	26.7	25.6
	R (%)	30.9	30.9	28.4	28.4	27.2
	F (%)	22.0	22.4	27.4	27.5	26.3
Cost	# samples reviewed	-	3	3	3	3
	providing comments (minutes)	-	3	3	3	3
	encoding comments (minutes)	-	30	240	60	30

Table 5.4: Cost and Contribution of Each Comment

own system already extensively used similar gazetteers for answer filtering. This reflects a drawback of our comment generation procedure - the assessor had no prior knowledge about the approaches used in the systems. Comment 2 (using coreference resolution to check sentence occurrence) took the most time to encode but also provides significant improvement. Comment 4 (consistency checking against responses with high votes) provided significant gains in precision (8.5%) but also some loss in recall (3.7%). The problem was that systems tend to make similar mistakes, and the human annotator was biased by those correct answers which appeared frequently in the pooled system output. However, Comment 4 was able to filter out many errors which are otherwise very difficult to detect. For example, because "*Najaf*" appears very often as a "*cities_of_residence*" in the pooled system responses, Comment 4 successfully removed six incorrect "*coun-*

tries_of_residence" answers for the same query: "*Syrian*", "*Britain*", "*Iranian*", "*North Korea*", "*Saudi Arabia*" and "*United States*". On the other hand, Comment 4 confirmed correct answers such as "*New York*" from "*Brooklyn*", "*Texas*" from "*Dallas*", "*California*" and "*US*" from "*Los Angeles*".

5.3.2.6 Impact of Data Size

We also did a series of runs to examine how our own system performed with different amounts of training data. The experiments of training the MaxEnt model with the above 4 validation features are summarized in Figure 5.2. It clearly shows that the learning curve converges quickly. Therefore, we only need a very small amount of training data (36 samples, 20% of total) in order to obtain similar gains (6.8%) as using the whole training set.

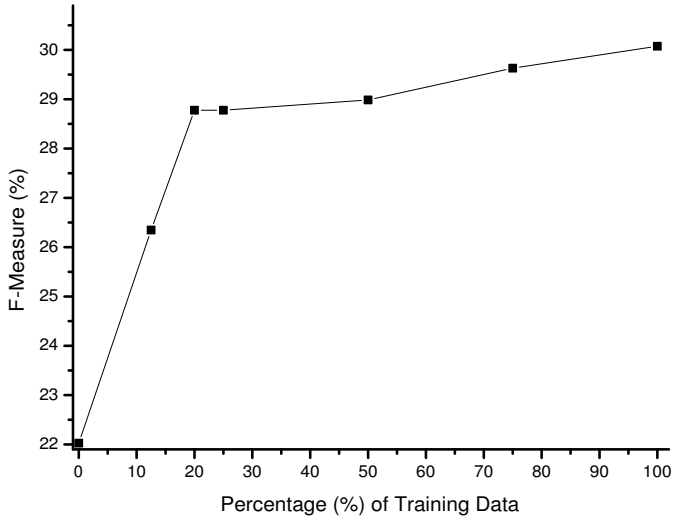

Figure 5.2: Impact of Training Data Size

5.3.2.7 Speed up Human Assessment

Human assessment for slot filling is also a costly task because it requires the annotators to judge each answer against the associated source document. Since our RAGL approach achieved positive impact on system output, can it be used to as feedback to speed up human assessment? We applied the RAGL assessor trained from comment 1 and comment 4 to the top 13 KBP systems for KBP2010 evaluation set. We automatically ranked the pooled system responses of residence slots according to their confidence values from high to low. For comparison, we also exploited the following methods:

- **Baseline**

  As a baseline, we ranked the responses according to the alphabetical order of slot type, query ID, query name and answer string and doc ID. This is the same approach used by LDC human annotators for assessing KBP2010 system responses.

- **Oracle (Upper-Bound)**

  We used an oracle (for upper-bound analysis) by always assessing all correct answers first.

Figure 5.3 summarizes the results from the above 3 approaches. For this figure, we assume a labor cost for assessment proportional to the number of non-NIL items assessed. Note that all redundant answers are also included in these counts because human annotators also spent time assessing them. This is only approximately correct; it may be faster (per response) to assess more responses to the same slot. The common end point of curves represents the cost and benefit of assessing all system responses. We can see that if we employ the RAGL assessor and apply some cut-off, the process can

be dramatically more efficient than the regular baseline based on alphabetical order. For example, in order to get 79 correct answers (76% of total), RAGL approach took human annotators only 5.5 hours, while the baseline approach took 13.4 hours.

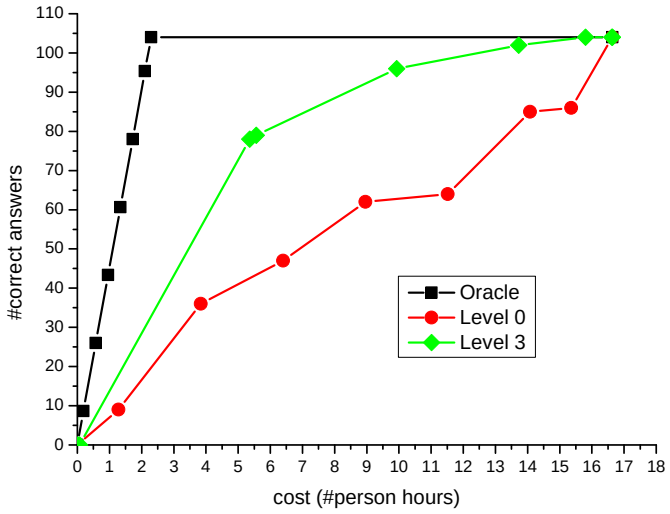

Figure 5.3: Human Assessment Method Comparison

5.3.2.8 Comparison with Alternative Methods

An alternative approach to validate answers is to use textual entailment techniques as in the RTE-KBP validation task (Bentivogli et al., 2011), which was partly inspired by CLEF Question Answering task (Penas et al., 2007). This task consists of determining whether a candidate answer (hypothesis "*H*") is supported in the associated source document (text "*T*") using entailment techniques. For the residence slots, we are considering in this chapter, they treat each context document as a "*T*", and apply pre-defined sentence templates such as "*[Query] lived in [Answer]*" to compose a "*H*" from system output. Entailment and reasoning methods from the TAC-RTE2010 systems are then applied

to validate whether "*H*" is true or false according to "*T*". These RTE-KBP systems are limited to individual *H-T* instances and optimized only on a subset of the pooled system responses. As a result, they aggressively filtered many correct answers and did not provide improvement on most slot filling systems (including the representative ones we used for our experiment). In contrast, our RAGL approach has the advantage of exploiting the generalized knowledge and feedback from assessors across all queries and systems.

5.3.3 Discussion

We have demonstrated that the comments from Level 3 provided significant improvement for TAC KBP Slot Filling task which require deep understanding of the contexts beyond surface texts. However, we also observed that some comments still require a system developer to fully understand and transfer the knowledge into detailed feature encoding by incorporating external resources. Therefore, the additional cost may vary based on the clarity of each comment and the availability of linguistic resources.

5.4 Conclusion

In a traditional supervised learning framework, a human annotator and a system are treated as isolated black-boxes to each other. We propose to better utilize the valuable knowledge from human annotators in the system development loop, by asking annotators to provide "rich annotations" for feature encoding. We investigated the trade-off between system performance and annotation cost, when adding rich annotations from various levels. We demonstrated the power and generality of this new framework on three

very different case studies. Experiments showed that the system trained from rich annotations can significantly save annotation cost in order to obtain the same performance as using basic annotations. It also outperformed some traditional validation methods, which, unlike ours, involved a great deal of feature engineering effort. The novelty of our approach lies in its declarative use of the privilege knowledge that human annotators utilize during annotation, which may address some typical errors that a system tends to make. Some of such feedback will be otherwise difficult to acquire for feature encoding (e.g., Comment 4 in slot filling). On the other hand, the simplicity of our approach lies in its low cost because it incorporates the bi-product of human annotation, namely their evidence, comments and explanations, instead of tedious instance-based human correction into the learning process. In this way the human annotator's knowledge is naturally transferred to the automatic system. Hence, rich-annotation based learning is amenable to implement but pertinent to a series of common errors identified, and thus fill in the knowledge gap between human annotators and feature engineers.

Chapter 6

User Profiling for Content Recommendation

6.1 Introduction

Nowadays the web plays an important role in the distribution of information from different sources to the users. The problem of *Information Overload* necessitates the use of the content recommendation techniques to help choose the best items matching users' interests. Thus users get better response to meet their needs without wasting much time filtering returned information. Most of these content recommendation systems face various difficulties while identifying and providing high-quality items to users. This is our motivation for conducting this analysis on content recommendation, and modeling user interests is a key and challenging component for personalized content recommendation.

Meeting user requirements involves a thorough understanding of their interests expressed explicitly through search queries or implicitly through content view and ad clicks.

Accurate understanding of current user interests and predicting their future interests are core tasks for user modeling, with a range of possible applications. For example, a query such as "*Michael Jordan*" could be interpreted differently depending on what entities they have previously queried or read in the pages, such as "*National Basketball Association (NBA)*" vs. "*University of California, Berkeley*". This contextual semantic knowledge extracted from queries and page content could be used to facilitate an accurate understanding of current and future user interests, which could be further employed to dynamically adapt search interfaces to support different tasks, such as reranking search results, classifying the query, suggesting alternative query formulations, or recommending news feed or ads. For example, *Yahoo News Stream* recommends items for the content feed or stream on Yahoo's homepage, shown in Figure 6.1.

Traditionally, user interests are modeled using different sources of profile information (e.g., explicit demographic or interest profiles, or implicit profiles based on previous queries, search result clicks, general browsing activity, or even richer desktop indices). And user preference is usually inferred from their activities (e.g., clicking on a hyperlink, viewing/saving/bookmarking a page), rather than trying to understand the semantics of the queries and the content of visited pages. The use of deep semantic knowledge allows us to furnish rich contextual information. For example, Wikipedia entities extracted either from the search queries or the contents of the webpages the user has visited make it possible to connect with knowledge bases where plenty of deep semantic knowledge about the entities exists. Given the query "*Michael Jordan*" (basketball player), we can infer the user's interest in basketball based on the fact that, *Michael Jordan* is a superstar basketball player in *NBA* from Wikipedia infobox. Hence, we can better model users'

Figure 6.1: Yahoo News Stream on Yahoo Homepage

interests from a deeper semantic aspect by investigating the information network based on the entities that the users are really interested in.

The use of semantic knowledge is not new. For example, Shen et al. (2005) have tried to infer users' interests from semantics by analyzing topics from queries and URL contents. But topics are too general to accurately capture the specific entities or areas that the users are interested in. For instance, a user may be interested in "*Michael Jordan*" (basketball player) and in basketball in general, but it does not mean all sports, such as swimming, golf, and horse racing, interest that user, so that the general topic *Sports* is not accurate and precise to summarize that user's interests. Hence, we try to understand users' interests from a semantic aspect by studying the entities and the underlying

relations/events related with the entities.

Besides Yahoo news stream recommendation, there are many other content recommendation products that recommend articles, videos, products, etc. based on users' search behaviors. Some of these are listed below:

- *Google Now Cards* based on personal web queries
- *Facebook* notifications based on "likes"
- *Amazon*'s product recommendation based on recent product queries and checkouts
- *Youtube*'s recommended videos based on viewing history

However, most of these limit themselves to heavily utilizing the surface-level features, such as user demographic profile, browsing activities, and counting of named entities. On the contrary, our proposed approach can understand more deeply about the contents by exploiting the entity knowledge encoded in knowledge bases. In this chapter, we focus on developing models capable of accurately predicting user interests for content recommendation, but we believe the proposed approach can also be used for a wide variety of applications, including supporting proactive changes to the interface to emphasize results of likely interest or to suggest contextually-relevant query alternatives, more traditional applications to ranking and filtering, news feed and appropriate ad recommendation, etc.

6.2 Task Background

In this section we present background and preliminaries of our work, including input data, semantic knowledge resources, and definition of user interest modeling.

Data The primary source of the data that we use is the logs collected from the Yahoo News Stream which include various raw and meta information. User click log contains the web pages/streams that users have visited. For each user, a user log sequence can be collected in an ascending timestamp order. We denote user click log sequence for user u as

$$L^u = \langle w_1^u, w_2^u, \ldots, w_t^u, \ldots, w_T^u \rangle,$$

where w_t^u is the web page that u visited at timestamp t. Related raw information including user id, geo information, type of web page, language, click/skip labels, timestamps, geological and demographical information, etc. Meta information involves topical category information (results of various Yahoo in-house classifiers) and Wikipedia (Wiki) entities extracted from the page content that the user viewed. Then, the user profiling problem is to learn a model that can construct a sparse matrix to represent user's interests over the features.

Semantic Knowledge Base For knowledge sources, we can use the whole Wikipedia corpus as well as more processed knowledge bases such as Yahoo Knowledge Graph, Freebase and NELL. In this chapter, we will use Yahoo Knowledge Graph to enrich the feature space. Yahoo! has its own internal knowledge graph, which is used to improve search results. This knowledge graph builds on both public data (e.g., Wikipedia and Freebase), as well as closed commercial sources for various domains (e.g., IMDb in movie domain). It uses wrappers for different sources and monitors evolving sources, such as

Algorithm 1 High-level Pipeline for User Interest Modeling and Prediction

1: Extract the queries, page titles and contents that the user has searched/visited/viewed from the user click logs.
2: Extract the named entities.
3: Link and resolve the identified named entities to the corresponding knowledge base entity entries.
4: Import and extract more related entities from the knowledge base, re-score the entity weights, and iterate to augment with more entities.
5: Infer future user interests utilizing all original and augmented entities.
6: Predict user activities on the contents based on user interest profiles.

Wikipedia, for constant updates. Yahoo Knowledge Graph contains roughly 3.5 million entities and 1.4 billion relations. Its schema, which is aligned with schema.org, comprises 250 types of entities and 800 types of relations (Blanco et al., 2013).

Software There is no Wiki entity information available for the news stream page. Thus we use *FastEL* (Blanco et al., 2015) for entity linking.

6.3 User Profile Modeling

In this section, we will discuss our approach to modeling user interests through entities. We propose a general, high-level pipeline for modeling user interests and predicting future user interests, which is described in Algorithm 1 and Figure 6.2.

Algorithm 2 Entity Augmentation

Input: User visited document D, Global Knowledge Graph G containing relation triples $\sigma = (E_a, p, E_b)$, where p is the relation predicate, number of iterations n, and maximum number of augmented entities m.

1: Generate initial entities $E = \{e_i\}$ from D
2: **repeat**
3: Augment entities using facts from G
4: Re-score interest weights of augmented entities
5: **until** converged or reach n iterations
6: **return** top m augmented entities from the list

6.3.1 Entity Augmentation

Based on the previous activities of the users, we can extract the named entities from the contents of visited news stream pages and link these entities to Wiki entities. But these entities themselves may not be sufficient to summarize users' interests and predict future user interests. Hence as the first step, we can exploit the global knowledge graph, such as *Yahoo Knowledge Graph* in this work, to augment the entities with the relational facts to include more entities that may interest the users. For example, if a user is interested in *Michael Jordan* (basketball player), there is a high probability that this user is also interested in *Chicago Bulls* (the basketball team where Michael Jordan served and won championships) and *NBA* (the basketball league). Even though these two entities may not co-occur with *Michael Jordan* in that visited news article, they may still intrigue user interests based on their close relationships with *Michael Jordan* in a much larger context. These related entities can be captured and linked through the massive number

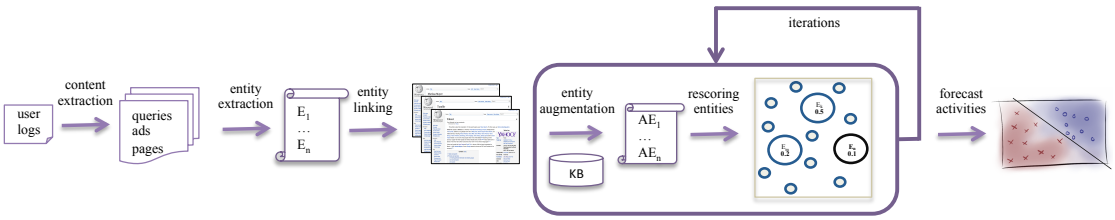

Figure 6.2: The High-level Pipeline of User Interests Modeling

of factual relations in the knowledge bases. Algorithm 2 shows the general process of entity augmentation.

After retrieving augmented entities, we can assign an decayed interest weight to each entity to indicate a smaller chance that users are interested in these augmented entities. For example, if the weight of the original entity E_a extracted from the content is 0.9 (linkage score from *FastEL*), we can apply entity augmentation based on the relation triples, (E_a, P_1, E_b) and (E_b, P_2, E_c), from Yahoo Knowledge Graph. Then we add entity E_b into the feature space to enrich the knowledge about user interests, where a decay weight is applied to decrease the weight of E_b to indicate our lesser confidence in this augmented entity. In this work, we set decay as 0.5, so the weight of E_b will be 0.45 ($= 0.9 * 0.5$). If we want to do one more iteration of entity augmentation, we can follow the same procedure to incorporate augmented entity E_c with 0.225 ($= 0.45 * 0.5$) weight into the feature space.

Besides this decay parameter, we have another parameter to control the maximum number of augmented entities for an entity after a certain number of iterations. For example, if we set the maximum number of augmented entities as 5, at most 5 top-ranked augmented entities (based on weights) from one entity will be added into the feature space.

6.3.2 User Profiling Framework

In this chapter, we will describe our user profiling framework for content recommendation. This model is based on Factorization Machines (FM) (Rendle, 2010), a state-of-the-art framework for latent factor models with rich features. FMs include and can mimic the most successful approaches in recommender systems including matrix factorization (Srebro et al., 2004), SVD++ (Koren, 2008) or PITF (Rendle and Schmidt-Thieme, 2010).

Although the independent learning model produces user interest features that are discriminative, the data sparsity problem can still affect the model performance. To solve this challenge, we propose to construct each user's profile by using data from all users. The main idea is to construct a latent space for all users, where users' interests can be distinguished while still learning from all available data.

Motivated by the success of matrix factorization methods in the domain of multi-task learning (Zhang et al., 2008) and collaborative filtering (Koren, 2008), we introduce a factorization-machine-based latent factor model to build profiles. Specifically, we decompose each user's profiles into two components, one is a common mapping from content item features to latent factors, which is shared by all users; and the other contains the latent factors of each user. Each latent factor represents the user's interest in that latent topic or feature. As the mapping from item features to latent factors are stable for all users, these feature factors can be considered as a bridge to propagate knowledge across users. Thus users who lack interaction data can benefit from the enriched information provided by interactions from other users. In addition, each user's latent factor membership distribution is specific, thereby allowing the model to reflect the users' personalized

interests.

In this factorization-machine-based framework, we use both article categorical features (results of various Yahoo in-house topical classifiers, e.g., SPORT and POLITICS) and entities (e.g., "*Michael Jordan*" and "*Barack Obama*") that are extracted from the content as the rich features, where we use *FastEL* (Blanco et al., 2015) to link entities to the Yahoo Knowledge Graph entries. And we also associate the features with the corresponding weights to indicate the confidence in feature value precision.

We briefly describe this model as follows. A Factorization Machine (FM) (Rendle, 2010) models all interactions between pairs of variables with the target, including nested ones, by using factorized interaction parameters[1]:

$$\hat{Y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} x_i x_j \tag{6.1}$$

where $\hat{w}_{i,j}$ are the factorized interaction parameters between pairs:

$$\hat{w}_{i,j} := \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{6.2}$$

and the model parameters Θ that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \tag{6.3}$$

A row $\mathbf{v}_i$ within $\mathbf{V}$ describes the i-th variable with k factors. $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the dimensionality of the factorization, which is set 20 in our implementation.

1. We restrict our discussion to 2-way FMs (d = 2)

A 2-way FM (dgree d = 2) captures all single and pairwise interactions between variables. That means w_0 is the global bias, w_i models the interaction of the i-th variable to the target and $\hat{w}_i, j$ models the factorized interaction of a pair of variables with the target. Note also that unlike other factorization models like matrix factorization or PARAFAC, FMs can work with any continuous input data $\mathbf{x}$.

Example of Content Recommendation Assume we have the user click log data of news stream. The system records which user $u \in U$ clicks a news item $i \in I$ with a response label R, {CLICK, SKIP}, where each item is represented using a feature vector $f \in F$ (i.e., article topic features and embedded entities). Let the users U, items I, features F, and labels R be:

$$U = \{Alice, Eileen, Alex, ...\}$$
$$I = \{item1, item2, item3, ...\}$$
$$F = \{\text{SPORT}, \text{MOVIE}, \textit{Barack Obama}(\text{OBAMA}), ...\}$$
$$R = \{\text{CLICK}, \text{SKIP}\}$$

Let the observered data S be:

$$
\begin{aligned}
S = \{&(u_1, Alice, item1, \{(\textsc{Movie}, 0.2), (\textsc{Obama}, 0.3), ...\}, \textsc{Click}),\\
&(u_1, Alice, item2, \{(\textsc{Movie}, 0.4), ...\}, \textsc{Skip}),\\
&(u_2, Eileen, item3, \{(\textsc{Obama}, 0.9), ...\}, \textsc{Skip}),\\
&(u_2, Eileen, item4, \{(\textsc{Sport}, 0.1), (\textsc{Obama}, 0.3), ...\}, \textsc{Click}),\\
&(u_3, Alex, item5, \{(\textsc{Sport}, 0.1), ...\}, \textsc{Click}),\\
&(u_3, Alex, item6, \{(\textsc{Movie}, 0.2), (\textsc{Obama}, 0.9), ...\}, \textsc{Skip})\}
\end{aligned}
$$

Figure 6.3 shows one example of how feature vectors can be created from S for this content recommendation task. Here, first there are $|U|$ binary indicator variables (blue) that represent the active user in each click log – there is always exactly one active user in each click log $(u, i, f, r) \in S$, e.g., user $Alice$ in the first one (e.g., $x^{(1)}_{Alice} = 1$). The next $|I|$ binary indicator variables (orange) hold the active item – again there is always exactly one active item (e.g., $x^{(1)}_{item1} = 1$). The feature vectors in Figure 6.3 contain indicator variables (purple) for all features that all items have. And finally the vector contains information of the click information, i.e., the active user clicks or skips that item.

6.4 Experiments

In this section, we present the experiments to evaluate the performance of our proposed user profiling approaches, by verifying the effectiveness of discriminative user profiles and the impact on news recommendation systems.

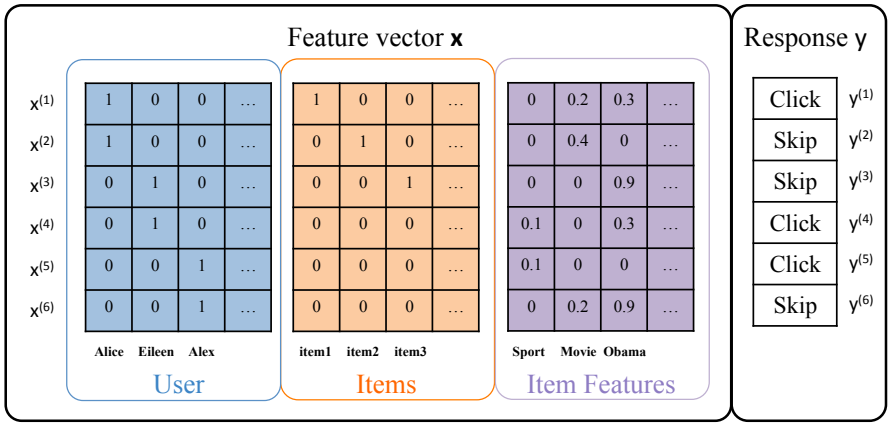

Figure 6.3: Example of Observed Data Representation in Content Recommendation

6.4.1 Experiment Setting

Our experiments are conducted based on a sample of the event log data from *Yahoo News Stream*. This data is collected over a period of four weeks (*09/21/2015 - 10/18/2015*), and it contains over 32.09 billion click events from around 56.65 million users and around 0.49 million news items. The number of original content features of the news items is around 125.06 billion.

In order to evaluate the quality of user profiles, we split the dataset into a training set and a test set based on the timestamps of events. We use the data from all weeks but the last one as training data and the data of last week as the test set. To be specific, the training dataset contains about 23.68 billion click events and 45.31 million users, and the test set contains about 8.42 billion click events and 22.08 million users. The training set is used to train the user profile models, i.e., each user's profile is built based on the data from the training dataset. In the test set, we generated a ranked list of new items for each user based on the user profiles generated from the training procedure, and evaluate

the performance of the ranking against the ground truth labels. The labels are the user positive and negative activities on the news items.

In the experiments, we use inner-product value between user profiles and item features to generate the ranking score for each user-item pair. Then we rank the items for each user based on these scores and check the positions of those clicked items, i.e., the items whose labels are positive. The basic idea is if the clicked items can be ranked higher than the negative ones, the model performs better. Three ranking evaluation metrics are used: *Mean Average Precision (MAP)*, *Mean Reciprocal Rank (MRR)*, and *Area under the Curve (AUC)*, which are defined as following:

$$MAP = \frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{k=1}^{n_i} P(k)}{n_i} \tag{6.4}$$

$$MRR = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{r_i^1} \tag{6.5}$$

$$AUC = \frac{1}{m}\sum_{i=1}^{m}\frac{(\sum_j r_i^j) - P_i(P_i+1)/2}{P_i * N_i} \tag{6.6}$$

where $P(k)$ is the precision at k, n_i is the number of items related to user u_i, r_i^1 is the rank of the first clicked item of user u_i, P_i is the number of clicked items, and N_i is the number of non-clicked items of user u_i.

6.4.2 Discussion

To conduct a systematic study, we have experimented with different entity augmentation iteration settings, and Table 6.1 lists the different settings and the corresponding number of features.

Figure 6.4 shows when the iteration number is set to 1, it achieves the best performance in terms of MAP and MRR evaluation metrics. It achieves more than 10% absolute and 193% relative improvement in MAP, as well as around 17% absolute and 191% relative improvement in MRR, compared to the baseline system (without entity augmentation). While, augmenting entities with two iterations performs the best using AUC metric, obtaining around 7% absolute and 12% relative improvement over the baseline. As we know, MAP averages the precision scores of a ranked list over all positions of relevant items, and MRR is the inverse position of the first relevant item, so both of them focus more on the ranking performance of the ranked items. Whereas, AUC characterizes the trade-off between true positives and false positives as a threshold parameter is varied. This indicates if the ranking performance is more concerned, augmenting entities with one iteration can gain the best performance; if the portion of correct items returned by the system is more important, then apply entity augmentation with two iterations work the best. If both ranking and coverage are the interests and needs, three iterations can achieve the most stable improvement in the overall performance. Figure 6.4 clearly demonstrates the effectiveness of the entity augmentation approach in this content recommendation task, as this technique does bring in more related entities from the knowledge bases to greatly enrich the feature space.

We think the reason that, the performance quickly drops regarding MAP and MRR after three iterations of entity augmentation, is, the entity augmentation may introduce some entities that are too far away from either the user's interests or the original semantic focus. For instance, assume the original entity is *Michael Jordan*, after one iteration of entity augmentation, we may add *Chicago Bulls* into the feature space. Then based

Iteration Number	Max Number	Feature Number	Incremental	Total Incremental
0	0	125,058,212,688	-	-
1	3	241,409,890,254	116,351,677,566	116,351,677,566
2	5	297,142,629,695	55,732,739,441	172,084,417,007
3	7	342,511,178,864	45,368,549,169	217,452,966,176
4	10	399,450,554,993	56,939,376,129	274,392,342,305
5	15	477,610,271,794	78,159,716,801	352,552,059,106

Table 6.1: Number of Features in Each Iteration Settings

on *Chicago Bulls*, we may add *NBA* entity, but the entity augmentation of *NBA* may introduce *TNT*, which is one of the top TV partners of *NBA*. But in fact, the users may not have any interest in this TV broadcast itself at all. Therefore, entity augmentation may further enrich the feature space through more iterations of augmentation, but more noisy information may also overweight the useful entities, which results in the decreased performance.

6.5 Conclusion

Online news reading has become very popular as the web provides access to news articles from millions of sources around the world. A key challenge of news websites is to help users find the articles that are interesting to read. In this chapter, we present our research on utilizing the semantic knowledge encoded in the named entities from *Knowledge Bases* to improve user profiling for large-scale content recommendation, which

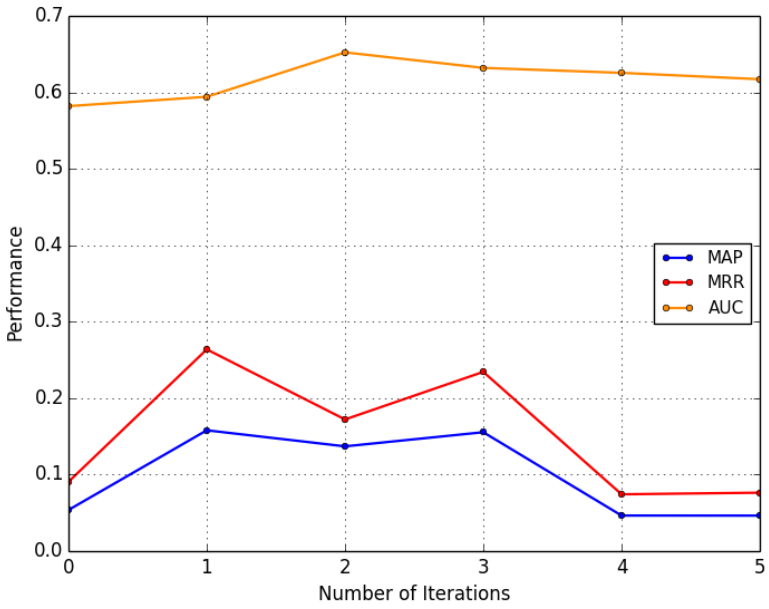

Figure 6.4: Performance Comparison with Different Numbers of Entity Augmentation Iterations

can help provide a richer feature space to tackle challenges in data sparsity and cold-start items. The proposed Factorization Machine (FM) based framework exploits both positive and negative implicit feedback from all users on content items to build user and item feature factors, and the user vector is used to represent user interests. Brand new items that have no interactions with the users can still be recommended by applying FM on users' interest vectors. In addition, we have also incorporated this FM system into a large-scale framework based on MapReduce, which provides the capability of handling massive amounts of users and content items. From the results on the Yahoo News Stream data, it demonstrates the effectiveness of our proposed approach that significantly improves the content recommendation performance.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The rise of Web 2.0 technology has provided a platform for user generated content through web blogs, forums, etc. This has lead to information overload on the web, and it has become an extremely difficult task for users to find the precise information they are looking for. Knowledge Bases (KBs) are data resources that encode world knowledge in machine-readable formats, and KBs like *Wikipedia* act as a rich source of information for various user needs and are important for a wide range of applications across all areas of science, technology, and culture like semantic search, question answering, text mining, virtual assistant, etc. Knowledge Base Population (KBP) aims at understanding this knowledge and extending KBs with more semantic information,

In order to build better large-scale knowledge bases efficiently and leverage KBs in other tasks, this dissertation focused on the following questions and introduced our research on each of these fields:

- **Question 1: "*How can we build knowledge bases?*"**

  We introduced a statistical language understanding approach to automatically construct personal (user-centric) knowledge graphs from conversational dialogs. Three key language understanding components are built: (1) *Personal Assertion Classification* identifies the user utterances that are relevant with personal facts, e.g., *"my mother's name is Rosa"*; (2) *Relation Detection* classifies the personal assertion utterance into one of the predefined relation classes, e.g., *"parents"*; and (3) *Slot Filling* labels the attributes or arguments of relations, e.g., *"name(parents):Rosa"*.

- **Question 2: "*How can we validate the correctness of information in knowledge bases?*"**

  Using the TAC Knowledge Base Population Slot Filling task as a case study, we proposed a confidence estimation model based on the Maximum Entropy framework, and the effectiveness of this model is demonstrated in both precision and the capability to improve the slot filling task through a weighted voting strategy.

- **Question 3: "*How can we build knowledge bases more efficiently?*"**

  We presented a new and general Rich Annotation Guided Learning framework to fill in the gap between an expert annotator and a feature engineer. This new framework can enrich features with the guidance of all levels of rich annotations from human annotators. We also evaluate the comparative efficacy, generality and scalability of this framework by conducting the case study on a slot filling task in the TAC KBP settings. Empirical studies demonstrate that with slightly more annotation time, we can significantly improve the performance for all tasks.

- **Question 4: "*How can we use these better knowledge bases to advance other tasks?*"**
  We showed the effectiveness of incorporating deep semantic knowledge encoded in the entities for modeling users' interests, by utilizing the abundance of knowledge populated in *Knowledge Bases*. Our approach can deeply understand the semantics behind the users' requests, compared to the traditional systems that usually infer users' interests from surface-level features derived from online activity logs and user demographic profiles, such as browsing history and geographic information.

7.2 Future Work

Although the current knowledge bases are impressive in their size, they still fall short of representing many kinds of knowledge that humans possess. Notably missing are representations of *common sense* facts (such as the fact that fire is hot, and fire can cook food), as well as *procedural* or *how-to* knowledge (such as how to drive a car or how to send an email) (Nickel et al., 2016). Representing, learning, and reasoning with these kinds of knowledge remains the next frontier for AI and machine learning. With these in mind, we are interested in exploring the following directions for better construction and utilization of knowledge bases:

Personal Knowledge Graph Population Since the current slot filling approach cannot handle utterances that involve two or more links, we plan to integrate an inference scheme into the framework to solve sophisticated relations invoked in the utterances. Given "*my wife was born in China*", for example, directly link place_of_birth:*China* to spouse_s node. We are also interested in exploring the personal preferences depicted in the utterances,

such as "*I am vegetarian*", since we believe this interested_in-style relation could enhance the performance of VPA to a great extent, like recommending appropriate restaurants in this case. In addition, we find that it is also very important to identify the negation expression and its scope within the utterances, which is crucial to determine whether a relation should be populated into the knowledge graph. We plan to boost our proposed framework towards these directions in the future.

Confidence Estimation for Knowledge Base Population In the future, further experiments are planned to investigate more elaborate models, explore more interesting feature sets, and study the contribution of each feature through a more detailed and thorough analysis. Furthermore, other information extraction case studies will be undertaken to validate the generality and reliability of this confidence estimation model.

Rich Annotation Guided Learning Remaining error analysis suggested that our future work should focus on mining deeper world knowledge and global reasoning from annotators. Moreover, we will investigate the effects of different rich annotations provided by multiple annotators and also apply on other problem settings. In the future, we are interested in extending this idea to improve other NLP applications and integrating it with human reasoning. The current setup mainly improved precision but we also plan to embrace the idea of revertible query in question answering literature (e.g., Prager et al., 2006) and relation graph traverse to enhance recall. Ultimately we intend to investigate automatic ways to prioritize comments and convert comments to features so that we can better simulate the role of teacher in human learning.

User Profiling For the future, we are interested in studying more sophisticated approaches to rescore the entity weights to indicate the importance. We also plan to explore more knowledge about the entities from the knowledge base and utilize the various attributes and categories accordingly. In addition, we intend to investigate the performance in predicting users' short-, mid-, and long-term interests and activities within a longer time frame, e.g., one year. We can also split the "*future data*" into two subsets, one contains only the *seen* entities for each user, while the other involves *unseen* entities, so that we can evaluate the capability of our framework for predicting future user interests on both *seen* and *unseen* entities. It will be interesting to investigate if our system can predict future user interests on the entities that have never been explored by the user before.

Bibliography

Adomavicius, Gediminas and Tuzhilin, Alexander (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: *IEEE Trans. Knowl. Data Eng.* 17.6, pp. 734–749. DOI: 10.1109/TKDE.2005.99.

Agarwal, Deepak, Chen, Bee-Chung, and Elango, Pradheep (2009). "Spatio-temporal models for estimating click-through rate". In: *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pp. 21–30. DOI: 10.1145/1526709.1526713.

Agarwal, Deepak, Chen, Bee-Chung, Elango, Pradheep, and Wang, Xuanhui (2012). "Personalized click shaping through Lagrangian duality for online recommendation". In: *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pp. 485–494. DOI: 10.1145/2348283.2348350.

Agarwal, Deepak, Chen, Bee-Chung, Gupta, Rupesh, Hartman, Joshua, He, Qi, Iyer, Anand, Kolar, Sumanth, Ma, Yiming, Shivaswamy, Pannagadatta, Singh, Ajit, and Zhang, Liang (2014). "Activity ranking in LinkedIn feed". In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 1603–1612. DOI: 10.1145/2623330.2623362.

Agichtein, Eugene (2006). "Confidence estimation methods for partially supervised relation extraction". In: *In SDM 2006*.

BIBLIOGRAPHY

Auer, Sören, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, Cyganiak, Richard, and Ives, Zachary G. (2007). "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52.

Bach, Nguyen, Huang, Fei, and Al-Onaizan, Yaser (2011). "Goodness: A Method for Measuring Machine Translation Confidence". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pp. 211–219.

Banko, Michele, Cafarella, Michael J, Soderland, Stephen, Broadhead, Matt, and Etzioni, Oren (2007). "Open information extraction from the web". In: *IJCAI*, pp. 2670–2676.

Belleau, François, Nolin, Marc-Alexandre, Tourigny, Nicole, Rigault, Philippe, and Morissette, Jean (2008). "Bio2RDF: Towards a mashup to build bioinformatics knowledge systems". In: *Journal of Biomedical Informatics* 41.5, pp. 706–716. DOI: 10.1016/j.jbi.2008.03.004.

Bentivogli, Luisa, Clark, Peter, Dagan, Ido, and Giampiccolo, Danilo (2011). "The Seventh PASCAL Recognizing Textual Entailment Challenge". In: *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

Berger, Adam L., Pietra, Stephen Della, and Pietra, Vincent J. Della (1996). "A Maximum Entropy Approach to Natural Language Processing". In: *Computational Linguistics* 22.1, pp. 39–71.

Biega, Joanna, Kuzey, Erdal, and Suchanek, Fabian M. (2013). "Inside YAGO2s: a transparent information extraction architecture". In: *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pp. 325–328.

BIBLIOGRAPHY

Blanco, Roi, Cambazoglu, Berkant Barla, Mika, Peter, and Torzec, Nicolas (2013). "Entity Recommendations in Web Search". In: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, pp. 33–48. DOI: 10.1007/978-3-642-41338-4_3.

Blanco, Roi, Ottaviano, Giuseppe, and Meij, Edgar (2015). "Fast and Space-Efficient Entity Linking for Queries". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. Shanghai, China: ACM, pp. 179–188. ISBN: 978-1-4503-3317-7. DOI: 10.1145/2684822.2685317.

Bodenreider, Olivier (2004). "The Unified Medical Language System (UMLS): integrating biomedical terminology". In: *Nucleic Acids Research* 32.Database-Issue, pp. 267–270. DOI: 10.1093/nar/gkh061.

Bollacker, Kurt D., Evans, Colin, Paritosh, Praveen, Sturge, Tim, and Taylor, Jamie (2008). "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pp. 1247–1250. DOI: 10.1145/1376616.1376746.

Bollacker, K., Cook, R., and Tufts, P. (2007). "Freebase: A Shared Database of Structured General Human Knowledge". In: *Proc. National Conference on Artificial Intelligence (Volume 2)*.

Boser, Bernhard E., Guyon, Isabelle M., and Vapnik, Vladimir N. (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, pp. 144–152.

Bril, Eric (1995). "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging". In: *Computational Linguistics (Volume 21, Number 1, March 1995)*.

BIBLIOGRAPHY

Broder, Andrei (2002). "A Taxonomy of Web Search". In: *SIGIR FORUM* 36.2, pp. 3–10.

Byrne, Lorna and Dunnion, John (2010). "UCD IIRG at TAC 2010". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

Carlson, Andrew, Betteridge, Justin, Kisiel, Bryan, Settles, Burr, Jr., Estevam R. Hruschka, and Mitchell, Tom M. (2010). "Toward an Architecture for Never-Ending Language Learning". In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*.

Castelli, Vittorio, Florian, Radu, and Han, Ding jung (2010). "Slot Filling through Statistical Processing and Inference Rules". In: *Proc. TAC 2010 Workshop*.

Castro, Rui, Kalish, Charles, Nowak, Robert, Qian, Ruichen, Rogers, Timothy, and Zhu, Xiaojin (2008). "Human Active Learning". In: *Proc. NIPS2008*.

Cheekula, Siva Kumar, Kapanipathi, Pavan, Doran, Derek, Jain, Prateek, and Sheth, Amit P. (2015). "Entity Recommendations Using Hierarchical Knowledge Bases". In: *Proceedings of the 4th Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data co-located with 12th Extended Semantic Web Conference (ESWC 2015), Portoroz, Slovenia, May 31, 2015*.

Chen, Li and Pu, Pearl (2004). *Survey of Preference Elicitation Methods*. Tech. rep. Ecole Politechnique Federale de Lausanne (EPFL), IC/2004/67.

Chen, Wenliang, Kazama, Jun'ichi, and Torisawa, Kentaro (2010a). "Bitext Dependency Parsing with Bilingual Subtree Constraints". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 21–29.

Chen, Zheng, Tamang, Suzanne, Lee, Adam, Li, Xiang, Lin, Wen-Pin, Snover, Matthew G., Artiles, Javier, Passantino, Marissa, and Ji, Heng (2010b). "CUNY-BLENDER

TAC-KBP2010 Entity Linking and Slot Filling System Description". In: *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*.

Chrupala, Grzegorz, Momtazi, Saeedeh, Wiegand, Michael, Kazalski, Stefan, Xu, Fang, Roth, Benjamin, Balahur, Alexandra, and Klakow, Dietrick (2010). "Saarland University Spoken Language Systems at the Slot Filling Task of TAC KBP 2010". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

Cortes, Corinna and Vapnik, Vladimir (1995). "Support-Vector Networks". In: *Machine Learning*, pp. 273–297.

Culotta, Aron and McCallum, Andrew (2004). "Confidence Estimation for Information Extraction". In: *Proceedings of HLT-NAACL 2004: Short Papers*. HLT-NAACL-Short '04. Boston, Massachusetts: Association for Computational Linguistics, pp. 109–112. ISBN: 1-932432-24-8.

Dauphin, Y., Tür, G., Hakkani-Tür, D., and Heck, L. (2014). "Zero-Shot Learning and Clustering for Semantic Utterance Classification". In: *Proceedings of the ICLR*. Banff, Canada.

Davis, Randall, Shrobe, Howard E., and Szolovits, Peter (1993). "What Is a Knowledge Representation?" In: *AI Magazine* 14.1, pp. 17–33.

Deng, L., Tür, G., He, X., and Hakkani-Tür, D. (2012). "Use of Kernel Deep Convex Networks and End-to-End Learning for Spoken Language Understanding". In: *In Prooceedings of the IEEE SLT Workshop*. Miami, FL.

Di Noia, Tommaso, Mirizzi, Roberto, Ostuni, Vito Claudio, Romito, Davide, and Zanker, Markus (2012). "Linked open data to support content-based recommender systems". In: *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, pp. 1–8. DOI: 10.1145/2362499.2362501.

BIBLIOGRAPHY

Dini, Luca, Di Tornaso, Vittorio, and Segond, Frederique (1998). "Error Driven Word Sense Disambiguation". In: *Proc. COLING1998*.

Dong, Xin, Gabrilovich, Evgeniy, Heitz, Geremy, Horn, Wilko, Lao, Ni, Murphy, Kevin, Strohmann, Thomas, Sun, Shaohua, and Zhang, Wei (2014). "Knowledge vault: a web-scale approach to probabilistic knowledge fusion". In: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 601–610. DOI: 10.1145/2623330.2623623.

Druck, Gregory, Mann, Gideon, and McCallum, Andrew (2008). "Learning from Labeled Features Using Generalized Expectation Criteria". In: *Proc. ACM SIGIR2008*.

El-Kahky, Ali, Liu, Derek, Sarikaya, Ruhi, Tür, Gökhan, Hakkani-Tür, Dilek Z., and Heck, Larry (2014). "Extending Domain Coverage of Language Understanding Systems via Intent Transfer Between Domains Using Knowledge Graphs and Search Query Click Logs". In: *Proceedings of the IEEE ICASSP*.

Etzioni, Oren, Fader, Anthony, Christensen, Janara, Soderland, Stephen, and Mausam (2011). "Open Information Extraction: The Second Generation". In: *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pp. 3–10.

Fader, Anthony, Soderland, Stephen, and Etzioni, Oren (2011). "Identifying Relations for Open Information Extraction". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1535–1545.

Fan, James, Ferrucci, David, Gondek, David, and Kalyanpur, Aditya (2010). "PRISMATIC: Inducing Knowledge from a Large Scale Lexicalized Relation Resource". In: *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology*

for Learning by Reading. FAM-LbR '10. Los Angeles, California: Association for Computational Linguistics, pp. 122–127.

Ferrucci, David A., Brown, Eric W., Chu-Carroll, Jennifer, Fan, James, Gondek, David, Kalyanpur, Aditya, Lally, Adam, Murdock, J. William, Nyberg, Eric, Prager, John M., Schlaefer, Nico, and Welty, Christopher A. (2010). "Building Watson: An Overview of the DeepQA Project". In: *AI Magazine* 31.3, pp. 59–79.

Gandrabur, Simona and Foster, George F. (2003). "Confidence estimation for translation prediction". In: *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pp. 95–102.

Gandrabur, Simona, Foster, George, and Lapalme, Guy (2006). "Confidence Estimation for NLP Applications". In: *ACM Trans. Speech Lang. Process.* 3.3, pp. 1–29. ISSN: 1550-4875. DOI: 10.1145/1177055.1177057.

Gao, Sanyuan, Cai, Yichao, Li, Si, Zhang, Zongyu, Guan, Jingyi, Li, Yan, Zhang, Hao, Xu, Weiran, and Guo, Jun (2010). "PRIS at TAC2010 KBP Track". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

Gorin, A. L., Riccardi, G., and Wright, J. H. (1997). "How May I Help You?" In: *Speech Communication* 23, pp. 113–127.

Grishman, Ralph, Westbrook, David, and Meyers, Adam (2005). "NYU's English ACE 2005 System Description". In: *Proc. ACE2005*.

Guha, R., McCool, R., and Miller, E. (2003). "Semantic Search". In: *Proceedings of the WWW*. Budapest, Hungary.

Gupta, N., Tür, G., and Hakkani-Tür, D. In:

BIBLIOGRAPHY

Haffner, P., Tür, G., and Wright, J. (2003). "Optimizing SVMs for Complex Call Classification". In: *Proceedings of the ICASSP*. Hong Kong.

Haghighi, Aria and Klein, Dan (2006). "Prototype-driven Learning for Sequence Models". In: *Proc. NAACL-HLT2006*.

Hakkani-Tür, Dilek Z., Heck, Larry, and Tür, Gökhan (2013). "Using a Knowledge Graph and Query Click Logs for Unsupervised Learning of Relation Detection". In: *the Proceedings of the ICASSP 2013*. IEEE.

Hakkani-Tür, Dilek Z., Celikyilmaz, Asli, Heck, Larry, Tür, Gökhan, and Zweig, Geoff (2014). "Probabilistic Enrichment of Knowledge Graph Entities for Relation Detection in Conversational Understanding". In: IEEE. ISCA - International Speech Communication Association.

He, Y. and Young, S. (2003). "A Data-Driven Spoken Language Understanding System". In: *Proceedings of the IEEE ASRU Workshop*, pp. 583–588.

Heck, Larry P. and Hakkani-Tür, Dilek (2012). "Exploiting the Semantic Web for Unsupervised Spoken Language Understanding". In: *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT 2012)*, pp. 228–233.

Heck, Larry, Hakkani-Tür, Dilek Z., and Tür, Gökhan (2013). "Leveraging Knowledge Graphs for Web-Scale Unsupervised Semantic Parsing". In: *Proceedings of Interspeech*. International Speech Communication Association.

Hoffart, Johannes, Suchanek, Fabian M., Berberich, Klaus, and Weikum, Gerhard (2013). "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia". In: *Artif. Intell.* 194, pp. 28–61. DOI: 10.1016/j.artint.2012.06.001.

Intxaurrondo, Ander, Lacalle, Oier Lopez de, and Agirre, Eneko (2010). "UBC at Slot Filling TAC-KBP2010". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

BIBLIOGRAPHY

Ji, Heng, Westbrook, David, and Grishman, Ralph (2005). "Using Semantic Relations to Refine Coreference Decisions". In: *Proc. HLT/EMNLP 05*.

Ji, Heng, Grishman, Ralph, Dang, Hoa Trang, Griffitt, Kira, and Ellis, Joe (2010). "Overview of the TAC 2010 Knowledge Base Population Track". In: *Proc. TAC 2010 Workshop*.

Ji, Heng and Grishman, Ralph (2011). "Knowledge Base Population: Successful Approaches and Challenges". In: *Proc. of ACL2011*, pp. 1148–1158.

Ji, Heng, Grishman, Ralph, and Dang, Hoa Trang (2011). "Overview of the TAC2011 Knowledge Base Population Track". In: *Proc. Text Analytics Conference (TAC2011)*.

Ji, Heng, Nothman, Joel, and Hachey, Ben (2014). "Overview of TAC-KBP2014 Entity Discovery and Linking Tasks". In: *Proc. Text Analytics Conference (TAC2014)*.

Ji, Heng, Nothman, Joel, Hachey, Ben, and Florian, Radu (2015). "Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking". In: *Proc. Text Analytics Conference (TAC2015)*.

Joachims, Thorsten (1999). "Making Large-scale Support Vector Machine Learning Practical". In: *Advances in Kernel Methods*. MIT Press, pp. 169–184.

Kapanipathi, Pavan, Jain, Prateek, Venkatramani, Chitra, and Sheth, Amit P. (2014). "User Interests Identification on Twitter Using a Hierarchical Knowledge Base". In: *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, pp. 99–113. DOI: 10.1007/978-3-319-07443-6_8.

Koren, Yehuda (2008). "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pp. 426–434. DOI: 10.1145/1401890.1401944.

BIBLIOGRAPHY

Koren, Yehuda, Bell, Robert M., and Volinsky, Chris (2009). "Matrix Factorization Techniques for Recommender Systems". In: *IEEE Computer* 42.8, pp. 30–37. DOI: 10.1109/MC.2009.263.

Krishnamurthy, Jayant and Mitchell, Tom (2012). "Weakly Supervised Training of Semantic Parsers". In: *Proceedings of EMNLP-CoNLL*, pp. 754–765.

Kuhn, R. and Mori, R. De (1995). "The application of semantic classification trees to natural language understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, pp. 449–460.

Lenat, Douglas B. and Feigenbaum, Edward A. (1991). "On the Thresholds of Knowledge". In: *Artif. Intell.* 47.1-3, pp. 185–250. DOI: 10.1016/0004-3702(91)90055-O.

Lenat, Douglas B. (1995). "CYC: A Large-Scale Investment in Knowledge Infrastructure". In: *Commun. ACM* 38.11, pp. 32–38. DOI: 10.1145/219717.219745.

Li, Xiang, Lin, Wen-Pin, and Ji, Heng (2011). "Comment-guided Learning: Bridging the Knowledge Gap between Expert Assessor and Feature Engineer". In: *Proc. International Conference on Advances in Information Mining and Management (IMMM2011)*.

Li, Xiang, Ji, Heng, Farooq, Faisal, Li, Hao, Lin, Wen pin, and Yu, Shipeng (2012). *Rich Annotation Guided Learning*.

Li, Xiang and Grishman, Ralph (2013). "Confidence Estimation for Knowledge Base Population". In: *Recent Advances in Natural Language Processing, RANLP 2013, 9-11 September, 2013, Hissar, Bulgaria*, pp. 396–401.

Li, Lei and Li, Tao (2013). "News recommendation via hypergraph learning: encapsulation of user behavior and news content". In: *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pp. 305–314. DOI: 10.1145/2433396.2433436.

BIBLIOGRAPHY

Liu, Jiahui, Dolan, Peter, and Pedersen, Elin Rønby (2010). "Personalized news recommendation based on click behavior". In: *Proceedings of the 2010 International Conference on Intelligent User Interfaces, February 7-10, 2010, Hong Kong, China*, pp. 31–40. DOI: 10.1145/1719970.1719976.

Louis, Annie and Nenkova, Ani (2009). "Performance Confidence Estimation for Automatic Summarization". In: *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pp. 541–548.

Lv, Yuanhua, Sun, Le, Zhang, Junlin, Nie, Jian-Yun, Chen, Wan, and Zhang, Wei (2006). "An Iterative Implicit Feedback Approach to Personalized Search". In: *Proc. Proc. ACL-COLING2006*.

Mahdisoltani, Farzaneh, Biega, Joanna, and Suchanek, Fabian M. (2015). "YAGO3: A Knowledge Base from Multilingual Wikipedias". In: *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*.

Mausam, Schmitz, Michael, Soderland, Stephen, Bart, Robert, and Etzioni, Oren (2012). "Open Language Learning for Information Extraction". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pp. 523–534.

McIlraith, S. A., Sun, T. C., and Zeng, H. (2001). "Semantic Web Services". In: *IEEE Intelligent Systems*, pp. 46–53.

Middleton, Stuart E., Shadbolt, Nigel, and Roure, David De (2004). "Ontological user profiling in recommender systems". In: *ACM Trans. Inf. Syst.* 22.1, pp. 54–88. DOI: 10.1145/963770.963773.

BIBLIOGRAPHY

Milidiu, Ruy L., dos Santos, Cicero Nogueira, and Duarte, Julio C. (2008). "Phrase Chunking Using Entropy Guided Transformation Learning". In: *Proc. ACL-HLT2008*.

Miller, George A. (1995). "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11, pp. 39–41. DOI: 10.1145/219717.219748.

Min, Bonan and Grishman, Ralph (2012). "Challenges in the Knowledge Base Population Slot Filling Task". In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey.

Min, Bonan, Li, Xiang, Grishman, Ralph, and Sun, Ang (2012). "New York University 2012 System for KBP Slot Filling". In: *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012*.

Minsky, M. (1974). "A Framework for Representing Knowledge". In: *MIT-AI Laboratory Memo 206*.

Momtchev, Vassil, Peychev, Deyan, Primov, Todor, and Georgiev, Georgi (2009). "Expanding the pathway and interaction knowledge in linked life data". In: *In Proc. of International Semantic Web Challenge*.

Nakashole, Ndapandula, Theobald, Martin, and Weikum, Gerhard (2011). "Scalable knowledge harvesting with high precision and high recall". In: *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pp. 227–236. DOI: 10.1145/1935826.1935869.

Nakashole, Ndapandula, Weikum, Gerhard, and Suchanek, Fabian M. (2012). "PATTY: A Taxonomy of Relational Patterns with Semantic Types". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pp. 1135–1145.

BIBLIOGRAPHY

Nemeskey, David, Recski, Gabor, Zseder, Attila, and Kornai, Andras (2010). "BUDAPESTACAD at TAC 2010". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

Nickel, Maximilian, Murphy, Kevin, Tresp, Volker, and Gabrilovich, Evgeniy (2016). "A Review of Relational Machine Learning for Knowledge Graphs". In: *Proceedings of the IEEE* 104.1, pp. 11–33. DOI: 10.1109/JPROC.2015.2483592.

Niu, Feng, Zhang, Ce, Re, Christopher, and Shavlik, Jude W. (2012a). "DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference". In: *Proceedings of the Second International Workshop on Searching and Integrating New Web Data Sources, Istanbul, Turkey, August 31, 2012*, pp. 25–28.

Niu, Feng, Zhang, Ce, Ré, Christopher, and Shavlik, Jude W. (2012b). "Elementary: Large-Scale Knowledge-Base Construction via Machine Learning and Statistical Inference". In: *Int. J. Semantic Web Inf. Syst.* 8.3, pp. 42–73. DOI: 10.4018/jswis.2012070103.

Passant, Alexandre (2010). "dbrec - Music Recommendations Using DBpedia". In: *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part II*, pp. 209–224. DOI: 10.1007/978-3-642-17749-1_14.

Penas, Anselmo, Rodrigo, Alvaro, Sama, V., and Verdejo, Felicia (2007). "Testing the Reasoning for Question Answering Validation". In: *Journal of Logic and Computation*.

Pieraccini, R., Tzoukermann, E., Gorelov, Z., Gauvain, J.-L., Levin, E., Lee, C.-H., and Wilpon, J. G. (1992). "A Speech Understanding System Based on Statistical Representation of Semantics". In: *Proceedings of the ICASSP*.

Pieraccini, Roberto and Levin, Esther (1995). "A Learning Approach to Natural Language Understanding". In: *In Speech Recognition and Coding, New Advances and Trends, NATO ASI Series*. Springer Verlag, pp. 139–155.

Prager, J., Duboue, P., and Chu-Carrol, J. (2006). "Improving QA Accuracy by Question Inversion". In: *Proc. ACL-COLING2006*.

Price, P. J. (1990). "Evaluation of spoken language systems: The ATIS domain". In: *Proceedings of the DARPA Workshop on Speech and Natural Language*. Hidden Valley, PA.

Raghavan, H., Madani, O., and Jones, R. (2006). "Active learning with Feedback on both Features and Instances". In: *Journal of Machine Learning Research*.

Raymond, Christian and Riccardi, Giuseppe (2007). *Generative and Discriminative Algorithms for Spoken Language Understanding*.

Rendle, Steffen (2010). "Factorization Machines". In: *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pp. 995–1000. DOI: 10.1109/ICDM.2010.127.

Rendle, Steffen and Schmidt-Thieme, Lars (2010). "Pairwise interaction tensor factorization for personalized tag recommendation". In: *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pp. 81–90. DOI: 10.1145/1718487.1718498.

Richardson, Matt and Domingos, Pedro (2006). "Markov Logic Networks". In: *Machine Learning*.

Ruttenberg, Alan, Rees, Jonathan, Samwald, Matthias, and Marshall, M. Scott (2009). "Life sciences on the Semantic Web: the Neurocommons and beyond". In: *Briefings in Bioinformatics* 10.2, pp. 193–204. DOI: 10.1093/bib/bbp004.

Sarikaya, R., Hinton, G. E., and Ramabhadran, B. (2011). "Deep Belief Nets for Natural Language Call-Routing". In: *Proceedings of the ICASSP*.

Sarwar, Badrul M., Karypis, George, Konstan, Joseph A., and Riedl, John (2001). "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the Tenth*

International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001, pp. 285–295. DOI: 10.1145/371920.372071.

Schapire, R. E. and Singer, Y. (2000). "Boostexter: A Boosting-based System for Text Categorization". In: *Machine Learning* 39.2/3, pp. 135–168.

Scheffer, Tobias, Decomain, Christian, and Wrobel, Stefan (2001). "Active Hidden Markov Models for Information Extraction". In: *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001, Cascais, Portugal, September 13-15, 2001, Proceedings*, pp. 309–318. DOI: 10.1007/3-540-44816-0_31.

Schmachtenberg, Max, Bizer, Christian, Jentzsch, Anja, and Cyganiak, Richard (2014). "Linking Open Data Cloud Diagram". In: *http://lod-cloud.net/*.

Seneff, S. (1992). "TINA: A Natural Language System for Spoken Language Applications". In: *Computational Linguistics* 18.1, pp. 61–86.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). "The Semantic Web Revisited". In: *IEEE Intelligent Systems*, pp. 96–101.

Shen, Xuehua, Dumais, Susan T., and Horvitz, Eric (2005). "Analysis of topic dynamics in web search". In: *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005 - Special interest tracks and posters*, pp. 1102–1103. DOI: 10.1145/1062745.1062889.

Sieg, Ahu, Mobasher, Bamshad, and Burke, Robin D. (2007). "Web search personalization with ontological user profiles". In: *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pp. 525–534. DOI: 10.1145/1321440.1321515.

Singhal, A. (2012). "Introducing the knowledge graph: things, not strings". In: *[Online Blog] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html*.

BIBLIOGRAPHY

Sowa, J. F. (2008). "Semantic Networks". In: *Encyclopedia OF Cognitive Science*.

Srebro, Nathan, Rennie, Jason D. M., and Jaakkola, Tommi S. (2004). "Maximum-Margin Matrix Factorization". In: *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pp. 1329–1336.

Steedman, Mark (1996). *Surface Structure and Interpretation*. The MIT Press.

Suchanek, Fabian M., Kasneci, Gjergji, and Weikum, Gerhard (2007). "Yago: A Core of Semantic Knowledge". In: *The 16th International World Wide Web conference*.

Sugiyama, Kazunari, Hatano, Kenji, and Yoshikawa, Masatoshi (2004). "Adaptive web search based on user profile constructed without any effort from users". In: *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pp. 675–684. DOI: 10.1145/988672.988764.

Suh, Bongwon, Convertino, Gregorio, Chi, Ed H., and Pirolli, Peter (2009). "The singularity is not near: slowing growth of Wikipedia". In: *Proceedings of the 2009 International Symposium on Wikis, 2009, Orlando, Florida, USA, October 25-27, 2009*. DOI: 10.1145/1641309.1641322.

Surdeanu, Mihai, McClosky, David, Tibshirani, Julie, Bauer, John, Chang, Angel X., Spitkovsky, Valentin I., and Manning, Christopher D. (2010). "A Simple Distant Supervision Approach for the TAC-KBP Slot Filling Task". In: *Proceedings of Text Analytics Conference (TAC) 2010*.

Surdeanu, Mihai and Ji, Heng (2014). "Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation". In: *Proc. Text Analytics Conference (TAC2014)*.

Tamang, Suzanne and Ji, Heng (2011). "Adding Smarter Systems Instead of Human Annotators: Re-ranking for System Combination". In: *Proceedings of the 1st International*

Workshop on Search and Mining Entity-relationship Data. SMER '11. Glasgow, Scotland, UK: ACM, pp. 3–8. ISBN: 978-1-4503-0957-8. DOI: 10.1145/2064988.2064992.

Thompson, Cynthia A., Califf, Mary Elaine, and Mooney, Raymond J. (1999). "Active Learning for Natural Language Parsing and Information Extraction". In: *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 406–414.

Tür, G., Hakkani-Tür, D., and Heck, L. (2010). "What is Left to be Understood in ATIS?" In: *Proceedings of the IEEE SLT Workshop*. Berkeley, CA.

Tür, Gökhan and DeMori, Renato (2011). *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York, NY: John Wiley and Sons.

Tür, G., Jeong, M., Wang, Y.-Y., Hakkani-Tür, D., and Heck, L. (2012). "Exploiting the Semantic Web for Unsupervised Natural Language Semantic Parsing". In: *In Prooceedings of the Interspeech*. Portland, OR.

Tür, Gökhan, Wang, Ye-Yi, and Hakkani-Tür, Dilek Z. (2014). "Understanding Spoken Language". In: *Computing Handbook, Third Edition: Computer Science and Software Engineering*, 41: 1–17.

Tyler, Sarah K. and Teevan, Jaime (2010). "Large Scale Query Log Analysis of Re-Finding". In: *Proc. WSDM2010*.

Vapnik, Vladimir (2009). "Learning with Teacher: Learning Using Hidden Information". In: *Proc. International Joint Conference on Neural Networks 2009*.

Vrandecic, Denny and Krötzsch, Markus (2014). "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57.10, pp. 78–85. DOI: 10.1145/2629489.

Wang, Y.-Y. and Acero, A. (2006). "Discriminative models for spoken language understanding". In: *Proceedings of the ICSLP*. Pittsburgh, PA.

Wang, Lu, Heck, Larry, and Hakkani-Tür, Dilek Z. (2014). "Leveraging Semantic Web Search and Browse Sessions for Multi-Turn Spoken Dialog Systems". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Conference*.

Ward, W. and S.Issar (1994). "Recent Improvements in the CMU Spoken Language Understanding System". In: *Proceedings of the ARPA HLT Workshop*.

Webb, Geoffrey I., Pazzani, Michael J., and Billsus, Daniel (2001). "Machine Learning for User Modeling". In: *User Model. User-Adapt. Interact.* 11.1-2, pp. 19–29. DOI: 10.1023/A:1011117102175.

West, Robert, Gabrilovich, Evgeniy, Murphy, Kevin, Sun, Shaohua, Gupta, Rahul, and Lin, Dekang (2014). "Knowledge base completion via search-based question answering". In: *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pp. 515–526. DOI: 10.1145/2566486.2568032.

White, Christopher, Droppo, Jasha, Acero, Alex, and Odell, Julian (2007). "Maximum Entropy Confidence Estimation for Speech Recognition". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, pp. 809–812. DOI: 10.1109/ICASSP.2007.367036.

Williams, Ken, Dozier, Christopher, and McCulloh, Andrew (2004). "Learning Transformation Rules for Semantic Role Labeling". In: *Proc. CoNLL-2004*.

Wu, Wentao, Li, Hongsong, Wang, Haixun, and Zhu, Kenny Qili (2012). "Probase: a probabilistic taxonomy for text understanding". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pp. 481–492. DOI: 10.1145/2213836.2213891.

Xu, P. and Sarikaya, R. (2013). "Convolutional Neural Network Based Triangular CRF for Joint Intent Detection and Slot Filling". In: *Proceedings of the IEEE ASRU*. Olomouc, Czech Republic.

Yao, K., Peng, B., Zweig, G., Yu, D., Li, X., and Gao, F. (2014). "Recurrent Conditional Random Field for Language Understanding". In: *Proceedings of the IEEE ICASSP*. Florence, Italy.

Yu, Jingtao, Mujgond, Omkar, and Gaizauskas, Rob (2010). "The University of Sheffield System at TAC KBP 2010." In: *Proceedings of Text Analytics Conference (TAC)*.

Yu, Shipeng, Farooq, Faisal, Krishnapuram, Balaji, and Rao, Bharat (2011). "Leveraging Rich Annotations to Improve Learning of Medical Concepts from Clinical Free Text". In: *Proc. ICML 2011 Workshop on Learning from Unstructured Clinical Text*.

Yu, Xiao, Ma, Hao, Hsu, Bo-June Paul, and Han, Jiawei (2014b). "On building entity recommender systems using user click log and freebase knowledge". In: *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pp. 263–272. DOI: 10.1145/2556195.2556233.

Yu, Dian, Huang, Hongzhao, Cassidy, Taylor, Ji, Heng, Wang, Chi, Zhi, Shi, Han, Jiawei, Voss, Clare R., and Magdon-Ismail, Malik (2014a). "The Wisdom of Minority: Unsupervised Slot Filling Validation based on Multi-dimensional Truth-Finding". In: *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pp. 1567–1578.

Zaidan, O. F., Eisner, J., and Piatko, C. D. (2007). "Using "Annotator Rationales" to Improve Machine Learning for Text Categorization". In: *Proc. NAACL-HLT2007*.

Zaidan, O. F. and Eisner, J. (2008). "Modeling Annotators: A Generative Approach to Learning from Annotator Rationales". In: *Proc. EMNLP2008*.

Zhang, Jian, Ghahramani, Zoubin, and Yang, Yiming (2008). "Flexible latent variable models for multi-task learning". In: *Machine Learning* 73.3, pp. 221–242. DOI: 10.1007/s10994-008-5050-1.

Zhong, Erheng, Liu, Nathan, Shi, Yue, and Rajan, Suju (2015). "Building Discriminative User Profiles for Large-scale Content Recommendation". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: ACM, pp. 2277–2286. ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2788610.

Zitouni, I., Kuo, H.-K. J., and Lee, C.-H. (2003). "Boosting and Combincation of Classifiers for Natural Language Call Routing Systems". In: *Speech Communication* 41.4, pp. 647–661.

Zukerman, Ingrid and Albrecht, David W. (2001). "Predictive Statistical Models for User Modeling". In: *User Model. User-Adapt. Interact.* 11.1-2, pp. 5–18. DOI: 10.1023/A:1011175525451.

"Slot Filler Validation/Ensembling at TAC 2015 Task Guidelines" (2015). In: *Proc. Text Analytics Conference (TAC2015)*.

"TAC KBP 2015 Slot Descriptions" (2015). In: *Proc. Text Analytics Conference (TAC2015)*.