國立臺灣大電機資訊學院資訊工程研究所

博士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

概念表徵及其應用

Concept Representation and Its Application

游基鑫

Chi-Hsin Yu

指導教授：陳信希　博士

Advisor: Hsin-Hsi Chen, Ph.D.

中華民國 102 年 8 月

August 2013

誌　謝

論文終於完成，要衷心感謝很多人。最要感謝的人是我的家人，在人生的這個階段回學校念書，家人們給了我最多的體諒及支持，沒有這些，研究將無法進行。

也要感謝指導老師陳信希教授，在這七年的時間中對我的指導、鼓勵、包容與各種支持，有他的耐心引導及支持，這論文才能順利完成，謹在此表達深摯的敬意與感激。另外，也要感謝眾多學弟妹們的陪伴、討論及各種幫助。

謹將這論文獻給他們。

Abstract

In this dissertation, we propose a concept definition in language, derive a concept representation scheme based on this definition, and apply this framework in two applications: commonsense knowledge classification and word sense disambiguation. In addition, we assert two important assumptions for building concept representation using knowledge extraction: does commonsense knowledge appear in texts and is a small part of the Web sufficient for supporting important NLP tasks. Last, we introduce processed ClueWeb09 datasets. We hope the produced datasets can boost NLP research.

We give a definition of concept that meets three criteria: having native origin in computational perspective, having no undefined terms in the definition, and having build-in nature for deep analysis by human and by intelligent system itself to understand internal structures of an intelligent system. We define concept a continuation, which is a temporary state in the concept computation process. This temporary state is interpreted within the context of the evolutionary language game. Based on this definition, we define concept representation to have two parts: static and dynamic parts. We investigate some theoretical aspects using theories in machine learning literatures.

In the application of commonsense knowledge classification, we adopt vector space model to build representation and interpret this machine learning process in our framework. In WSD, we further apply our framework to develop two new concepts for solving WSD: context appropriateness and concept fitness. We use these two new concepts to build many new algorithms to solve WSD problem.

For using knowledge extraction to build concept representation in the future, we verify two important perspectives: content of knowledge and size of knowledge sources. We find that commonsense knowledge are recorded in texts and assert that the web is a good source to

extract human knowledge. We use word ordering error task to indirectly assert that a small part of the web, such as ClueWeb09 dataset, can support NLP applications to produce comparable results to that of larger datasets, such as Google Web 5-gram dataset. These two assertions give us confidence to extract knowledge from a smaller dataset to build concept representation.

Lastly, we preprocess English and Chinese web pages in ClueWeb09 and produce many resources for researchers, including (1) POS-tagged, phrase-chunked, and partly parsed English dataset, (2) segmented, POS-tagged, and discourse markers identified Chinese dataset, and (3) NTU Chinese POS-5gram dataset.

摘要

在此論文中，我們為概念進行了定義，並基於此定義，提出了為系統建構概念表徵的架構，及將此架構，套用在常識知識分類以及文字歧義消解這兩應用中。除此之外，我們還驗證了兩個跟知識抽取有關的假設，這分別是常識知識是否出現在文字中，以及小規模網路文件集是否足以支援重要的自然語言處理工作。最後，我們介紹了 ClueWeb09 這一網絡規模資料集的一些前處理結果，希望能提供給其他研究者更好用的資源。

我們給出的概念定義符合三個標準：本質上具有可計算性、沒有無定義的組成、有內建的特質可被人或機器自身進行分析。我們將概念定義成一種延續（continuation），這種延續可看成是一種概念運算過程的暫存態，此暫存態則放在進化語言博弈（evolutionary language game）的架構下來詮釋。在此定義基礎上，我們將概念表徵分為靜態跟動態兩方面，並使用機器學習理論來對系統的許多面向進行了理論的探討。

將概念表徵應用在常識知識分類時，我們用向量空間模型來建構表徵，並展示如何用我們的概念定義，來詮釋一般的機器學習處理過程。而在文字歧義消解這一應用中，我們更進一步運用了我們發展出的概念，為文字歧義消解引入了脈絡適切性（context appropriateness）及概念適切性（concept fitness）此兩面向，並用此來建構嶄新的文字歧義消解演算法。

為了未來使用自動知識抽取的架構為機器建構概念，我們驗證了知識內容及大小這兩基本問題。為了確認文件是好的知識內容來源，我們發現甚至連常識知識都會出現在文件中。另外，我們利用文字語序錯誤這一問題，間接驗證了雖然 ClueWeb09 的規模只是網路網頁的一小部份，它的規模已可產生跟 Google Web 5-gram 同樣的實驗結果，能很好的支援重要的自然語言處理工作。

最後，我們對 ClueWeb09 這一網絡規模資料集進行了前處理，並產生了許多有用的資源可提供給研究者，這些資源包括（1）完成詞性標記、詞組切分及語句剖析的英文語料庫、（2）完成斷詞、詞性標記及語篇標記詞標記的中文語料庫、（3）中文詞性

n-gram資料集（NTU Chinese POS 5-gram）。

Content

Illustrations

Tables

Equations

Chapter 1. Introduction

Concept and its representations have been studied for a long time in many disciplines. Scholars of different fields such as philosophy, psychology, cognitive science, artificial intelligence and natural language processing try hard to define what is concept, to trace the history of a specific concept, to model concepts in human mind, to discover the subtleness between similar concepts, to organize concepts in ontology, to search words that refer to same concept, to study how to draw concept from materials and to represent concept in machine-readable resources. Different disciplines have different focuses when they study concept-related topics. In artificial intelligence and natural language processing, researchers are interest in how to define concept and how to represent concepts in machine-readable format in the hope of supporting the task in hand.

In this dissertation, we are interest in drawing a computational framework to define concept. The computational framework we used is based on continuation which is a concept used in programming language. Based on the framework, we define the concept representation scheme and apply the scheme to many applications to explore the usefulness of the computation framework and the representation scheme.

1.1 Motivation

In the pursuit of building human-like intelligent machines, defining concept and building concept representation are very important. Although concept has been studied for thousands years and scholars of different disciplines have proposed different definitions of concept for their uses, there are fewer definitions that explores computational perspective of concept. In addition, most concept definitions are always end up with some concepts that are needed to be further defined. Although using undefined terms to define something is possible for human mind, these kinds of definitions introduce difficulties when we want to use these definitions to build an intelligent machine.

For example, philosophers usually define concept in terms of the roles that concept plays in their problems of interests or the world they believe. If they believe the world has a pre -existing structure, they may prefer to define concept in terms of ontology or may believe that concept has a predefined structure which reflects the world's inherent properties and most fundamental structures. Plato’s theory of Forms holds this belief of relation between concept and the world in two thousand years ago. In this approach, philosophers give different structures for different concepts in terms of different terminologies, such as attributes, roles, categories, mental representations, abstract objects, and abilities. These terminologies are usually regarded as well-known or self-defined objects. When computer scientists adopt these definitions in their tasks such as machine reading, information extraction, and word sense disambiguation, these undefined objects are simply translated to features of a feature matrix in machine learning fields. For example, the features maybe the co-occurrence words in distributional representation approach (Harris, 1954). These words are undefined. More precisely, researchers interpret these words by themselves. In such cases, the whole system is a mathematical model and this model a black box for researchers. Researchers may manipulate different mathematical models or different model parameters to see how the models response to the operations, but researchers have little chance and face great difficulty to analyze the internal structures of these words. They do not know how the internal structures response to a specific model in a specific configuration of model parameters. If researchers use engineering perspectives to deal with the task, this is not a big problem because they have a workable system to solve their tasks in hand. If researchers want to build a real intelligent system, the system must interpret these words by itself and it must has knowledge on what it is doing. This problem highlight the need to eliminate the use of undefined terms and the use of human-interpret concepts.

We will explain these issues in length in later chapters. In summary, when concept definition ends up in undefined terms or human interpreted terms, it restricts the ability for researchers to conduct a deep analysis on the behaviors and internal structures of intelligent systems. It also restricts the ability for an intelligent system to interpret its behaviors by itself. Therefore, in this dissertation, we want to give a definition of concept that meets the criteria below:

(1) has native origin in computational perspective,

(2) has no undefined terms in the definitions,

(3) and has the build-in nature in deep analysis for human and for intelligent system itself to understand internal structures of an intelligent system.

We hope that with this concept definition, we can shed light on building real intelligent systems and boost the understanding of model building on solving a specific research task in engineering perspectives.

1.2 Overview of this Dissertation

In this dissertation, we define concept as continuation, define a representation scheme based on this definition, and adopt the concept representation in architecture of automatic knowledge extraction.

In chapter 2, we describe the concept definition and investigate some computational aspects of this definition. We elaborate advantages of new definition by comparing it with some well-known definitions.

In chapter 3, we draw a concept representation scheme from our definition of concept. The concept representation scheme is a simple instantiation of our definition for the implementation purpose. When using a simple instantiation, we can focus our attention on the definition and avoid describing a complicated system.

In chapter 4, we use the concept representation scheme to interpret a classical machine learning procedure. We explain that our representation scheme is capable of subsuming machine learning process and the is more general and useful for human to understand system. We use commonsense knowledge classification to demonstrate our claim.

In chapter 5, we use the concept representation scheme to consider the relation between concept and its context. We identify concept fitness and context appropriateness for word sense disambiguation (WSD) problems. Using these two perspectives, we develop a novel ranking algorithm for WSD. We conduct experiments and report results in this chapter.

In chapter 6, we describe resources processing procedures and results.

In the last chapter, we summarize our dissertation and picture some future work.

Chapter 2. Concept as Continuation

Our concept definition is originated from the context of language study. We propose a computational architecture, and define concept as continuation in this architecture. After that, we investigate some important issues related to this definition. We organize materials in the order below.

(1) We introduce concept theory (Hjørland, 2009) first, which concisely describes how scholars study the theories of concepts.

(2) We introduce Shannon’s communication system, evolutionary language game (Trapa & Nowak, 2000) and the Chinese Room problem, which inspire us to derive our concept definition.

(3) We describe our concept definition, which defines concept as continuation.

(4) We investigate the implication of this definition in theoretical perspectives.

(5) We compare our concept definition with other definitions.

(6) We describe some considerations and variations on implementing the proposed definition.

2.1 Concept Theory

Although concepts have been studied thousand years, people still do not have a generally accepted agreement on what concepts are. Researchers often credit Plato (424 – 348 BC) and Aristotle (384 – 322 BC) being the earliest scholars to study concept formally. However, their ideas of concepts provide an important reference on the study of concepts but not a generally accepted consensus. Scholars propose many theories of concepts and discuss many views of concepts, and that does not result in consensus but in enlarging the border of our understanding of concepts. Hjørland (2009) systematically survey the theories of concepts and classify these theories into four families. His classification is based on epistemological viewpoint, and he uses theories of knowledge to classify theories of concepts.

We describe these four families here because it can give us a reference framework when we want to understand our proposed concept definition. These four families of theories of concepts are empiricism, rationalism, historicism, and pragmatism.

Empiricism argues that knowledge is draw from observations. These observations are given by settings and are not contextual or theory-dependent. When applying empiricism to semantic, empiricism argues that meanings are defined based on observable features. When applying empiricism to concepts, empiricism argues that human’s sensations derive the concept. In computer science, empiricism argues that neural networks can be seen as modeling concept in empiricism.

Rationalism argues that knowledge is based on predefined structures or rules, which can be logics, principles, or ontology. Plato’s theory of Forms is in this family. When applying rationalism to concepts, rationalism argues that concepts are prior to human’s sensations. Hjørland (2009) regard Formal Concept Analysis (FCA) (Priss, 2004, 2006) as a prominent mathematical formation of rationalist concept theory. The FCA uses features to define concept, and these features are regarded as simple and well-defined for the human.

Historicism argues that knowledge has its social context and is time-variant. It argues that observations are theory-dependent and always be influenced by cultures, environments, or contexts. When applying historicism to concepts, historicism argues that concepts are always evolving. The concepts will change when the cognitive mechanisms are functioning. To understand a concept, historicism concerns about discovering the effective assumptions behind the concept and tracing the changes of these assumptions.

Pragmatism argues that knowledge is based on “the analysis of goals, purposes, values, and consequences“. That is to say, knowledge is always based on some specific aspects of reality. Pragmatism also argues that observations are theory-dependent, but it argues that knowledge cannot be neutral because it is derived for some purposes. When applying pragmatism to concepts, pragmatism argues that concepts are faceted. A concept describes reality in some aspects and ignores other aspects of the reality for their purposes.

Hjørland (2009) gives three examples to illustrate the difference of concept theories, but we summarize three factors to make a more concise distinctions of these concept theories. The factors are structure-depend, time-variant, and faceted. We show the summary in Table 1.

Family Factor	Structure-depend	Time-variant	Faceted
Empiricism	No	No	No
Rationalism	Yes	No	No
Historicism	Yes	Yes	No
Pragmatism	Yes	Yes	Yes

Table 1. The differences of families of theories of concepts

In Table 1, we replace theory-dependent with structure-depend because the theory sometimes refers to ontology or reflects a specific world structure. In those cases, structure is more precise for describing the idea. In historicism and pragmatism, the structure is from the context or the purpose, which are different from rationalism. The time-variant factor in historicism and pragmatism contains the cases that the perceived concepts may affect the process of following concept perception. In terminology of machine learning, system feeds outputs to its inputs, which may result in recurrent neural network architecture. We do not put any implication between time-variant factor and self-feed architecture here.

With the theories of concepts in mind, we can use it to explain our concept definition. Before defining concept, we put some words on the distinctions between concept definition and theories of concepts.

When we use the term *definition*, it means we refer one term to something. For example, a *concept definition* is to refer *concept* to *something*. Abstractly speaking, a *definition* connects object to other objects and use criteria to rate the goodness of this connection. In the philosophy of science, this means we use *something* to explain *concept* in order to reach a good understanding of *concept*. The criteria of judging the goodness of explanation is not easy to formulate. According to Friedman (1974), the judgment is the problem of scientific explanation:

> "The central problem for the theory of scientific explanation comes down to this: what is the relation between phenomena in virtue of which one phenomenon can constitute an explanation of another, and what is it about this relation that gives understanding of the explained phenomenon?" (Friedman, 1974)

In Friedman's article, he describes three views of scientific explanations. One of viewpoints of explanation is that "scientific explanations give us understanding of the world by relating (or reducing) unfamiliar phenomena to familiar ones." We adopt this viewpoint when we construct our concept definition[1].

In our concept definition, we relate concept to a computational architecture, which is unambiguous and well-defined mathematical computation model. In this way, we avoid the use of human interpreted terms in defining concept. Next section, we will ground our concept definition to existing computation models.

2.2 Concept and Language

When we want to define concept in context of language, we must consider the relations between concepts and languages first. The relations between concepts and languages are

[1] Actually, the explanation of definition covers three views in Friedman's article. We give a more detail analysis in Appendix A.

complicated. When studying the relations, different disciplines have different focus and different assumptions. For example, psychologist may focus on concept development, and hence the language is just tokens to denote concepts in human mind. For some theorists, language is just tokens. They ignore concepts and may focus on topics such as the learnability of language, language identification (Gold, 1967). For Noam Chomsky, language has its structures and is generated by deep structures in the human mind, and concept is a general term to refer idea in mind. For some linguists, concepts denote components in real languages such as phoneme, words, phrases, and sentences. In this case, concepts are denoted by tokens. However, some scholars assert that concept has its own internal structure. For example, Margolis and Laurence (2011) investigate many proposals about the structure of lexical concepts[2]. We adopt this viewpoint and use continuation as an instantiation of concept's internal structure. Moreover, we embed the continuation in communication model to let our concept definition have a solid computational ground.

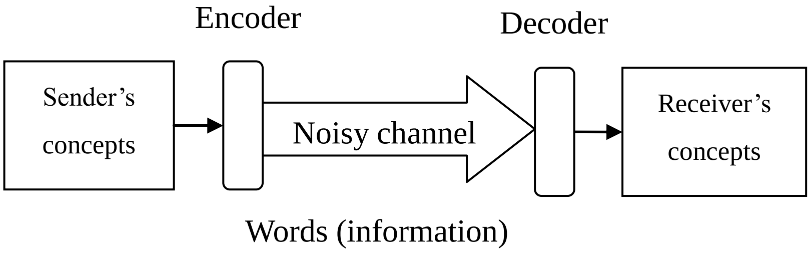

Figure 1. Diagram of Shannon's communication system for language.

Now, we consider computational models about language. In Shannon's communication system (Shannon, 1948), the language is signals from sender to receiver, and the signal may be corrupted by noises when signals are transmitted (see Figure 1). In this communication system, concepts are conveyed by words and are transmitted to receiver with possibility of

[2] In their definition, lexical concept is a word-sized concept, and it can be used to compose complex concepts.

misunderstanding. In this case, concepts are transparently encoded in words and jump into head when receiver decodes the received words. Although the concept definition is not necessary here, the communication system do capture an important aspect of language. Not like Shannon who concerns the communication process, we are interested in the words' ability to trigger receiver to do computation. Because words carry concepts, the concepts trigger computations in both sides.

Nowak's evolutionary language game (Komarova, Niyogi, & Nowak, 2002; Nowak, Plotkin, & Krakauer, 1999; Plotkin & Nowak, 2000; Trapa & Nowak, 2000) further extends the communication system in a game setting. The meanings of signals are explicitly modeled in the evolutionary language game. In his settings, sender and receiver have a matrix P and Q, respectively. The matrix P encodes the sender's knowledge of signals associated with meanings[3], and the matrix Q encodes the receiver's knowledge of signals associated with meanings. In this way, a concept can be denoted by many signals and vice versa. He then defines language $\mathrm{L}(\mathrm{P},\mathrm{Q})$ for an individual. For two individuals, they may have different knowledge about language, and hence they have different language $\mathrm{L}(\mathrm{P},\mathrm{Q})$ and $\mathrm{L}'(\mathrm{P}',\mathrm{Q}')$ respectively. Trapa and Nowak (2000) defines payoff function $\mathrm{F}(\mathrm{L},\mathrm{L}')$ and proves that a group of individuals with random knowledge of language can communicate to each other in the evolutionary language game setting. In summary, Nowak proves that it is possible to communicate concepts using signals even the initial knowledge of signals and concepts are different between individuals. But Nowak's model consider language $\mathrm{L}(\mathrm{P},\mathrm{Q})$ of an individual as a whole, it is difficult to apply his results in various applications.

Although Nowak asserts that the communication between different individuals with different language knowledge is possible no matter the individual being a human or a machine, some philosophers concern the ability for a machine to understand the communicated

[3] In Nowak's paper, signals are associated with objects, which are anything that can be referred to, including concepts and meanings in human mind.

information. This is the core problem questioned in the famous Chinese Room problem (Searle, 1980).

The Chinese Room problem is a thought experiment. It assumes that a computer system already passes the Turing test in Chinese language. If a man who has no knowledge about Chinese replaces the computer, the conversation in the Turing test can continue theoretically. The man runs the program, but this man doesn't understand Chinese. Searle concludes that the machines cannot understand human languages even thought machines conduct successful conversations with human. It means successful communication does not entail successful understanding. Although Searle’s argument is controversial, it highlights an important point that communication model cannot completely model all aspects of language. Language understanding is an important aspect of language and concept modeling. It inspires us to give a concept definition.

2.3 Defining Concept as a Continuation

Now, we have mentioned that Nowak’s evolutionary language game asserts communication between machine and human is possible. On the other hand, the Chinese Room problem argues that communication does not entail understanding. How do we build machines with abilities of language communication and understanding?

In logical positivism, all meaningful statements must be verifiable. This follows that if we want to assert the statement “a machine understands language”, we must have a proof for verification. The empirical proofs for language understanding can be anything that been used to test a human for his/her understanding of language. We denote these empirical proofs as a verification set V and proof $v \in V$.

In Nowak’s evolutionary language game, concept encoding matrix $P \in \mathbb{R}^{nxm}$, in which *n* is the number of objects (concepts) and *m* is the number of signals (words). The concept decoding matrix $Q \in \mathbb{R}^{mxn}$ is defined in the same way. Now, because we must have proofs for asserting statement "system understands a concept *i*", we let concept encoding matrix P be a product of two matrices $G \in \mathbb{R}^{nx|V|}$ and $H \in \mathbb{R}^{|V|xm}$ (see Figure 2). The concept decoding matrix $Q = RS$ can be defined in the same way, in which $R \in \mathbb{R}^{mx|V|}$ and $S \in \mathbb{R}^{|V|xn}$.

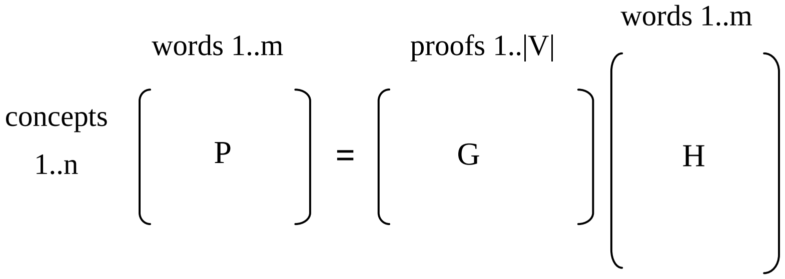

Figure 2. The relations between concepts, words, and proofs.

We denote matrix G as a concept-proof matrix, and matrix H as a proof-word information matrix. For example, for a concept $c_i \in \mathbb{R}^{1xm}$, $1 \leq i \leq n$, we may have multiple proofs $v_i \in V$, $1 \leq i \leq |V|$, to assert that system understand concept c_i. With this formation, the machine has verifiable nature of its internal structure for human understanding.

In Trapa and Nowak (2000), they define different types of languages based on constraints for matrices P, Q, and payoff function $F(L, L')$. For example, a language $L(P, Q)$ is a Nash language if $F(L, L) \geq F(L, L')$ for all language $L'(P', Q')$, in which P' has same dimension with P, Q' has same dimension with Q, $F(L, L') = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(p_{i,j}q'_{j,i} + p'_{i,j}q_{j,i}\right)$, $\sum_{j=1}^{m} p_{i,j} \leq 1$, and $\sum_{i=1}^{n} q_{j,i} \leq 1$. On the other hand, if $P = GH$ and $Q = RS$, we do not have the necessary to impose constraints for matrices G, H, R, and S. Matrices G and S are for verification by the third individuals. We can let $G = \left(g_{i,k}\right)$ and $g_{i,k} \in \{0, 1\}$ which means if proof *k* is used to verify concept *i*, $g_{i,k} = 1$, else, $g_{i,k} = 0$. In this case, a proof can be used to verify multiple concepts and vice-versa.

Now, we have grounded the understanding of a concept on Nowak's evolutionary language game, which is a solid computation model. In this model, concept $c_i = (c_{i,j})$, $1 \le j \le m$, $c_{i,j} = \sum_{k=1}^{|V|} g_{i,k} \times h_{k,j}$, in which $G = (g_{i,k})$ and $H = (h_{k,j})$. In other word, concepts are represented by words and proofs.

We can define an operational measure for language understanding. We have the Equation 1, which states that if the difference between matrices G and S of human and machine is smaller than a threshold $\delta \ge 0$, we assert the statement “the machine understands the language like human” is true.

$$\sum_{i=1}^{n} \sum_{k=1}^{|V|} \left| g_{i,k}^{machine} - g_{i,k}^{human} \right| + \left| s_{k,i}^{machine} - s_{k,i}^{human} \right| \le \delta \qquad \text{Equation 1}$$

Although the definition of concept understanding on Nowak’s evolutionary language game is clear, it is not a good choice to use this as a concept definition. Nowak’s evolutionary language game describes stationary status of a group of communicating individuals, but we want a concept definition that can be used to design a system to communicate with members of group with stationary language knowledge.

Now, consider the situation that two systems communicate with each other, and a scientist studies the change of concept understanding of the two systems in the communication. This situation is much like that developmental psychologists study human language development except the scientist has access to internal structures of systems. When a system receives a signal (word), it may triggers many concepts, and hence a concept may triggers proofs to be verified by the scientist. When this process continues, in a snapshot, how does the scientist define what is concept in a system?

This scientist will notice that there are words, concepts, computation, and computed proofs of concepts. For a concept, there is pre-existing structure in the last communication, and system uses the information to compute related proofs (see Figure 3). We can say that a concept is represented by this pre-existing structure, and this structure is actually an un-finished computation in the communication process. In other words, **concept is a continuation** in a computation process. We propose **concept as continuation** to be the definition of concept.

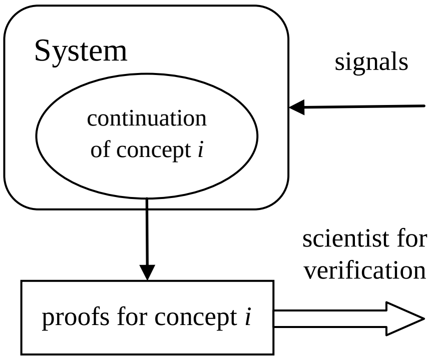

Figure 3. Relations of signal, continuation, concept and proofs.

This definition is well-defined because this is based on evolutionary language game setting. This definition also gives us a computational viewpoint of concept, which gives great advantages when we want to design real systems. In this concept definition, we have a uniform view of system's concepts. In this definition, we can unambiguously define concept, concept computation, and language understanding. We can have a single viewpoint to integrate NLP tasks, which usually have their own definitions of concepts and understanding. This uniform viewpoint of concept is very important when we are interesting in designing a real intelligent system and when we are integrating multiple NLP tasks into a single intelligent system.

Notice that although we define concept as continuation, we do not specify the structure of a continuation, what does a concept refers to, and what a proof is. We think these as the flexibility of our concept definition and will explain these issues in detail later. We consider some theoretical aspects of our concept definition first.

2.4 Some Theoretical Aspects of the Definition

We are concerned with some theoretical aspects of concept definition. The central problem is stability of a system. The stability problem has three different aspects, which are called **dogma stability**, **input stability**, and **test stability**. We describe them below.

1. A system of dogma stability means the system has ability to strictly obeying some specific rules embedded by human. Human may want machine to obey the rules in any situation that the system may encounter. We can also use controllability to denote dogma stability of a machine.
2. A system of input stability means its language understanding ability will not change significantly when the system has small changes inside the system.
3. A system of test stability means the system has a general knowledge of language, and its language understanding ability will not change significantly when we use different verification sets, which may also change when the language is in evolution. Because a system is also a learning algorithm, the test stability is just the generalization of a learning algorithm.

Using our concept definition, we can study these aspects of stability from machine learning viewpoint. When studying these stabilities, the theories of concepts and time orders of magnitude are two very important factors. The theories of concepts define how signals are sampled from reality. This sampling of signals reflects data distribution, and then affects the standard of judging system's understanding ability. The time order of magnitude defines the scope of validity of stability analysis. There are three time orders of magnitude to be concerned: static world, short-term period, and long-term period. We describe them below.

1. Static world: the time factor is not considered in the analysis. In this case, the distribution that generates signals is unknown but fixed for all time. Most analyses of

machine learning algorithms are in this situation.

2. Short-term period: the system has small changes which may be due to system's shift in its learning process or small environmental shift. For example, when a man changes his taste of food, a machine must adopt to this change to serve this man.
3. Long-term period refer to system's lifetime. Although we have full control in the birth of a system, we want to understand how it will behave in a long time without human's interference.

When the period is longer than system's lifetime, test stability can be modeled by Nowak's evolutionary language game. This implies that human language will co-evolve with concepts in machines because signals are exchanged between human and machines just like the setting in evolutionary language game. Input stability and dogma stability is meaningless when time period is larger than system's lifetime. In Table 2, we sketch the theorems that are adopted in analyzing relations between stability and time period. In the table, blank cell means relations are not covered in this dissertation, and ML denotes machine learning.

	Input stability	Test stability	Dogma stability
Static world	Sensitivity analysis (stability in ML)	Generalization in ML	Rice's Theorem
Short-term period	Stability in ML		
Long-term period	Stability in ML	The No Free Lunch Theorem	

Table 2. The relations between stability and time factor.

We will use existing theorems in machine learning literatures to analyze system's stability. We define some mathematical terminologies first.

Suppose a system has *n* concepts, $|V|$ proofs, and their continuations $C = \{c_i | 1 \le i \le n\}$, in which c_i denotes continuation of concept *i*. Concept *i* has proofs $\{ j | 1 \le j \le |V|, p_{i,j} = 1\}$, and $P = (p_{i,j})$ is concept-proof matrix which is given by human. We define empirical loss $\mathrm{h}(\mathrm{C},\mathrm{P}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{|\mathrm{V}|} \ell(\mathrm{p}_{\mathrm{i,j}},\ \mathrm{f}(\mathrm{j},\mathrm{c}_{\mathrm{i}}))$ to measure language understanding of system, in which $(\mathrm{i},\mathrm{c}_{\mathrm{i}}) \in \mathcal{X}$ is the input of function f, $\mathrm{f}{:}\,\mathcal{X} \to \{0,1\}$ is system learned knowledge to judge the relation between a concept and a proof, and $\ell(\mathrm{f},\mathrm{p}_{\mathrm{i,j}},(\mathrm{j},\mathrm{c}_{\mathrm{i}})) \in \mathbb{R}$ is a loss function for function f, $\mathrm{p}_{\mathrm{i,j}}$, and $(\mathrm{j},\mathrm{c}_{\mathrm{i}})$. Because continuation is an un-finished computation and may changes, it is difficult to analyze system in this form. Therefore, we let the system has a specific internal structure for the convenience of analysis. We suppose that the system uses its knowledge to generate intermediate data and uses intermediate data $\mathrm{D} \in \mathbb{R}^{\mathrm{nxd}}$ to learn a function for P (see Figure 4). In this setting, we can analyze system in two stages: the feature generating stage and machine learning stage. System generates features using continuations of concepts, and adopts standard machine learning approaches to learn a good function to show its understanding of language.

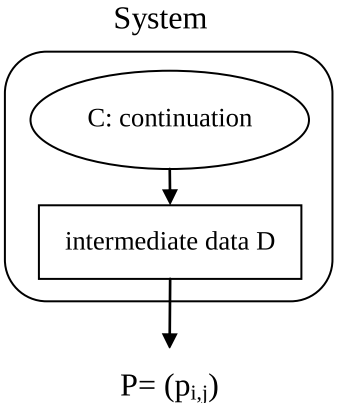

Figure 4. Internal structure of system for analysis.

The input stability of system considers the problem that the language understanding ability will not change significantly when the system has small changes. The small changes may be caused by external signals or by system’s internal operations, and the changes may be in continuations or in intermediate data D. This stability problem concerns if a system can act like a stable average person and won’t go crazy for changes from noises, inputs or internal operations.

If the small changes is in intermediate data D, according bipartite stability of ranking algorithms (Agarwal & Niyogi, 2005), there are learning algorithms that can result in a stable system. Because it is helpful to understand definition of system stability, we state stability of ranking functions in information retrieval in detail. In information retrieval, a query has a set of relevant documents, and system learns a ranking function to rank relevant documents and non-relevant documents of a set of queries. In our system, the concept acts like query, proof acts like relevant document, and non-proof of a concept acts like non-relevant document. Bousquet and Elisseeff (2002) gives many stability definitions for learning algorithms, and Agarwal and Niyogi (2005) uses similar definitions to prove that some ranking algorithms are stable. For example, ranking algorithm RankingSVM (Joachims, 2002), which uses reproducing kernel Hilbert space (RKHS) with kernel, has uniform leave-one-instance-out stability[4] (Geng et al., 2008).

Definition 1. **Uniform leave-one-instance-out stability**

Let $\mathrm{x} = (\mathrm{d}, \mathrm{p}) \in \mathcal{X} \times \mathcal{Y}$, in which $\mathrm{d} \in \mathrm{D}$, $\mathrm{p} \in \mathcal{Y}$, $\mathcal{Y} = \{0,1\}$,[5] D is the input space, and p is the label. Let $\mathrm{S} = \{\mathrm{x_i} = (\mathrm{d_i}, \mathrm{p_i})\}_{\mathrm{i}=1}^{\mathrm{n}}$ denotes a training set that is drawn i.i.d. (Independent and identically distributed[6]) from an unknown distribution, and $\mathrm{S^i} = \mathrm{S} - \{\mathrm{x_i}\}$ denotes a training set with one instance out. The $\mathrm{h_S}$ denotes the resulting model of a learning algorithm L using training set S. This model $\mathrm{h_S}$ minimizes loss function $\sum_{\mathrm{i}} \mathrm{loss}(\mathrm{h}, \mathrm{x_i})$, where $\mathrm{loss}(\mathrm{h}, \mathrm{x_i}) = \mathrm{c}(\mathrm{h}(\mathrm{x_i}), \mathrm{p_i})$. The cost function $\mathrm{c}: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. Now, we say L has uniform leave-one-instance-out stability φ, $\varphi: \mathbb{N} \to \mathbb{R}$, if

$$\forall \mathrm{x} \in \mathcal{X} \times \mathcal{Y},\ \left|\mathrm{loss}(\mathrm{h_S}, \mathrm{x}) - \mathrm{loss}(\mathrm{h_{S^i}}, \mathrm{x})\right| \le \varphi(\mathrm{n}).$$

[4] Uniform leave-one-instance-out stability is defined in Geng et al. (2008) and is similar to uniform score stability defined in Agarwal and Niyogi (2005).

[5] In Agarwal and Niyogi (2005), $\mathcal{Y}$ can be the set of real number.

[6] In our definition of intermediate data D, it must be drawn i.i.d. in order to adopt this stability definition. This can be done by many methods. For example, we can draw a i.i.d. signal set, and feed this signal set to generate an i.i.d. intermediate set.

This definition states that if learning algorithms have uniform leave-one-instance-out stability, the resulting model will not change significantly when the training set has a small change (leaving one instance out). In Agarwal and Niyogi (2005), they proves that $\varphi(\mathrm{n}) = \frac{\kappa^2}{\mathrm{n}\lambda}$, where $\lambda > 0$ is a regularization parameter, $\forall \mathrm{x}$ in input space, $\mathrm{K}(\mathrm{x},\mathrm{x}) \leq \kappa^2 < \infty$, and $\mathrm{K}(\cdot,\cdot)$ is the kernel of RKHS. Geng et al. (2008) show that for two training sets S_1 and S_2, $\forall \mathrm{x} \in \mathcal{X} \times \mathcal{Y}$, $\left|\mathrm{loss}\left(\mathrm{h}_{\mathrm{S}_1},\mathrm{x}\right) - \mathrm{loss}\left(\mathrm{h}_{\mathrm{S}_2},\mathrm{x}\right)\right| \leq \frac{\kappa^2\ (|\mathrm{S}_1|+|\mathrm{S}_2|-|\mathrm{S}_1 \cap \mathrm{S}_2|)}{\lambda \min\ (|\mathrm{S}_1|,|\mathrm{S}_2|)}$, where $|\mathrm{S}|$ is the size of set S.

In static world, the whole system still has good input stability, because the role of continuations is just like a component of algorithm, and it will generate same distribution of intermediate data D if the distribution to generate signals is the same. In this case, the generalization (test stability) of system can be analyzed by using generalization theorems in ML. The generalization ability depends on the learning algorithms we adopted in intermediate data D.

According to the No Free Lunch Theorem (Wolpert & Macready, 1997), if a system is built based on static world assumption, which is a viewpoint that holds by empiricism and rationalism on the theories of concepts, theoretically, the system cannot guarantee to have a good language understanding in long-term period for test stability. The No Free Lunch Theorem states that if a learning algorithm performs well in some tasks, it must perform badly in other tasks. The system is a learning algorithm, and it understands language well in the very beginning. Theoretically, we cannot guarantee how the world will change. Therefore, it is possible that this already built machine will perform badly in some cases in future. This theorem prompts that we can only have results in probability for system’s understanding ability in long-term period.

We turn our attention to dogma stability in static world. A system with dogma stability means it will strictly obey some specific rules without exception in any cases. The most famous example is The Three Laws of Robotics, which is coined by the science fiction author Isaac Asimov (1920 – 1992). Briefly speaking, scientists do not want a designed machine to hurt people. The possibility to design such machine is the issue of dogma stability. Dogma stability is different from input stability because system maybe misunderstands some concepts but is still have a good understanding of language. According to Rice's Theorem, which states that it is un-decidable for any non-trivial property of Turing machines, we can say that dogma stability can't be guaranteed if we do not put some assumptions on the human world. On the other hand, because scientists can always conclude a probability of violating dogma stability for a system, dogma stability may not a major concern for some real systems.

In the long-term period, the input stability is the same as short-term period because system can continuously make a small changes like in short-term period. Therefore, we discuss the input stability in short-term period. In the short-term period, the system may modify their concepts to adopt the environmental changes, which also means the distribution to generate signals may not be stationary. For input stability in short-term, the input stability of whole system and intermediate data D depends on how we design the system. If all concepts are interrelated, the system with continuations is much like a recurrent neural networks (RNNs), and the output of RNNs is usually non-linear and is hard to be predicted (Barabanov & Prokhorov, 2002). Therefore, if we design system to have independent concepts or independent group of concepts, we can analyze system theoretically. We can analyze this issue in different cases such as same signal distribution but different concepts, different signal distribution but same concepts, and different signal distribution and different concepts. In other words, the proposed concept definition can be analyzed mathematically. This is very important when we want to design a real intelligent system.

2.5 Related Work in Concept Definition

Scholars of different disciplines have their concept definitions to apply in their work. For philosophers, concept has intension and extension that represent knowledge of concept in human mind. For logicians (Jurafsky & Martin, 2009a, 2009b), a concept can be a symbol to denote an object in a logic model, can be a category to denote a group of objects, and can be a first order logic sentence(s) which specifies its relations with other concepts. In Formal Concept Analysis (Priss, 2006), concepts are objects that have attributes. The objects and attributes are defined by human's commonsense, in which its meaning is interpreted by human in the context. For linguists, concept may be represented by words. Therefore, they use words to denote a lexicalized concept. The distinguished WordNet (Fellbaum, 1998) database adopt this viewpoint, and no formal definition of concept is given. In WordNet, concept is represented by a *synset* which contains words for a concept. For ontology builders and users, concept may play different roles in ontology. It may be an object, predicate, quantifier, function, and relation. These terminologies gain their meaning in the ontology. Its connection to real world is also interpreted by human. For researchers in artificial intelligence, concept may be represented by words or an object in a logic model.

In summary, concept definition in these disciplines is an object to be operated, while in our concept definition, a concept itself is a computational process that uses a continuation to represent it. Moreover, the continuation exists in the environment it lives like a continuation in programming language. This viewpoint adopts pragmatism concept theories. Its connection to real world is defined by its ability of understanding language and is modeled inside the definition. A continuation do not contain all information because some information is stored in its environment. A continuation is similar to a device that stores links to its environment and links to machine's internal states. Therefore, a continuation may has its internal structures to store different types of information. We discuss this issue in next chapter.

In the viewpoint of continuation, human do not interpret a concept in machine. Human just provides proofs to test the comprehension of concept in language understanding of a machine.

Computer scientist John Sowa (Sowa, 1984) gives a concept definition in pragmatism viewpoint, and we quote it below.

> “Concepts are inventions of the human mind used to construct a model of the world. They package reality into discrete units for further processing, they support powerful mechanisms for doing logic, and they are indispensable for precise, extended chains of reasoning. But concepts and percepts cannot form a perfect model of the world,—they are abstractions that select features that are important for one purpose, but they ignore details and complexities that may be just as important for some other purpose.” (Sowa, 1984, p.p. 344)

The core insight of his definition is similar to our definition which captures the computational aspect of concept, but we further formulate concept definition in evolutionary language game and add mechanism for verifying language understanding. Marvin Minsky proposes a similar viewpoint of concept definition but uses different terminologies. In his book The society of Mind (Minsky, 1986), mind is a society which is composed of a group of agents. These agents represent various processes in human’s brain, and these processes can be any concepts interested by researchers such as free well, the sense of self, belief, memory, and consciousness. In our definition, we denote all processes in human mind as concepts and do not put any assumption on the structure and implementation of concept in order to gain the ability to analyze system theoretically in modern machine learning perspectives. Our concept definition is also similar to intelligent agent (Russell & Norvig, 2003) in artificial intelligence literatures, but we connect agent’s output to language understanding.

Barker (2004) emphasizes the similarities between formal languages and natural languages and uses continuation to analyze linguistic phenomena in natural language. He treats quantification words like *everyone, no one*, and *someone* as a continuation, and defines these words in formal language context. He uses control operators like *control*, *prompt*, *shift*, and *reset* in delimited continuation (Felleisen, 1988) to demonstrates computation of quantification words in syntax tree. Barker also studies a phenomenon called *focus*, which is denoted by focus particles such as *only*. His approach use first order logic to represent the semantic of sentences like the approaches in computational semantics (Jurafsky & Martin, 2009a). Because continuation is a flexible mechanism to handle execution flow of formal language, he uses continuation as a mechanism to handle complex relations and phenomena in natural language, such as coordination, ambiguity, and quantification. In Baker's formulation, a concept actually is a predefined continuation that has specific effects in parse tree. Although this definition is similar to our definition, the meaning of a concept is interpreted in FOL context, and hence, is interpreted by human.

When considering the relations between concept and language, researchers usually regard concepts as states of mind and study the procedure of translating mind states to languages. For Noam Chomsky (1986), the translation procedure is the knowledge of languages, and languages are internalized language (I-language) that translating the structure of concepts (mind states) to externalized language (E-language), which is independent of mind. In this viewpoint, language understanding is the problem to understand the correspondences between I-language grammars and E-language grammars. In our concept definition, the grammars are one type of concepts, and the E-language is just one type of proofs that can be adopted to measure system's understanding level.

When considering a concept to be a program that has the ability to do something in an environment, researchers usually regard concepts as a computer program. They follow the approaches of reductionism, which reduces complex thing to many simpler and smaller things and combines these smaller results to solve the complex thing. For example, when studying machine understanding, researchers in natural language frame the understanding problem to many smaller problems such as named entity recognition (NER), co-reference resolution, template element, template relation, and scenario template in the Message Understanding Conference. In this case, a program that archives good results in sub-problem is regarded as understanding language well. This approach is similar to our concept definition, which define concept to be a program represented by continuation, and we further link this approach to evolutionary language game to form a more general framework to integrate sub-problems. In other words, we provide a general framework to integrate many sub-systems, and this integration is still within language understanding framework.

In the next section, we will mention some considerations of the proposed definition in implementation.

2.6 Considerations of Implementation

We have sketched a framework in our concept definition, and we give detail descriptions of continuation and proofs here.

In our concept definition, we equally treat all types of concepts, but in literatures, researchers may manually gives definitions for concepts like beliefs, goals, plans, commonsense, knowledge, and intentions (Mueller, 2010). It is straightforward to build continuations for these concepts. Therefore, when we implement our concept definition, the implementation of continuation and the source of proofs are the keys to build intelligent systems. In this dissertation, we use a concept representation scheme to represent continuation and use automatically extracted knowledge as proofs. We will explain concept representation scheme in chapter 3 and knowledge extraction in chapter 4.

We put some words on the proofs. We already have a continuation to represent a concept. Now, we explain how to test concepts with proofs here. In Figure 3, we use a set of proofs to verify that a concept is studied by system. When the concept is settled in the concept-proof matrix, we use one row to represent it. For example, we want to know the concept <*car*> is well acquired by system, we can test it with proof <*a car is a vehicle*> or <*a car is a human*>. When we have a set of similar proofs such as <*a car is a machine*> and <*a car has four wheels*>, we can change the concept-proof matrix in other equivalent formats. One of the equivalent formats is to assign relations to concept. For example, the <*car*> has human-readable relations such as <*is_a*>, <*has*>, and <*type_of*>. Using this way, we connect system implementations to a general case and easy to be understand.

Chapter 3. Concept Representation

We use a concept representation scheme to represent continuation in this chapter. The proposed concept representation scheme is similar to continuation in programming language but has greater flexibility to adopt complex world. Just like a continuation in programming language store state of current program, we use structured format to represent the state of a continuation. In order to connect state to its environment, we use an explicitization process to do this job. In summary, we define our concept representation "**a scheme that employs an explicitization process in a specific perspective to elicit a mathematical object for a concept**." The mathematical object is the intermediate data D in Figure 4, and this object usually is adopted as feature matrix for machine learning algorithms.

In this chapter, we organize materials in the order below.

(1) We describe a proposed structured format used in a scheme.

(2) We describe related work of concept/word representation in natural language processing.

(3) We describe the connection to feature engineering of machine learning.

3.1 Representation of Continuation

We use traditional frame structure to store the static knowledge of a concept, and we use the explicitization process to represent the dynamic part of a concept. We show the relations between frame structure, explicitization process, and system in Figure 5. In Figure 5, system can access the internal frame structure, and this ability completes our third claim that the concept definition (concept as continuation) has the build-in nature in deep analysis for human and for intelligent system itself to understand internal structures of an intelligent system. In summary, our concept definition originates from the evolutionary language game and has native origin in computational perspective. We use continuation to eliminate the need of undefined terms in the definition, and use frame structure to let the system having build-in nature in deep analysis about system's behaviors. The whole concept representation scheme is similar a feature engineering process except the scheme is grounded in a language understanding context.

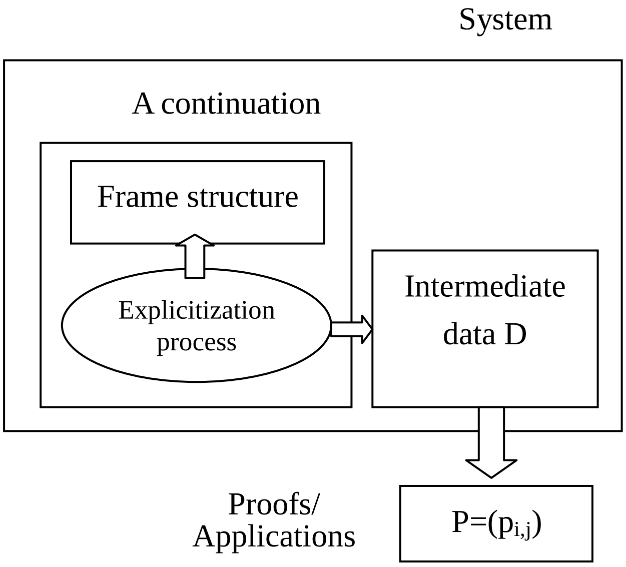

Figure 5. The relations between frame structure, explicitization process, and system.

The frame structure contains static knowledge of a concept, and explicitization process represents dynamic connection between static knowledge and intermediate data D. Now, we face a cold start problem, which means we do not have the static knowledge but we must use the knowledge to let explicitization process to generate intermediate data D. Therefore, we extract knowledge from web pages to solve cold start problem.

The frame structure may contains many kinds of knowledge. We classify frequently used knowledge in frame structure below.

(1) Knowledge of language:

This category contains knowledge about language, including lexical knowledge and syntactical knowledge.

(2) World knowledge:

This category contains world knowledge and knowledge that connects world knowledge to language knowledge. We further classify the knowledge into three types, including relation knowledge, pattern knowledge, and grounded knowledge. Relation knowledge connects two phrases by using relation such as *type_of* relation in knowledge <*bank*, *type_of*, *company*>. Pattern knowledge is a knowledge extraction pattern and connects concept to language usages. For example, pattern <*is_a*, *STRING*> extracts knowledge <*bank*, *is_a*, *company*>. Grounded knowledge contains source sentences like the sentences we remembered and frequently used as prototype examples. For example, <*The bank is a company incorporated …*> is a prototype sentence used in pattern <*is_a*, *STRING*>. We use these three types of knowledge to extraction world knowledge.

(3) Explicitization knowledge:

This is the knowledge that been learned in explicitization process. Its detail information of the knowledge is subject to adopted learning algorithms.

3.2 Related Work

In literature, many representation schemes have been proposed. Some schemes are used to represent words while some schemes are used to represent concepts. We classify these representation schemes (static knowledge) in two categories (from human and from texts) according the source of the representation. We describe these representations below.

The first type of knowledge source is human. Researchers directly derive static knowledge from human. These static knowledge include commonly used resources in NLP such as linguistic database (WordNet, FrameNet, VerbNet), ontology (Suggested Upper Merged Ontology, SUMO), commonsense knowledge (CYC, ConceptNet), and collaborative knowledge base (Freebase). Experts or general users manually enter the knowledge, and the size of the knowledge is limited.

The second type of knowledge source is texts. Researchers can design systems to extract knowledge from texts or design mathematical models to represent knowledge in texts. We ignore knowledge extraction here and describe the mathematical representations.

Researchers use mathematical objects to derive representations from texts for words and concepts. These derived representations may be adopted in machine learning algorithms, but some representations store representations in internal network. These mathematical objects can be classified into three categories.

(1) **Frequency-based**: This category counts the frequency of features in texts and uses approaches to select features. For example, distributional representation approach (Harris, 1954) collects co-occurrence words in texts to represent a target word or concept. Turney and Pantel (2010) gives a good review on using different kinds of lexical patterns to derive meaning for words.

(2) **Model-based**: This category relies on mathematical models to build representations and capture static meaning of a word. For example, Latent Semantic Indexing adopts singular value decomposition (SVD) to derive latent concepts for word representation. Brown clustering (Brown et. al., 1992) uses clustering algorithms to cluster similar words and assigns bit strings to represent words in a cluster. Word embedding (Bengio, 2008) encodes word knowledge in a real-valued vector and uses neural language model to learn the representation. Turian et. al. (2010) adopts many semi-supervised learning algorithms including word embedding to represent words and conducts experiments on many NLP tasks to compare the usefulness of different approaches.

(3) **Operation-based**: This category also relies on mathematical models to build

representations, but the algorithms in this category focus on capturing dynamic aspects of words, which are different from the model-based approaches. For example, holographic lexicon (Jones & Mewhort, 2007; Plate, 1995, 2003) uses neural network framework to learn representations that encode word order information and word composition information in distributed representations. Thater et. al. (2010) uses vector model to integrate compositionality knowledge of concepts and context information of a word.

In addition to above approaches, some researchers adopt logic to represents word meaning. These approaches usually need human to encode domain knowledge manually. Therefore, it is similar the approaches that derive knowledge from human.

When researchers want to measure the usefulness of a concept/word representation scheme, they proposed many criteria for this purpose. Commonly used intrinsic criteria to evaluate a representation include (a) encoded knowledge/information, (b) computational properties, such as accessibility, efficiency, affordance of generalization, robustness and graceful degradation (Plate, 2003), (c) supported operations, such as composition, decomposition, and manipulations, (d) expression power of the representation, (e) transparency, which means the results is easy to be understand by human, and (f) flexibility, which means the representation can be used in different situations. Researchers also adopt extrinsic applications to assert the usefulness of a representation. These applications can be any NLP applications such as chunking, WSD, word similarity, and GRE word test.

In our representation, we emphasize on transparency and flexibility because we are want the representation can easily integrate different kinds of knowledge and has internal structure that is readable for human. Therefore, we put the readable part in the frame structure and put the un-readable and somewhat mysterious part in explicitization process. In this way, we can easily adopt our representation in existing machine learning approaches, and we still have control on what is the learned knowledge of a system.

Our representation is similar to feature engineering step in traditional machine learning approaches. The explicitization process is just like a feature engineering process, and the frame structure is just the resources that a feature engineering process can use. We integrate these two parts to let the system has ability to know what is the learned knowledge. Researchers also have access to system’s internal structures in this approach. When we want to design a real intelligent system, this is a very important feature.

When we consider the frame structure and the concept-proof matrix, these two objects can form a closed loop in some settings, in which one object is the target to be learned and the other is the knowledge source.

In a summary, the proposed concept representation is a framework, and its intention is for developing real intelligent systems and is not for a specific applications. With the ability to have control on system’s internal knowledge and the solid concept definition in language understanding perspective, we will have a good starting point for implementing intelligent systems.

3.3 Framework of Knowledge Extraction

We use automatically extracted knowledge as proofs of system’s understanding and as knowledge in frame structure. By using automatically extracted knowledge, we can alleviate human’s efforts to assert system’s understanding level. The ultimate goal of our concept definition is to build a system like human, which can learn knowledge from environment and communicate to people for the knowledge they learned. Because we have limitations on time and computation power, we demonstrate the use of our concept definition and derived concept representation scheme in the knowledge extraction process. We will describe our work in next section.

In the algorithm, the classifier C can be for a set of proofs/knowledge of a concept or many concepts. The algorithm can also run in batch mode if we have limited resources, but this limitation does not downgrade the value of the algorithm. This algorithm is much like algorithm used in NELL architecture (Carlson et al., 2010), but our algorithm has the ability to analyze knowledge in its own frame representation.

The knowledge extraction framework is shown in Algorithm 1[7].

Algorithm 1. Automatic knowledge extraction from texts

```
Let   K= a seed of knowledge set
Build classifier C from K
until
    Identify knowledge in texts
    Induce new lexical or syntactical patterns P
    Extract new knowledge from texts using patterns P
    Assert the quality of new knowledge using classifier C
    Let K be union of new knowledge set and K
    Build new classifier C from K
until stop criteria
```

3.4 Applications of Concept Representation

Using extracted knowledge to represent concept is common in many literature (Chklovski, 2003; Etzioni, Banko, Soderland, & Weld, 2008; Singh et al., 2002; Yu & Chen, 2010). Therefore, in our study, instead of extracting knowledge to build concepts, we focus on other related perspectives about concept representation and knowledge extraction.

To demonstrate the application of our concept representation scheme, we adopt two problems. First, we study commonsense knowledge classification and interpret feature engineering process in concept representation scheme. This shows that our concept

[7] A similar algorithm appears in our paper (Yu & Chen, 2010).

representation is more general than feature engineering process and is more feasible in the point of view of natural language processing. We describe these issues in chapter 4. Second, we demonstrate the use of eliciting information from different perspectives to learn knowledge of word sense disambiguation. This process can be interpreted in same scheme, but we have a novel viewpoint to deal with WSD problem, which is a quite well-known problem in natural language processing. We describe these issues in chapter 5. These two demonstrations illustrate the application of our concept representation scheme.

To study perspectives that are important and usually ignored by researchers, we examine two assumptions about the content and size of knowledge. We describe these issues in chapter 6. In chapter 6, we also describe some important preprocess steps when we want to extract knowledge from the web.

Chapter 4.

Commonsense Knowledge Classification

In this chapter, we adopt our concept representation scheme in commonsense knowledge (CSK) classification, a task which is to know whether there is a specific relation between two noun phrases.

When representing the static part of a concept in this chapter, we represent commonsense concepts (phrases)[8] in predefined slots, and a learning algorithm is adopted to learn classifiers for commonsense knowledge. We formulate the CSK classification as a binary classification problem. The classifiers detect if a relation is valid between a pair of noun phrases. For example, relation *CausesDesire* holds between phrases "*the need for money" and "apply for a job"*.

We organize materials of this chapter[9] in the order below.

(1) We introduce OMCS database first. We use CSK in our experiments.

(2) We investigate related work about CSK mining.

(3) We propose our concept representation scheme for complex concepts, which is a noun phrase in this case.

(4) We describe data processing, experiment settings, and conducted experiments.

(5) We report our experimental results.

(6) We compare feature engineering and our concept representation scheme.

[8] These words (concept, word, and phrase) are identical in this paper, but word and phrase are different when they are in the context of language.

[9] The materials in this chapter are from our paper (Yu & Chen, 2010).

4.1 OMCS Database

We conduct our experiments by using dataset from OMCS project[10], a public available database from MIT. This database contains CSK contributed by volunteers. The web volunteers enter sentences to a web system in a predefined format. In this way, each sentence has two aligned concepts corresponding to two arguments with a specific predicate (relation). The concepts can be a word or a phrase. There are many predicate types in this database. Table 3 lists some examples.

Predicate	Concept 1	Concept 2
CausesDesire	the need for money	apply for a job
HasProperty	Stones	hard
Causes	making friends	a good feeling
HasPrerequi.	having a party	inviting people
CapableOf	a cat	catch a mouse
HasSubevent	having fun	laughing
UsedFor	a clothing store changing room	trying on clothes
IsA	a swiss army knife	a practical tool
AtLocation	a refrigerator freezer	the kitchen

Table 3. Examples from OMCS database

Besides, each sentence has a confidence score determined by users collaboratively. When a user asserts a sentence as a valid CSK, confidence score of this sentence is increased by one. On the contrary, if a user negates this sentence as a valid CSK, its score is decreased by one. In this way, a confidence score of a CSK can be considered as an indicator of its quality.

4.2 Related Work

To assert relations between two concepts is a common task, and this task is useful for many

[10] http://conceptnet.media.mit.edu/dist/ (Last access: 2013/06/18)

purposes. The most important purpose is to enlarge the knowledge base in order to alleviate knowledge acquisition bottleneck in many AI fields. There are many different types of knowledge that can be extracted from texts. Approaches that acquire commonsense knowledge from different sources (Chklovski, 2003; Schubert & Tong, 2003; Singh et al., 2002) have been proposed.

Chklovski and Gil (2005) roughly classified CSK acquisition approaches into three categories according to the knowledge sources. The approaches of the first category collect CSK from experts. WordNet (Fellbaum, 1998) and CYC (Lenat & Guha, 1989) are typical examples. A lot of knowledge is collected by linguists or by knowledge engineers. These approaches result in well-organized and high quality knowledge base, but the cost is expensive and then limits the scalability. The approaches of the second category collect CSK from untrained volunteers. Open Mind Common Sense (OMCS) (Singh et al., 2002) and LEARNER (Chklovski, 2003) are of this type. These approaches employ the vast volunteers in the Web to contribute CSK and correct the input CSK. The resulting CSK assertions can be in the order of million, but its quality is not as high as WordNet. The last approaches collect CSK from texts/the Web using algorithms. TextRunner (Etzioni et al., 2008), KNEXT (Schubert, 2009), and the systems (Cankaya & Moldovan, 2009; Clark & Harrison, 2009; Girju, Badulescu, & Moldovan, 2006) are of this type. These approaches usually process texts or web pages first, and then use lexical or syntactical patterns to extract facts or general knowledge. Because the CSK mined is large, it is not feasible to examine all CSK assertions manually. The performance of a knowledge acquisition algorithm is evaluated directly by assessors' small sampling or indirectly by employing the extracted knowledge to some tasks such as word sense disambiguation and examining the performance of applications. These approaches have the feasibility of controllable knowledge domain and scalability of extracted knowledge base, but the quality of resulting CSK is hard to control.

4.3 Concept Representation Scheme for Phrase

For CSK classification, we propose a representation scheme to denote an assertion. In OMCS database, an assertion is already preprocessed to a tuple, i.e., (PredicateType, Concept$_1$, Concept$_2$). Because a concept is usually a phrase, we represent a concept by using slots. The number of slots depends on different approaches shown as follows. A slot in turn contains words, and a word is represented by a co-occurrence vector.

We use a co-occurrence vector $\mathbf{W}_\mathrm{i} = \left(\mathrm{f}(d_1, w_\mathrm{i}), \mathrm{f}(d_2, w_\mathrm{i}), \dots, \mathrm{f}\left(d_{|D|}, w_\mathrm{i}\right)\right)$ to represent a word w_i in a slot, where *D* is a dictionary with size |*D*|, d_j is the *j*-th entry in *D*, and $\mathrm{f}\left(d_j, w_\mathrm{i}\right)$ is the co-occurrence frequency of entry and in a corpus.

We propose three approaches to determine the number of slots and how to place words in a concept into slots. The three approaches are discussed as follows:

(1) **Bag-of-Words (BoW) Approach**: All words are placed in one slot. BoW is considered as a baseline.

(2) **N-V-OW Approach**: All words are categorized into three slots, named HeadNoun, FirstVerb, and OtherWords. HeadNoun and FirstVerb are the head nominal of a phrase and the first verb of a phrase, respectively. Those words that are not in the two slots are put into OtherWords slot.

(3) **N-V-ON-OW Approach**: All words are categorized into four slots, named HeadNoun, FirstVerb, Other Nouns, and OtherWords. HeadNoun and FirstVerb are interpreted the same as those in the second approach. We further distinguish other words by their parts of speech (i.e., noun vs. non-noun).

With the approaches defined above, we define vector $\mathrm{S}_{\mathrm{j,k}}$ of slot *k* in concept *j* to be $\boldsymbol{s}_{j,k} = \sum_{i \in \text{slot } k \text{ of concept } j} \boldsymbol{W}_i$. Vector C_1 of Concept$_1$ and vector C_2 of Concept$_2$ by using N-V-OW approach are define in $\boldsymbol{C}_1 = \left(\boldsymbol{s}_{1,1}, \boldsymbol{s}_{1,2}, \boldsymbol{s}_{1,3}\right), \boldsymbol{C}_2 = \left(\boldsymbol{s}_{2,1}, \boldsymbol{s}_{2,2}, \boldsymbol{s}_{2,3}\right)$. The concept vectors for BoW and N-V-ON-OW approaches are defined in the similar way.

An assertion is a tuple as described, but we ignore the PredicateType information here because it is usually the same within a predicate. For example, in *IsA* predicate, the keywords are “is” or “are” which did not help much for binary classification in our setting. Hence, an assertion is a vector $(\boldsymbol{C}_1, \boldsymbol{C}_2)$ in which $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ come from Concept1 and Concept2, respectively.

4.4 CSK Classification Algorithm

CSK classification algorithm is described in Algorithm 2.

Algorithm 2. CSK Classification Algorithm

```
    Preprocessing
1       Use POS Tagger to tag an assertion
2       Place words of concepts into slots
3       Derive vector of word $\boldsymbol{W}_i$ from a corpus
4       Represent an assertion
    Feature Selection
5       Normalize concepts $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ to 1, respectively
6       Calculate Pearson correlation coefficient of each feature in slot
7       Select the first 10% features in each slot
    Classification
8       Use support vector machine to classify assertions
```

In step 1, we use Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003) to get tags. In step 2, we identify the head noun and the first verb of concepts by heuristic rules. For example, the first appearance of a verb in a tagged sequence is regarded as the first verb, and the last noun in a phrase is considered as the head noun. We distinguish noun and non-noun by parts of speech. In step 3, we consider Google Web 1T 5-Gram as our reference corpus (Brants & Franz, 2006), and employ only 5-gram entries in this corpus.

The dictionary *D* in step 3 is a combination of WordNet 3.0 and Webster online dictionary[11] (noun, verb, adjective, and adverb). The resulting lexicon contains 236,775 entries. In step 5, the concept vectors are normalized to 1 respectively to equally emphasize on the two concepts. Only top 10% of features are selected.

4.5 Experiment Settings

We select positive assertions from OMCS and automatically generate negative assertions to produce a balance dataset for a predicate type. The positive assertions must meet the following four criteria: (1) the confidence score of an assertion must be at least 4; (2) the polarity (note that there are positive and negative polarities in OMCS[12]) must be positive; (3) a concept contains no relative clause, conjunct, or disjunct; and (4) the length of a concept is less than 8 words for simplicity. The negative assertions are generated by randomly selecting and merging concepts from the OMCS database. We ignore datasets of size smaller than 200. Table 4 lists the resulting nine datasets among 18 predicate types in OMCS.

Predicate	*CausesDesire*	*HasProperty*	*Causes*
Size	204	254	510
Predicate	*HasPrerequisite*	*CapableOf*	*HasSubevent*
Size	912	916	1026
Predicate	*UsedFor*	*IsA*	*AtLocation*
Size	1442	1818	2580

Table 4. Datasets for CSK classification.

For each dataset, we randomly split 90% for training and 10% for testing. Next, we use LibSVM (Chang & Lin, 2011) for classification. In SVM training, we adopt radial basis for

[11] http://www.mso.anu.edu.au/~ralph/OPTED/index.html (Last access: 2013/08/15)

[12] Only 1.8% of assertions with CS ≥ 4 have negative polarity.

kernel function, grid search in parameters (c, g) (80 pairs). After the best parameters are obtained, we train a model on training set by using these parameters, and apply trained model to test set. The same procedure is repeated ten times to obtain statistically significant results.

Note the performance variation of a classifier in classifying commonsense knowledge. We can view a train set as a knowledgebase that one person owns, and this person may be 38 good at some aspects but bad at other aspects. This kind of train set may over-fit on some aspects and miss-classify other valid CSK. Because we aim to obtain a general purpose CSK classifier with broad coverage, the performance variation is an important indicator to evaluate a CSK classification algorithm.

4.6 Experiment Results

The test performance of classifiers is shown in Figure 6. The standard deviation and database size are also shown in this figure.

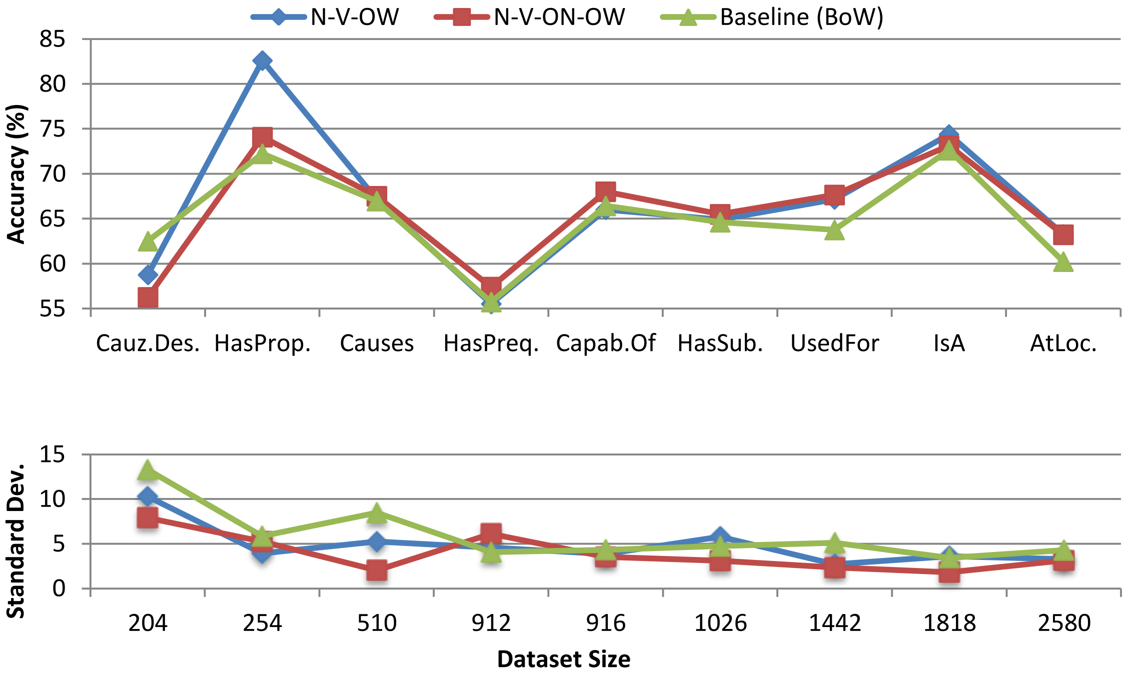

Figure 6. Classifiers' accuracy on nine datasets.

In Figure 6, N-V-OW approach and N-V-ON-OW app-roach are better than BoW approach except in *CausesDesire* predicate type. N-V-ON-OW approach tends to have smaller performance variation than BoW and N-V-OW approaches. N-V-OW approach has the best accuracy (82.6%) and variation in *HasProperty* predicate. *IsA*'s best result is 74.4%, which is comparable to similar problems in SemEval-2007 Task 4 Classification of Semantic Relations between Nominals (Girju et al., 2006).

In this task, we can see that it is possible to design classifiers to detect CSK in the texts.

We adopt the concept representation scheme to interpret feature engineering process in next section.

4.7 Interpretation of Feature Engineering Process

In Algorithm 2, we describe a general feature engineering process in machine learning viewpoint. If we are concerned the learned concepts in a machine and we want to integrate many tasks in a single viewpoint to understand the content of machine's concepts, feature engineering viewpoint is hard for this purpose because different tasks may have its own feature engineering procedure. Our concept representation scheme can be used for this purpose.

In integrating different tasks using concept representation scheme, we can see that all learned concepts are stored in a uniform continuation, and the explicitization process just connect datasets, learned models, continuation, and its environment. In this viewpoint, we can treat different tasks and different learning algorithms in a uniform way. If we want to analyze whole system in a formal mathematical perspective, this interpretation give us an advantage because all continuations (concepts) and learning algorithms can be treated in a same way. When we want to build human-like machine, this interpretation is very important.

Chapter 5. Word Sense Disambiguation

In this chapter, we use word sense disambiguation (WSD) to demonstrate the application of our concept representation framework. We organize materials of this chapter[13] in the order below.

(1) We introduce WSD first and mention relation between concept and context, ***which is just like a continuation in computing environment***.

(2) We investigate related work about WSD and mention **context appropriateness** and **concept fitness**.

(3) We explore context appropriateness and concept fitness formally.

(4) We describe problem formulations in WSD using the proposed concepts.

(5) We report data processing, experiment settings, and conducted experiments.

5.1 Introduction

Word Sense Disambiguation (WSD) is an important task that has gained great attention of many researchers for a long time. Because human always reuse same word to denote different meanings, it is natural to let a computer system to automatically recover the exact meaning in a given context. For example, word ***bank*** is reused to denote a concept *depository financial institution* in context "*he saved his money in the biggest bank*", and to denote concept *sloping land of water* in context "*he takes a walk on the river bank*". In the mentioned cases, a word sense denotes a specific meaning of a word, and the mission of a WSD system is to discover the denoted meaning for a word in a given context. It is obvious that if a WSD system can precisely recover the exact meaning for a word in a given context, this will be beneficial for many NLP applications, such as Machine Translation and Information Extraction. For

[13] We will use materials in this chapter in a paper.

instance, because *the biggest bank* in the example denotes a company, we may co-refer a company name to this bank in an Information Extraction task.

Many literatures dedicate to discuss WSD. Agirre and Edmonds (2006) edit a thorough book on WSD related issues, including the history of WSD, the word sense inventory approaches, evaluation methods and datasets, WSD algorithms, resources, and applications. Navigli (2009) gives a newer and shorter survey on WSD. It lists many applications of WSD, including Information Extraction, Information Extraction (IE), Machine Translation (MT), Content Analysis, Word Processing, Lexicography, and the Semantic Web.

Many studies (Carpuat & Wu, 2005; Sanderson, 1994; Stokoe, Oakes, & Tait, 2003; Zhong & Ng, 2012) focus on discussing the usefulness of WSD in IR systems. Zhong and Ng (2012) conduct their experiments in standard TREC collections and conclude that supervised WSD system can significantly improve the performance of a state-of-the-art IR system. Other studies (Carpuat & Wu, 2005, 2007; Chan & Ng, 2007) show that WSD can improve the performance of MT systems. Carpuat and Wu (2007) demonstrate a very promising results on Chinese-English machine translation task. They find that WSD systems can improve phrase-based statistical MT models in many metrics such as BLEU. Chan and Ng (2007) also show that WSD system significantly improves the performance of MT systems.

In Information Extraction, WSD appears in a richer linguistic phenomenon. For example, traditional WSD concerns homonym, which has same word form and different meanings in different contexts. For IE researchers, they also consider synonym (different word forms but same meaning or same denotation in an entity) and metonymy. Markert and Nissim (2007) organized a metonymy resolution task in SemEval-2007. This task tries to make a distinction of BMW in context “my BMW runs fast”, which refers to a transportation vehicle and not a denotation to the famous automobile company. In biomedical domain, WSD plays an important role for automatic biomedical literature analysis (Schuemie, Kors, & Mons, 2005).

WSD improves the accuracy of literature understanding and improves the identification of ambiguous entities. Stevenson and Guo (2010) study three types of ambiguous terms in biomedical documents including ambiguous terms, ambiguous abbreviations and ambiguous gene names. Their systems reach very high performance ranging from 87.9% to 99.0%. Dai, Tsai, and Hsu (2011) study Entity Linking (EL) task, which links different mentions (synonyms) in biomedical literature to database entries to help document analysis.

Standard performance evaluation of WSD algorithms comes from Senseval, which is a series of competitions related to NLP tasks. After Senseval-1 in 1998, Senseval-2 (Edmonds & Cotton, 2001) formulates two WSD tasks: the lexical sample WSD and all-words WSD. Lexical sample WSD decides a word sense of a single word in a given fragment of text which usually contains many sentences. The all-words WSD decides senses of multiple words in the same time. These datasets contain many human-annotated examples in many languages and in different settings. Many WSD systems adopt datasets in Senseval-2 and Senseval-3 (R. Mihalcea, Chklovski, & Kilgarriff, 2004) for evaluating their WSD systems. With the standard evaluation datasets, researchers can give more meaningful performance comparisons between different WSD systems.

In the evaluation, researchers usually adopt WordNet (Fellbaum, 1998) as the sense inventory, which defines a closed set of senses for each word. In this situation, a word sense refers to a sense key in WordNet, and WSD problem is considered a classification problem because we want to decide the exact sense in the closed set. Kilgarriff (2006) gives a good survey on word sense. Some researchers did not use WordNet for sense inventory. They either use other ontologies or create their own sense clusters for sense inventory (Pantel & Lin, 2002). In these cases, comparisons of system performances are not easy, but the systems can have domain-specific sense inventory for their study.

Some researchers (Erk, McCarthy, & Gaylord, 2009; Erk & McCarthy, 2009; McCarthy, Koeling, Weeds, & Carroll, 2004) study different perspectives of WSD. Because the most common sense is very useful in WSD systems, McCarthy et al. (2004) use unsupervised methods to find predominant word senses in text. Erk et al. (2009) investigate the word usages and word sense, and build datasets with graded senses in a given contexts. This setting is different from traditional WSD task which usually select the best-fit sense in a given context. They (Erk & McCarthy, 2009) propose many metrics to evaluate sense grading system and implement system for this tasks.

In this study, we explore a more general problem which concerns the relation between concepts and contexts. We consider two aspects of this relation: **context appropriateness** and **concept fitness**. The context appropriateness is a function of modeling the appropriateness of contexts for a concept. For example, if we consider concept *depository financial institution* for word *bank*, context “*he saved his money in the biggest bank*” is appropriate but context “*he takes a walk on the river bank*” is inappropriate. On the other hand, the concept fitness is a function of modeling the fitness of concepts in a context. For example, if we consider context “*he saved his money in the biggest bank*”, concept *depository financial institution* is more fit than concept *sloping land of water* for word bank. It is obvious that WSD concerns concept fitness problem, while concept appropriateness problem is considered in knowledge extraction literature (Chklovski & Gil, 2005; Etzioni et al., 2008; Schwartz & Gomez, 2009; Singh et al., 2002). For example, if we consider concept *IS-A*, knowledge extraction researchers want to judge the appropriateness of the extraction (*IS-A*, a car, a vehicle) in context “a car is a vehicle usually driven by an engine of sorts”. But in context “Toyota Rent a Car is a vehicle rental system”, the extraction (*IS-A*, a car, a vehicle) is inappropriate but the extraction (*IS-A*, Toyota Rent a Car, a vehicle rental system) is good. In this case, concept *IS-A* is fixed but contexts (a car, a vehicle) and (Toyota Rent a Car, a vehicle rental system) are varying, which is directly opposite to WSD case. Knowledge extraction researchers try to find a good way to model the appropriateness of context to extract vast knowledge from free text for alleviating knowledge acquisition bottleneck. In this study, we explore these two aspects of the relation in the same time to see if it is helpful for WSD task.

When we interpret the relation between context and concept using our concept representation scheme, we can find that machine may learn models from perspectives of concept and context. The context (sentences) is the environment that a continuation (learned WSD models) lives. That is to say we can elicit information from different viewpoints. In this viewpoint, we can learn different models using different perspectives and combine the resulting models for WSD. We can also combine all information from different perspectives to learn a single model. We adopt the later approach for WSD.

5.2 Related Work

In literature, WSD systems study many aspects of the problem. In this section, we describe related work in WSD evaluation methods, algorithms, knowledge sources, feature processing, and training data utilization approaches. Although WSD are studied in many languages, we focus on English only in this study.

In WSD evaluation, researchers not only evaluate different problem settings like all-words and lexical sample (Palmer, Fellbaum, Cotton, Delfs, & Dang, 2001), but they evaluate system results in different granularities of sense. Because defining a clear cut on word sense is not easy, researchers evaluate systems in fine-grained and coarse-grained settings (R. Mihalcea et al., 2004; Navigli, Litkowski, & Hargraves, 2007; Snyder & Palmer, 2004). Generally, if the coarse-grained sense inventory is adopted, systems usually have higher performances. For example, in SemEval-2007 coarse-grained English all-words task, many systems can reach 87-88% F-measure. But if fine-grained sense inventory are used, the performance is in 65% accuracy (Snyder & Palmer, 2004). Therefore, some researchers think that graded sense (Erk & McCarthy, 2009) is more natural for sense assignment.

WSD algorithms can be roughly categorized into supervised, semi-supervised unsupervised algorithms. Supervised WSD algorithms usually have better performances. Researchers studied many supervised machine learning algorithms including exemplar-based algorithms (Escudero, Màrquez, & Rigau, 2000; Ng & Lee, 1996), neural networks (Towell & Voorhees, 1998), support vector machines (Lee, Ng, & Chia, 2004; Lee & Ng, 2002), multi-task learning (Ando, 2006), transfer learning (Dhillon, Foster, & Ungar, 2011; Dhillon & Ungar, 2009), and ensemble methods (Florian & Yarowsky, 2002). In SensEval-2 SensEval-3 lexical sample tasks, SVMs achieves good performance on these two datasets (Lee et al., 2004; Lee & Ng, 2002), which is 65.4% and 72.4% micro-averaged fine-grained recall, respectively. Ando (2006) proposes Alternating Structure Optimization (ASO) algorithm to learn latent structures that assume to be shared by different words. He reports that ASOs are significantly better than SVMs in SensEval-2 and SensEval-3 lexical sample datasets. Dhillon, Foster and Ungar (2011) use transfer learning to select features using Minimum Description Length (MDL) principle, and report that their method TransFeat is significantly better than ASO in SensEval-2 dataset.

Although supervised algorithms have better WSD performance, labeling sufficient training data is time-consuming. Researchers usually select a small set of words to label word senses for training and evaluation, and the learned models using supervised algorithms are only for these words. Semi-supervised can reduce the effort of labeling, and, sometimes, improve learning performance when combines with more labeled data (Ando, 2006). On the other hand, unsupervised algorithms do not need labeled data and can be used to learn WSD models for any words (Gonzalo & Verdejo, 2006). Some unsupervised approaches (Agirre & Martinez, 2004; Martinez, de Lacalle, & Agirre, 2008; R. F. Mihalcea, 2002; R. Mihalcea & Moldovan, 1999; Stevenson, Guo, & Gaizauskas, 2008) use heuristics to construct sense tagged data by utilizing search engines. For example, Martinez et al. (2008) construct queries for each polysemous noun in WordNet, submit queries to search engine, parse snippets to extract examples for a noun, and assign a word sense to the extracted example. They collect 150 million examples from Web, conduct experiments, and report promising results. Navigli and Lapata (2010) experiment unsupervised graph-based algorithm for all-word task. In graph-based algorithm, word senses are graph nodes, and relations between word senses are graph edges. Researchers usually use different relations in WordNet such as hypernymy semantic relation to link graph nodes (word senses). After building a graph for testing sentences, graph-based algorithms determine the importance of nodes and select most important nodes as the word senses. Navigli and Lapata (2010) investigate several graph connectivity measures to choose best senses for all testing words. Their experiments show that graph-based approaches are very useful for unsupervised WSD.

In supervised WSD, an example is represented by features, and features are extracted from different knowledge sources (Agirre & Stevenson, 2006). Agirre and Stevenson (2006) survey and list a complete knowledge sources used in WSD-related articles. In their survey, the knowledge sources can be syntactic, semantic, or topical. For example, a system can use dictionary as syntactic knowledge source to extract part-of-speech (POS) information of a word. With different knowledge sources, WSD systems extract many types of features in a given context of a word to resolve word sense. For example, Lee and Ng (2002) extract four types features from different knowledge sources: POS of neighboring words, bag-of-words in the surrounding context, local collocations, and syntactic relations. They use 0/1 encoding for all features and experiment many supervised machine learning algorithms. In their experiments, SVMs have the best results.

After extracted features, how to process feature values is an important consideration in supervised machine learning algorithms. Researchers may adopt simplest 0/1 encoding to represent features, in which 1 means that feature appears and 0 means otherwise. Researchers may use real number to represent some features like word frequency. In this case, scaling the feature to range [-1, 1] may result in a better performance. Because different feature types may have very different feature sizes, how to deal with this problem is also important. For example, if context features are used, the feature size may in the order of thousands features. But if word POS feature is used, the feature size is usually smaller than two hundred. Researchers may normalize one feature type to unit vector before combining different feature types to re-balance difference of feature sizes.

Meaning composition (Erk & Padó, 2008; Mitchell & Lapata, 2008; Thater et al., 2010) is an important feature processing approach in natural language processing, but it has not gained enough attention by WSD researchers. Meaning composition is a fundamental problem for meaning representation languages (Jurafsky & Martin, 2009b) and is crucial when researchers want to represent complex structures such as phrases and sentences in NLP applications. If researchers adopt first-order logic to represent meaning, meaning composition is embedded in the reference process. For example, meaning of a phrase is derived by composing the meaning of its individual words. When connectionist representations are adopted, meaning composition is usually considered a function that takes the meaning of components as input. Mitchell and Lapata (2008) explore many composition functions for cases that use vector space models to represent word meaning. They find that simple composition functions such as add and multiply are useful in judging word similarity. For example, Kintsch (2001) proposes that word senses are modified by context. For example, if word ran is modified in The color ran, its meaning is closer to dissolve instead of gallop in The horse ran. In WSD, researchers usually use horse and color as collocation or context information, and they did not consider the relation between meaning composition and WSD. In this paper, we will consider this issue.

In supervised WSD algorithms, how to maximally utilize the scarce training data is an important research topic. Ando (2006) uses multi-tasks learning to discover shared latent structures between different words. Dhillon, Foster and Ungar (2011) propose TransFeat algorithms to select features across different words. These two approaches demonstrate the approaches that best utilizing the scarce data in hand. In this view point, semi-supervised is also of this type because this approach extents information in labeled data to un-label data in the hope that there are more useful information can be captured by using semi-supervised approaches.

We will investigate different ways to enlarge training set by using meaning composition and utilizing context appropriateness and context fitness. We explain our approaches later.

5.3 Context Appropriateness and Concept Fitness

In this section, we will explore context appropriateness and concept fitness more formally, and prepare notations that will be used later.

Suppose we have a set of concepts[14] $\{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_\mathrm{n}\}$. If we consider the relation between concepts and contexts, we get **context appropriateness** by fixing concept. In other words, we want to know the appropriateness of a specific context for a given concept. For example, if concept *bank*$_{\textit{financial institution}}$ is given, context t_1="*he saved his money in the biggest bank*" is appropriate but context t_2= "*he takes a walk on the river bank*" is inappropriate. Please note that although in WSD case, the set of concepts may be for senses of a word, but this formulation did not be restricted to WSD case only.

Now, we define context appropriateness to be a real-valued function $f^{appr.}$ that can correctly rank the appropriateness of a context given a concept s_i in Equation 2.

[14] In this chapter, we consider concept and word sense to be the same.

$$f^{appr.}(\mathrm{s_i}, \mathrm{t_{i,j}}) > f^{appr.}(\mathrm{s_i}, \hat{\mathrm{t}}_{\mathrm{i,k}}) \qquad \text{Equation 2}$$

If we category contexts into two levels, we can say that

$\mathrm{t_{i,j}} \in \{\text{all contexts appropriate for concept } \mathrm{s_i}\} = \mathrm{T_i}$, and

$\hat{\mathrm{t}}_{\mathrm{i,k}} \in \{\text{all contexts not appropriate for concept } \mathrm{s_i}\} = \hat{\mathrm{T}}_{\mathrm{i}}$,

Equation 2 just maintains an order between the appropriateness of contexts. In simplest case, we use $\mathrm{T_i}$ and $\hat{\mathrm{T}}_{\mathrm{i}}$ to denote the set of contexts that appropriate and not appropriate for concept s_i, respectively. In the mentioned example, context t_1 belongs to $\mathrm{T_i}$ and t_2 belongs to $\hat{\mathrm{T}}_{\mathrm{i}}$ for concept s_i = *bank* *financial institution*. This kind of formulation is motivated by Kintsch's (2001) and Mitchell and Lapata (2008) in measuring sentence similarity after meaning composition. Their idea is that a good meaning composition model should result in a new vector that closer to vectors with similar meaning. When applying to concept and its context, we want machine can learn a good context appropriateness function that gives higher scores to appropriate contexts than inappropriate contexts for a concept. In Equation 2, meaning composition is important but not necessary. If there is enough information to judge the score, meaning composition is not necessary. But meaning composition gives us a new way to process features in WSD problems. We can use the composed feature vector for machine learning algorithms instead of the raw features for concept and context. This viewpoint is new in WSD, and we will illustrate approaches later to utilize this feature processing approach to enlarge the size of training data.

In knowledge extraction literatures, as we mentioned earlier, context appropriateness function is used to judge the reliability of extracted knowledge. Knowledge extraction researchers want to extract knowledge from free text in tuple format (relation, argument 1, argument 2) such as extracting (*IS-A*, a car, a vehicle) from sentence "*a car is a vehicle usually driven by an engine of sorts*". In this case, concept is *IS-A* relation, and meaning composition may take place between two arguments "a car" and "a vehicle". We do not restrict in how meaning composition is applied in Equation 2, and we will demonstrate how meaning composition work in WSD.

If meaning composition is used, we use $f^{appr.}(\mathrm{s_i}, \mathrm{t_{i,j}}) = f^{appr.}(\Phi(\mathrm{s_i}, \mathrm{t_{i,j}}))$ to denote it, where $\Phi(\mathrm{s_i}, \mathrm{t_{i,j}})$ is a vector and, and Φ is a meaning composition function for context appropriateness. We will use $f^{appr.}(\mathrm{s_i}, \mathrm{t_{i,j}})$ to denote functions without and with meaning composition hereafter for simplicity.

If we consider the relation between concepts and contexts, we get **concept fitness** by fixing context. We want to disambiguate the precise concept (or word sense) in a specific context. For example, concept s_i = *bank*$_{\textit{financial institution}}$.is fitter than concept s_k = *bank*$_{\textit{sloping land of water}}$ in a given context t_1="*he saved his money in the biggest bank*". This is exactly a WSD problem, but we use a more general viewpoint to re-consider it. We will find that we can reformulate WSD in many ways if we adopt this more general viewpoint. In addition, this viewpoint can unify different WSD settings in a single viewpoint such as standard WSD and graded word sense problems (Erk & McCarthy, 2009).

Like context appropriateness, we define concept fitness to be a real-valued function $f^{fit.}$ that can correctly rank the fitness of concepts given a context t_j in Equation 3.

$$f^{fit.}(\mathrm{s_{i,j}}, \mathrm{t_j}) > f^{fit.}(\hat{\mathrm{s}}_{\mathrm{k,j}}, \mathrm{t_j}) \qquad \text{Equation 3}$$

If we category contexts into two levels, we can say that

$\mathrm{s_{i,j}} \in \{\text{all concepts fit for context } \mathrm{t_j}\} = \mathrm{S_j}$, and

$\hat{\mathrm{s}}_{\mathrm{k,j}} \in \{\text{all concepts not fit for context } \mathrm{t_j}\} = \hat{\mathrm{S}}_{\mathrm{j}}$,

Equation 3 also just maintains an order between the fitness of concepts. In simplest case, we use $\mathrm{S_j}$ and $\hat{\mathrm{S}}_{\mathrm{j}}$ to denote the set of concepts that fit and not fit for context $\mathrm{t_j}$, respectively. In standard WSD setting, $\mathrm{t_j}$ is a context to be disambiguated, $\mathrm{S_j}$ usually contains one word sense or multiple word senses, and $\hat{\mathrm{S}}_{\mathrm{j}}$ contains other word senses that not fit in context $\mathrm{t_j}$. In graded word sense problem (Erk & McCarthy, 2009), function $f^{fit.}$ returns different scores for different word senses. Therefore, concept fitness is more general than standard WSD problems. Like context appropriateness, meaning composition can be adopted in a useful way.

If meaning composition is used, we use $f^{fit.}\left(\mathrm{s}_{\mathrm{i,j}}, \mathrm{t}_{\mathrm{j}}\right) = f^{fit.}\left(\Phi(\mathrm{s}_{\mathrm{i,j}}, \mathrm{t}_{\mathrm{j}})\right)$ to denote it, where $\Phi\left(\mathrm{s}_{\mathrm{i,j}}, \mathrm{t}_{\mathrm{j}}\right)$ is a vector, and Φ is a meaning composition function for concept fitness. We will use $f^{fit.}\left(\mathrm{s}_{\mathrm{i,j}}, \mathrm{t}_{\mathrm{j}}\right)$ to denote functions without and with meaning composition hereafter for simplicity.

Now, we jointly consider context appropriateness and concept fitness. We can derive equations Equation 4 to Equation 6 when different constraints are adopted. If we use two levels (fit and not-fit) for context appropriateness and concept fitness, we get Equation 4 and Equation 5, where $\mathrm{s}_{\mathrm{i}} \in \mathrm{S}_{\mathrm{j}}$, $\mathrm{t}_{\mathrm{j}} \in \mathrm{T}_{\mathrm{i}}$, $\hat{\mathrm{s}}_{\mathrm{k}} \in \hat{\mathrm{S}}_{\mathrm{j}}$, and $\hat{\mathrm{t}}_{\mathrm{k}} \in \hat{\mathrm{T}}_{\mathrm{i}}$.

$$f^{appr.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) + f^{fit.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) > f^{appr.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) + f^{fit.}\left(\hat{\mathrm{s}}_{\mathrm{k}}, \mathrm{t}_{\mathrm{j}}\right) \qquad \text{Equation 4}$$

$$f^{appr.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) + f^{fit.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) > f^{appr.}(\mathrm{s}_{\mathrm{i}}, \hat{\mathrm{t}}_{\mathrm{k}}) + f^{fit.}\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) \qquad \text{Equation 5}$$

Equation 4 and Equation 5 can be derived from definitions of Equation 3 and Equation 2, respectively. They suggest that context appropriateness and concept fitness provide different viewpoints to judge the validness of combining a concept and a context. If we careful choose different representation schemes for these two viewpoints, we can generate more training data for supervised algorithms. For example, if there are 5 senses $\{\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5\}$ for a word and there is a context t_1 with only one correct concept $\mathrm{s}_1 \in \mathrm{T}_1 = \{\mathrm{s}_1\}$, the concept that not fit for context t_1 are in $\hat{\mathrm{T}}_1 = \{\mathrm{s}_2, \mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5\}$, and then we can generate 4 training instances by using Equation 4. These training instances are showed in the followings.

$$f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_1},\mathrm{t_1}) > f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_2},\mathrm{t_1})$$
$$f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_1},\mathrm{t_1}) > f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_3},\mathrm{t_1})$$
$$f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_1},\mathrm{t_1}) > f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_4},\mathrm{t_1})$$
$$f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_1},\mathrm{t_1}) > f^{appr.}(\mathrm{s_1},\mathrm{t_1}) + f^{fit.}(\mathrm{s_5},\mathrm{t_1})$$

Equation 5 can be used in the same way for contexts.

If we further apply Equation 2 and Equation 3 in Equation 4 and Equation 5, we get Equation 6 in the below, in which $\hat{\mathrm{t}}_\mathrm{l} \in \widehat{\mathrm{T}}_\mathrm{i}$ is a context that inappropriate for concept $\mathrm{s_i}$.

$$f^{appr.}\left(\mathrm{s_i},\mathrm{t_j}\right) + f^{fit.}\left(\hat{\mathrm{s}}_\mathrm{k},\mathrm{t_j}\right) > f^{appr.}(\mathrm{s_i},\hat{\mathrm{t}}_\mathrm{l}) + f^{fit.}\left(\hat{\mathrm{s}}_\mathrm{k},\mathrm{t_j}\right) \qquad \text{Equation 6}$$
$$f^{appr.}(\mathrm{s_i},\hat{\mathrm{t}}_\mathrm{l}) + f^{fit.}\left(\mathrm{s_i},\mathrm{t_j}\right) > f^{appr.}(\mathrm{s_i},\hat{\mathrm{t}}_\mathrm{l}) + f^{fit.}\left(\hat{\mathrm{s}}_\mathrm{k},\mathrm{t_j}\right)$$

Equation 6 just says that a learned function must return larger score of one error than that of two errors. We can use a simple score assignment to adopt those equations in WSD problem. For example, we can assign left hand side of Equation 4 to have value 2, right hand sides of Equation 4 and Equation 5 to have value 1, and right hand side of Equation 6 to have value 0. In this way, we have more training data and more ways to utilize the information we have. We will illustrate approaches to construct training datasets by utilizing these equations in next section.

5.4 Problem Formulations in WSD

In this section, we illustrate different approaches to utilize context appropriateness and concept fitness along with the meaning composition feature processing approach. We will introduce baseline approach without meaning composition, multi-class classifications with meaning composition, binary classifications with meaning composition, and ranking approaches with meaning composition.

5.4.1 Multi-class Classification (Baseline)

In WSD, it is straightforward to formulate sense disambiguation of a word as a multi-class classification problem. Suppose we have a set of *n* word senses $S = \{s_1, s_2, \dots, s_n\}$ for word *w*, and we also have a set of *m* contexts $\{(s_i,\ t_j)\}_{j=1}^{m}$ that word *w* occurs, where $s_i \in S$ and $(s_i,\ t_j)$ denotes *j*-th context which its sense is s_i. The multi-class classifiers learn a function $f((s_i,\ t_j))$ which takes a context as input and predicts the correct sense s_i. In most WSD settings, s_i is ignored because word *w* is same for all contexts. In this case, classifiers learn function $f((t_j)) = f(t_j)$ instead of $f((s_i,\ t_j))$. If classifiers learn function $f(t_j)$ for a word with dataset $\{(s_i,\ t_j)\}_{j=1}^{m}$, we denote this kind of formulation **M**ulti-**C**lass **w**ith**o**ut **M**eaning **C**omposition (MCwoMC).

In this formulation, the decision function depends on classifiers we adopt. Researchers usually use one-vs-the-rest multi-class strategy to train a binary classifier for a sense, and ensemble many binary classifiers to produce final prediction. For example, researchers (Lee et al., 2004; Lee & Ng, 2002) train a binary SVM classifier for each word sense and output sense s_i with highest prediction score. We show the sense decision function in Equation 7, where function f_i is a binary classifier for sense s_i using one-vs-the-rest multi-class strategy, and *i* = 1...*n*.

$$s_i = \mathrm{argmax}_i\, f_i(t_j) \qquad \text{Equation 7}$$

5.4.2 Multi-class Classification with Meaning Composition

If meaning composition is used, $f((s_i,\ t_j))$ cannot be simplified into $f((t_j)) = f(t_j)$ because different word senses will result in different meanings. With meaning composition, *j*-th context can compose with different word senses and classifiers must give different predictions. For example, although the disambiguating word is the same for all contexts, we have $\mathrm{f}\big(\Phi(\mathrm{s}_1,\ \mathrm{t}_\mathrm{j})\big) \neq \mathrm{f}\big(\Phi(\mathrm{s}_2,\ \mathrm{t}_\mathrm{j})\big)$, in which Φ is a meaning composition function for concept s_i and its context t_j. If classifiers learn function $\mathrm{f}\big(\Phi(\mathrm{s}_\mathrm{i},\ \mathrm{t}_\mathrm{j})\big)$ for a word with dataset $\big\{\Phi\big(\mathrm{s}_\mathrm{i},\ \mathrm{t}_\mathrm{j}\ \big)\big\}_{\mathrm{j}=1}^{\mathrm{m}}$, we denote this kind of formulation **M**ulti-**C**lass **w**ith **M**eaning **C**omposition (MCwMC).

In this formulation, the decision function must be carefully handled because the class information is encoded in training features though meaning composition. In our study, we find that if we use decision function like that of MCwoMC, the performances using different types of features can reach 92% accuracy which is wrong[15]. To have a correct decision function, we must test every possible meaning compositions of word sense and choose the highest score as predicted sense. We show the sense decision function in Equation 8, where f is decision function returned by classifier and sense score decision function g is a function of assigning a score to value $\mathrm{f}\big(\Phi(\mathrm{s}_\mathrm{i},\ \mathrm{t}_\mathrm{j})\big)$.

$$\mathrm{s}_\mathrm{i} = \mathrm{argmax}_\mathrm{i}\, \mathrm{g}\Big(\mathrm{f}\big(\Phi(\mathrm{s}_\mathrm{i},\ \mathrm{t}_\mathrm{j})\big)\Big) \qquad \text{Equation 8}$$

For example, if we use Support Vector Regression (SVR) as classifier, and let class $\mathrm{s}_\mathrm{i} = \mathrm{i}$ for value assignment. The regression function $\mathrm{f}\big(\Phi(\mathrm{s}_\mathrm{i},\ \mathrm{t}_\mathrm{j})\big)$ returns a regression value v_i, and sense score decision function can adopt $\mathrm{g}(\mathrm{v}_\mathrm{i}) = 1 - |\mathrm{i} - \mathrm{v}_\mathrm{i}|$. This means we measure the distance between class's value and regression value and choose closest one as sense assignment. We denote this approach to be MCwMC-Reg.

[15] In this case, it means that classifier can make a clear distinguish between concepts but not concept in specific context. The reason is that concept's features are derived by summing all contexts of a sense.

If we use many binary SVM classifiers for $f(\Phi(s_i,\ t_j))$ like that of MCwoMC, sense score decision function is the predicted value of that class s_i. For example, if we have 3 senses $\{s_1, s_2, s_3\}$, we must test three vectors $\Phi(s_1,\ t_j)$, $\Phi(s_2,\ t_j)$, and $\Phi(s_3,\ t_j)$ which represent three different meaning composition using different senses. A multi-class classifiers may output three scores of probability (score of s_1, score of s_2, score of s_3) for a test case. If test case $\Phi(s_1,\ t_j)$ has scores (0.95, 0.01, 0.11), we let $g\left(f\left(\Phi\left(s_1,\ t_j\right)\right)\right)$ = 0.95. If test case $\Phi(s_2,\ t_j)$ has scores (0.05, 0.89, 0.21), we let $g\left(f\left(\Phi\left(s_2,\ t_j\right)\right)\right)$= 0.89. If test case $\Phi(s_3,\ t_j)$ has scores (0.16, 0.05, 0.76), we let $g\left(f\left(\Phi\left(s_3,\ t_j\right)\right)\right)$= 0.76. In this example, sense s_1 is the predicted sense because it has highest value (probability). We denote this approach to be MCwMC-SVM.

5.4.3 Binary Classification with Meaning Composition

We can simplify multi-class classification problem to a single binary classification problem by using meaning composition. The idea is simple: if the composition $\Phi(s_i,\ t_j)$ is valid, we assign 1 to it, and assign -1 otherwise. The resulting WSD dataset is $\bigcup_{j=1}^{m} \left\{\left(\Phi\left(s_i,\ t_j\right), y_{i,j}\right)\right\}_{i=1}^{n}$, where $y_{i,j} \in \{+1, -1\}$. We denote this kind of formulation **B**inary **C**lassification **w**ith **M**eaning **C**omposition (BCwMC).

This formulation has two advantages. First, it is simple and does not need to train many classifiers for multiple classes. Second, there are more training samples than that of original multi-class problem. For example, if there are 100 contexts and 10 senses for a word, we will have 1000 feature vectors in dataset.

This formulation comes with two disadvantages. First, it may result in an unbalanced dataset if there are a lot of senses of a word. For example, in SensEval-2 lexical sample task, word art.n (art with noun POS) has 19 senses. This means positive sample is only 5.3% in whole dataset. It is not easy to train a good classifier in this case. Second, the resulting feature matrix may be a dense matrix which is not good if we want to handle large problems. In our study, the ratio of non-zero count of feature matrix using meaning composition may reach 2~30%.

Like MCwMC, for a context t_j, we have multiple vectors which compose meaning with different senses. If we use binary classifier with output of probability, we can use sense decision function like Equation 8. We denote this approach to be BCwMC-SVM.

If we use regression, the sense score decision function is like that of MCwMC-Reg. But the classes are +1 and -1 in BCwMC. We denote this approach to be BCwMC-Reg.

5.4.4 Ranking 2-Level with Meaning Composition

Unlike BCwMC which uses single binary classification for a word's disambiguation, we now try to formulate WSD as a ranking problem. In BCwMC formulation, if composition $\Phi(s_i, t_j)$ is valid, we assign +1 to it, and assign -1 otherwise. But if we want to learn a linear function that can rank all valid compositions before invalid compositions, we can formulate WSD using ranking (Cohen, Schapire, & Singer, 1999; Joachims, 2002).

Ranking is a common technique used to order things. One of the most important applications of ranking is ranking results of search engines (T.-Y. Liu, 2009). In WSD with meaning composition, if we can rank valid compositions in high performance, we have a good model for WSD. The ranking formulation of WSD is shown in Equation 9 which is similar to Ranking SVM formulation (Joachims, 2002).

$$\text{minimize} \quad \mathrm{V}(\mathrm{w}, \xi) = \frac{1}{2} \|\mathrm{w}\|_2^2 + \mathrm{C} \sum \xi_{\mathrm{i,j,k}} \qquad \text{Equation 9}$$

subject to:

$$\forall \mathrm{j} \forall \hat{\mathrm{s}}_{\mathrm{k}} \in \hat{\mathrm{S}}_1 \quad \mathrm{w}\Psi\left(\mathrm{s}_1, \mathrm{t}_{\mathrm{j}}\right) \geq \mathrm{w}\Psi\left(\hat{\mathrm{s}}_{\mathrm{k}}, \mathrm{t}_{\mathrm{j}}\right) + 1 - \xi_{1,\mathrm{j,k}}$$

...

$$\forall \mathrm{j} \forall \hat{\mathrm{s}}_{\mathrm{k}} \in \hat{\mathrm{S}}_{\mathrm{n}} \quad \mathrm{w}\Psi\left(\mathrm{s}_{\mathrm{n}}, \mathrm{t}_{\mathrm{j}}\right) \geq \mathrm{w}\Psi\left(\hat{\mathrm{s}}_{\mathrm{k}}, \mathrm{t}_{\mathrm{j}}\right) + 1 - \xi_{\mathrm{n,j,k}}$$

$$\forall \mathrm{i} \forall \mathrm{j} \forall \mathrm{k} \quad \xi_{\mathrm{i,j,k}} \geq 0 \,,$$

$$\text{where } \Psi\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right) = \left(\Phi\left(\mathrm{s}_{\mathrm{i}}, \mathrm{t}_{\mathrm{j}}\right)\right)$$

In Equation 9, w is the learned linear function, $\xi_{\mathrm{i,j,k}}$ is slack variable, s_{i} is a sense, and t_{j} is a context. We can assign relevance score of valid compositions to be 2 (relevant) and that of invalid compositions to be 1 (non-relevant). In this setting, a sense is like a query, and we retrieve valid meaning compositions to be relevant documents of that sense.

The ranking formulation is similar to formulation of binary classification, and we show if in Equation 10. We denote this approach to be **R**anking **2**-Levels with **M**eaning **C**omposition (R2wMC-Ses), in which “Ses” means one **se**n**s**e one query.

$$\mathrm{s}_{\mathrm{i}} = \mathrm{argmax}_{\mathrm{i}}\, \mathrm{w}\Phi(\mathrm{s}_{\mathrm{i}},\ \mathrm{t}_{\mathrm{j}}) \qquad \text{Equation 10}$$

In R2wMC-Ses formulation, the constraints are hold inside a sense. We can enforce all constraints to be hold for a word. In this case, we only have one query for a word. We use the same formulation except every sense is in one query. We denote this approach to be R2wMC-Wrd, in which “Wrd” means one **w**o**rd** one query.

5.4.5 Ranking 3-Level with Meaning Composition

Now, we utilize context appropriateness and concept fitness using Equation 4 to Equation 6 in this section. Equation 4 to Equation 6 can be adopted to generate more training data when we assign different relevance scores to different situations. This is equivalently to direct adding Equation 4 to Equation 6 as constraints of Equation 9. We show the formulation in Eq. 10 below.

$$
\begin{aligned}
&\text{minimize} \quad \mathrm{V}(\mathrm{w},\xi) = \tfrac{1}{2}\|\mathrm{w}\|_2^2 + \mathrm{C}\sum \xi_{\mathrm{i,j,k,l}} \\
&\text{subject to:} \\
&\forall \mathrm{j} \forall \hat{\mathrm{t}}_\mathrm{k} \in \widehat{\mathrm{T}}_\mathrm{j} \quad \mathrm{w}\Psi(\mathrm{s_i}, \mathrm{t_j}) \geq \mathrm{w}\Psi^{appr.}(\mathrm{s_i}, \mathrm{t_j}, \hat{\mathrm{t}}_\mathrm{k}) + 1 - \xi_{\mathrm{i,j,k,1}} \\
&\forall \mathrm{j} \forall \hat{\mathrm{s}}_\mathrm{k} \in \hat{\mathrm{S}}_\mathrm{i} \quad \mathrm{w}\Psi(\mathrm{s_i}, \mathrm{t_j}) \geq \mathrm{w}\Psi^{fit.}(\mathrm{s_i}, \hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j}) + 1 - \xi_{\mathrm{i,j,k,2}} \\
&\forall \mathrm{j} \forall \hat{\mathrm{t}}_\mathrm{k} \in \widehat{\mathrm{T}}_\mathrm{j} \quad \mathrm{w}\Psi^{appr.}(\mathrm{s_i}, \mathrm{t_j}, \hat{\mathrm{t}}_\mathrm{k}) \geq \mathrm{w}\Psi(\mathrm{s_i}, \hat{\mathrm{t}}_\mathrm{k}) + 1 - \xi_{\mathrm{i,j,k,3}} \\
&\forall \mathrm{j} \forall \hat{\mathrm{s}}_\mathrm{k} \in \hat{\mathrm{S}}_\mathrm{i} \quad \mathrm{w}\Psi^{fit.}(\mathrm{s_i}, \hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j}) \geq \mathrm{w}\Psi(\hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j}) + 1 - \xi_{\mathrm{i,j,k,4}} \\
&\mathrm{i} = 1 \ldots \mathrm{n} \\
&\forall \mathrm{i} \forall \mathrm{j} \forall \mathrm{k} \forall \mathrm{l} \quad \xi_{\mathrm{i,j,k,l}} \geq 0\,, \\
&\text{where } \Psi(\mathrm{s_i}, \mathrm{t_j}) = \left(\Phi^{appr.}(\mathrm{s_i}, \mathrm{t_j}), \Phi^{fit.}(\mathrm{s_i}, \mathrm{t_j})\right), \\
&\qquad \Psi^{appr.}(\mathrm{s_i}, \mathrm{t_j}, \hat{\mathrm{t}}_\mathrm{k}) = \left(\Phi^{appr.}(\mathrm{s_i}, \hat{\mathrm{t}}_\mathrm{k}), \Phi^{fit.}(\mathrm{s_i}, \mathrm{t_j})\right), \\
&\qquad \Psi^{fit.}(\mathrm{s_i}, \hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j}) = \left(\Phi^{appr.}(\mathrm{s_i}, \mathrm{t_j}), \Phi^{fit.}(\hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j})\right).
\end{aligned}
\qquad \text{Equation 11}
$$

Equation 11 is simple. If context and concept match each other, we give it the highest ranking score. If context appropriateness and concept fitness are all missed, we give it the lowest score. If one of context appropriateness and concept fitness is missed, we give it middle score. In relevant score assignment of WSD, we set $\Psi(\mathrm{s_i}, \mathrm{t_j})$ to 3 (most relevant),

$\Psi^{appr.}(\mathrm{s_i}, \mathrm{t_j}, \hat{\mathrm{t}}_\mathrm{k})$ and $\Psi^{fit.}(\mathrm{s_i}, \hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j})$ to 2 (relevant), and $\Psi(\mathrm{s_i}, \hat{\mathrm{t}}_\mathrm{k})$ and $\Psi(\hat{\mathrm{s}}_\mathrm{k}, \mathrm{t_j})$ to 1 (non-relevant). In this way, we generate many training samples for ranking algorithm even though the labeled dataset is small. We denote this approach to be **R**anking **3**-Levels with **M**eaning **C**omposition (R3wMC-Ses), in which "Ses" means one **se**n**s**e one query.

In R3wMC-Ses formulation, the constraints are hold inside a sense. We can enforce all constraints to be hold for a word. In this case, we only have one query for a word. We use the same formulation except every sense is in one query. We denote this approach to be R3wMC-Wrd, in which "Wrd" means one **w**o**rd** one query.

The sense decision functions of R3wMC-Ses and R3wMC-Wrd is the same. We use the equation in Equation 10 for sense decision function.

5.5 Feature Extraction and Experiment Settings

For a sample of SensEval's lexical sample task, we extract nine types of features. The first four types are commonly used by WSD researchers (Lee & Ng, 2002). The next five types of features are new. We describe them below.

1. Part-of-Speech of neighboring words (**POS**):

   We encode 7 POS features of the disambiguating word. A feature P_i (i=-3, -2, ..., 3) is a POS of neighboring word. P_0 refer to the disambiguating word. If there is no such word in a position, we use NIL to be the tag. Each feature is regarded as a bag-of-word feature, and we assign weight of feature using 0/1 encoding, which means if that word appear, we assign weight 1 to that word, and assign 0 if that word did not appear.

2. Words in context (**Context**):

   We lemmatize word unigram of surrounding context to WordNet 3.0 lemma and exclude stop words. The surrounding context includes all words in neighboring sentences. This is a single feature and we use bag-of-word 0/1 encoding.

3. Collocation (**Colloc**):

   We implement 11 collocations described in Lee and Ng's paper (Lee & Ng, 2002). There are 11 features and we use bag-of-word 0/1 encoding for each feature.

4. Syntactic relations of target disambiguating word (**SyntaxRel**):

   We implement syntactic relations described in Lee and Ng's paper (Lee & Ng, 2002). There are different types features for noun, verb, and adjective. We use bag-of-word 0/1 encoding for each feature.

5. Words in dependency relation (**drWord**):

   We parse sentence of target disambiguating word to get dependency relations (de Marneffe, MacCartney, & Manning, 2006; Toutanova et al., 2003). For a list of tuples of dependency relations (grammatical relation, governor, dependent), there is a sub-list R that the target disambiguating word are in the relations. We add all words in R (in governor or in dependent) for this feature except the target disambiguating word itself. This is a single feature and we use bag-of-word 0/1 encoding.

6. Grammatical relation in dependency relation (**drRelt**):

   We add all grammatical relation that in sub-list R that the target disambiguating word is in the relations. This is a single feature and we use bag-of-word 0/1 encoding.

7. Words in dependency relation with role information (**drRole**):

   We add all words with its role information that in sub-list R except disambiguating word itself. For example, for word *w*, we add string *w*_gov to indicate that word *w* is a governor in the dependency relation. Similarly, we add string *w*_gov to indicate its role is dependent. This is a single feature and we use bag-of-word 0/1 encoding.

8. Extension of words in dependency relation using WordNet definition (**drDefi**):

   For each word *w* in feature drWord, we add all word unigrams of word *w*'s definition.

If word *w* has multiple senses, we add all definitions in WordNet 3.0. The stop words are excluded. This is a single feature and we use bag-of-word 0/1 encoding.

9. Extension of words in dependency relation using WordNet synset (**drCnpt**):
   For each word *w* in feature drWord, we add all synset ids of word *w*. If word *w* has multiple senses, we add all synset ids in WordNet 3.0. This is a single feature and we use bag-of-word 0/1 encoding.

In meaning composition, concept's representation is build by summing all samples representation. Context, POS, and colloc are representation for context appropriateness. Other types of features are representations for concept fitness because these features are closely related to concept's meaning and are usually adopted in meaning composition in knowledge extraction. We combine add and multiplication operations for meaning composition (Mitchell & Lapata, 2008). For example, for feature vector *v* and *u*, the meaning composition function $\Phi(\mathrm{u},\ \mathrm{v}) = (\mathrm{u} + \mathrm{v}, \mathrm{uv})$, where uv is point wise multiplication.

We conduct experiments in lexical sample tasks in SensEval-2 and SensEval-3. There are 73 words and 57 words in SensEval-2 and SensEval-3 lexical sample tasks, respectively. These words are in three categories: noun, verb and adjective. There are total 8611 training samples and 4328 testing samples in SensEval-2, and 7860 and 3944 in SensEval-3. Some words have many senses but samples are little. For example, there are 43 senses for verb *turn* along with 131 training and 67 testing samples.

We use LibSVM (Chang & Lin, 2011) classifiers for multi-class and binary class classification, use Liblinear (Fan, Chang, Hsieh, Wang, & Lin, 2008) for regression function of MCwoMC-Reg, and use Ranking SVM (Joachims, 2002) for our ranking algorithms.

We adopt RBF kernel and perform a grid search for (c, g) in using LibSVM. We try 9 parameter C in using Liblinear. We use 3-fold cross validation for model selection in LibSVM and Liblinear.

We find that higher cost c of Ranking SVM usually resulting in better performance but taking more training time, and we fix cost to 10. We also set parameter -# 20000 for truncate long time training.

5.6 Experiment Results

First, we want to know the performance of each feature, and we show fine-grained summary results of SensEval-2 and SensEval-3 in the table below.

Features	SensEval-2		SensEval-3	
	Train	Test	Train	Test
Colloc	40.78	53.27	55.51	59.62
Context	42.50	53.93	64.23	61.95
drCnpt	55.47	50.86	64.86	57.37
drDefi	58.75	52.15	67.98	59.21
POS	58.50	54.82	66.22	59.26
SyntaxRel	61.79	53.04	68.03	57.19
drRelt	60.19	53.29	68.51	59.42
drRole	60.26	53.43	70.12	60.64
drWord	59.11	53.62	66.70	57.32
drDefi+drCnpt	55.63	52.03	65.81	59.26
drWord+drRelt+drRole	64.42	55.64	71.75	61.72
drWordvdrRelt+drRole+drDefi+drCnpt	65.55	56.86	68.17	61.62
SyntaxRel+POS+Context+Colloc (baseline)	56.20	57.54	66.48	**64.94**
baseline+drDefi+drCnpt	58.78	57.82	65.67	64.89
baseline +drWord+drReRelt+drRole	56.61	58.01	64.21	64.12
All features	59.69	**58.48**	63.72	64.00
(Lee & Ng, 2002): micro averaged recall on all words	n/a	65.4	n/a	n/a
(Ando, 2006)	n/a	65.3	n/a	74.1

Table 5. WSD results of MCwoMC using different features.

In Table 5, we can see that our performance is far behind the state of the art. We notice that our baseline is 57.54 which is still smaller than that of Lee and Ng (2002)'s system. But we use same types of features. One possible reason is that Lee and Ng (2002) train binary classifiers using different parameters while we use same parameter for all binary classifiers of a word.

Next, we want to know the performance of different problem formulations. We show results in the table below.

Problem Formulation	SensEval-2		SensEval-3	
	Train	Test	Train	Test
MCwoMC	59.69	58.48	66.48	64.94
BCwMC-Reg	44.01	35.57	39.57	32.45
BCwMC-SVM	85.98	47.06	92.75	59.80
MCwMC-Reg	45.53	36.81	45.50	47.29
MCwMC-SVM	53.03	48.07	56.99	52.10
R2wMC-Ses	97.78	**59.06**	95.79	**64.97**
R2wMC-Wrd	97.90	58.31	96.06	64.64
R3wMC-Ses	97.87	**59.06**	95.56	64.10
R3wMC-Wrd	97.99	58.22	97.45	64.59

Table 6. WSD results in different problem formulations.

In Table 6, R2wMC-Ses has best performance, but it is still smaller than the performance of state of the art. We can see that our models have better performances based on same experiment process. Especially, the training results are very high. Because the resulting models using ranking is linear, we think that this phenomenon is very useful if we integrate unsupervised algorithms with our methods.

There are many possible ways to improve our performance. One possible direction is using different approaches to derive concept representation. In our experiments, we sum all features of that sense to construct its representation. It may be better if we use unsupervised method to construct the representation. One possible direction is using dimension reduction to shorten the gap between training and testing. Approaches like principle component analysis (PCA) and feature selections may work for this case. We leave these issues for future work.

Chapter 6. Knowledge Sources

Many good sources can be adopted for researchers to extract knowledge for building concept representation of computer. One of most common sources is using web pages. Researchers already use web crawlers to download web pages and prepare a large dataset for researchers. We adopt ClueWeb09, a web-scale dataset, as knowledge source because it contains half billon of web pages in multiple languages. Although we did not extract knowledge from ClueWeb09, we want to prepare datasets for other researchers to boost research of knowledge extraction.

We organize materials in the order below.

(1) We shortly describe ClueWeb09 dataset.

(2) We want to know what kinds of knowledge can be extracted from the web pages. Therefore, we search commonsense knowledge in the web, and find that even this kind of knowledge can be found in the web. This asserts that web page is a good knowledge source.

(3) We describe the preprocessing of English web pages, the resulting dataset, and some statistical information.

(4) We describe the preprocessing of Chinese web pages, the resulting datasets, and some statistical information.

(5) Because we have build resources based on ClueWeb09 dataset. We want to know whether ClueWeb09 is sufficient large to be adopted in common NLP tasks. We compare the results with that of using Google web 5-gram dataset and find that the answer is positive. We also demonstrate the use of concept representation scheme here.

6.1 ClueWeb09 Dataset

In order to support information retrieval and natural language processing researches, researchers in University of Massachusetts, Amherst and Carnegie Mellon University begin a The Lemur Project. As part of results of this project, it creates a system to crawl the web and create ClueWeb09 dataset[16] in 2009. In ClueWeb09, there are total 1,040,809,705 web pages in 10 languages, 503,903,810 pages in English[17], and 177,489,357 pages in Chinese. Those pages are stored in gzipped WARC format and are easy to read by using programming languages such as Java.

This dataset has 25 TB (uncompressed) and 5 TB (compressed), which is huge for researchers to handle it. The large ClueWeb09 is a good dataset for us to use it in extracting knowledge. On the other hand, processing the dataset may beyond the reach of many academic laboratories in this scale. For example, if it takes 1 second to segment and tag POS for a Chinese web page, it will take 5.6 years by a computer with a single node.

In processing ClueWeb09, parsing web pages, POS tagging, and phrase chunking are all time consuming and need a lot of computational resources to accomplish these tasks. In this study, we handle these issues and produce many datasets for researchers. We will explain these datasets in the following sections.

6.2 Commonsense Knowledge in the Web

We first examine what kinds of knowledge can be extracted from the web pages. Researchers already adopt many approaches to extract knowledge from the texts (Cankaya & Moldovan, 2009; Chklovski, 2003; Clark & Harrison, 2009; Etzioni et al., 2008; Girju et al., 2006; Schubert & Tong, 2003; Schubert, 2009; Schwartz & Gomez, 2009). Some approaches claims

[16] http://lemurproject.org/clueweb09/ (Last access: 2013/01/20)

[17] There is a larger dataset ClueWeb12, which contains 1 billion English pages. This dataset is released in 2012.

their system can even extract commonsense knowledge from texts. We want to assert this point before we design knowledge extraction system.

Our idea is simple. If commonsense knowledge (CSK) exists in texts and we can extract this kind of knowledge from texts, using knowledge extracted from texts is a very good proof to test language understanding level of a system. Therefore, the first step is to verify does commonsense knowledge exist in texts. In our study (Yu & Chen, 2010), we show that commonsense knowledge is actually explicit stated and exists in texts.

The verification step is simple. We adopt commonsense knowledge in OMCS dataset, which is a well-known public contributed commonsense knowledge dataset. We search these CSK in the web and find that a lot of CSK can be found in the web. The more reliable that human consider a sentence to be a commonsense knowledge, the more probable that we can found this knowledge in the web. We show the results in Figure 7[18].

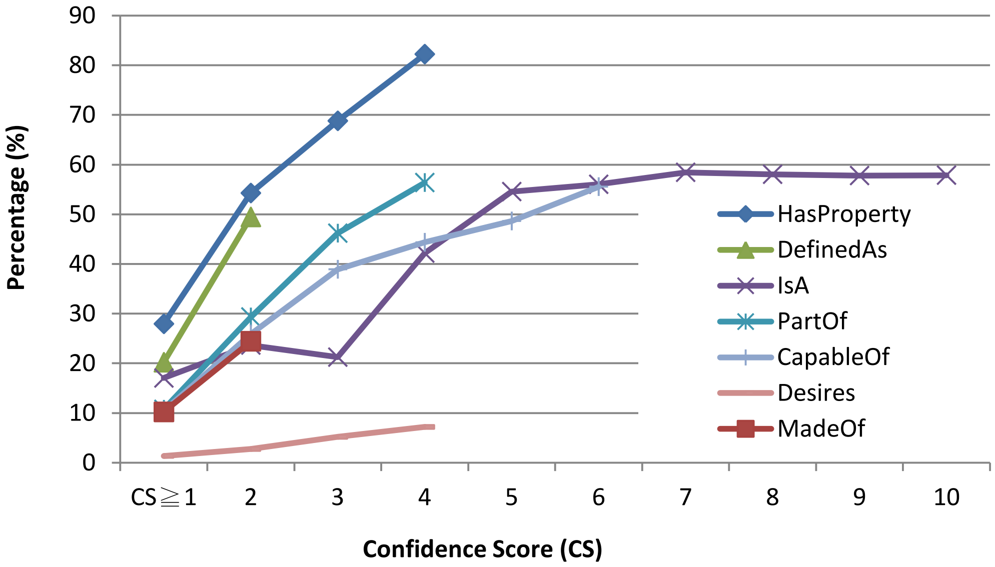

Figure 7. Relationship between predicate types and explicitly stated CSK

The CS in Figure 7 is confidence score, which is determined by users collaboratively.

[18] This figure also appears in our paper (Yu & Chen, 2010).

When a user asserts a sentence as a valid CSK, the corresponding score is increased by one, and vice versa. The confidence score is the quality of a CSK sentence. The relations such as *IsA* and *PartOf* are defined in OMCS dataset. We can see that a lot of CSK relations can be found in the web. Therefore, using ClueWeb09, which is part of the web, is a good knowledge source for our purpose. In section 6.5 we will examine the issue that if the size of ClueWeb09 is large enough for many general NLP tasks.

6.3 Preprocessing of English Web Pages

We preprocess English web pages in the following steps:

(1) We translate web pages to pure text in RFC3676 format.

(2) We filter noise texts.

(3) We tag part-of-speech for each sentence.

(4) We chunk phrases.

We describe the first two steps in APPENDIX II. For POS tagging, we adopt Stanford POS tagger (Toutanova et al., 2003) to tag sentences and use trained tagging model wsj-0-18-left3words-distsim.tagger, which archives 97.01% correct on WSJ 19-21 and 89.81% correct on unknown words. The Stanford tagger uses Penn Treebank POS tagset for tagging. We show the statistics of resulting dataset in Table 7.

Number of documents (pages)	397,947,894
Number of paragraphs	4,328,910,952
Number of sentences	10,338,365,571
Number of words	183,978,577,151

Table 7. Statistics of English POS-tagging dataset.

The paragraph in Table 7 is different from paragraph in texts. If we encounter some HTML tag such as <p>, <br>, and table related tags, we break string to different paragraphs. We also show tag distribution in APPENDIX III.

We use chunker from Apache OpenNLP 1.5.2. This chunker uses CoNLL 2000 shared task for its training set[19]. We show one example in Table 8. In the table, the chunked adjective phrase is <*very conservative and tasteful*>. Other two phrases are also chunked but its length is only one word. The chunker uses BIO scheme for its tags.

English Word	**POS Tag**	**Chunker's Tag**
You	PRP	B-NP
are	VBP	B-VP
very	RB	B-ADJP
conservative	JJ	I-ADJP
and	CC	I-ADJP
tasteful	JJ	I-ADJP
.	.	O

Table 8. A phrase-chunking example.

The resulting dataset is compressed and be saved in Java serialization format. We have Java APIs to access the preprocessed English dataset. It is easy to access documents, paragraphs, sentences, chunked phrases, POS tag, and word. This dataset will be used in our knowledge extraction task, and we hope this dataset will be useful for researchers.

6.4 Preprocessing of Chinese Web Pages

Chinese web pages are more difficult to be handled. The first obstacle is encoding of web pages. In Chinese, there are many encodings for Chinese characters. In addition, there are different character sets that contain different Chinese characters. For example, Unicode is a

[19] http://www.clips.ua.ac.be/conll2000/chunking/ (Last access: 2013/01/21)

character set, and its newest version 6.2 contains more than more than 70,000 Han characters. Those Han characters are widely used in China, Japan, Korea, Taiwan, and Vietnam. Unicode has many encoding standards to store characters in different digit formats for different usages such as UTF-8, UTF-16, and UCS-4 (UTF-32). In addition to Unicode, the most popular character set in traditional Chinese may be the Big5 character set, which also has many versions and contains different Chinese characters. In simplified Chinese, there are many encoding standards such as GBK and GB2312. In order to deal with those variations, we convert all web pages to Unicode to simplify the question.

Therefore, the first step is to detect encoding of a web page. In our paper (Yu, Tang, & Chen, 2012), we propose an algorithm to detect page's encoding. Also, the proposed algorithm deals with mixed languages problem, in which there are more than two languages in single web page. For example, Chinese and Japanese language may be in a page from game forum.

After converting all pages to correct encodings, we extract web pages to pure text in RFC3676 format. Because most of Chinese pages in ClueWeb09 are in simplified Chinese, we translate all web pages to simplified Chinese. We then adopt the Stanford Chinese Word Segmenter (Tseng, Chang, Andrew, Jurafsky, & Manning, 2005) and the Stanford Log-linear Part-Of-Speech Tagger (Toutanova et al., 2003) to process the texts. The details are in our paper (Yu et al., 2012), and we show the statistics of resulting dataset in Table 9 (also appears in our paper).

Number of web pages	173,741,587
Number of sentences	9,598,430,559
Number of tokens (terms)	141,179,769,123
Number of digit terms	4,308,254,253
Number of foreign words	4,095,774,930
Number of character sequences	29,078,949,574
Average sentences per page	55.2
Average tokens per sentence	14.7

Table 9. The statistics of the resulting Chinese POS-tagged dataset.

Based on this Chinese POS-tagged dataset, we extract POS information for Chinese words and produce *NTU PN-Gram corpus* (Yu et al., 2012), which contains POS information for each *n*-gram (*n* in [1,5]). We also implement a web system for users to access the dataset easily (Yu & Chen, 2012a). We show statistics about *NTU PN-Gram corpus* in Table 10 (also appears in our paper).

N	# NTU PN-Grams	# Google Chinese N-Grams	Ratio
1	2,219,170	876,004[20]	2.5 : 1
2	62,728,971	281,107,315	1 : 4.5
3	200,066,527	1,024,642,142	1 : 5.1
4	294,016,661	1,348,990,533	1 : 4.6
5	274,863,248	1,256,043,325	1 : 4.6

Table 10. Comparison of *NTU PN-Gram corpus* and Google Chinese Web N-gram corpus

In Table 10, Google Chinese Web N-gram corpus (F. Liu, Yang, & Lin, 2010) uses 882,996,532,572 tokens to derive n-gram information while our *NTU PN-Gram corpus* uses 141,179,769,123 tokens (ratio 6.3 : 1). In next section, we want to know is this size enough for general NLP tasks.

In addition to POS information, we also identify discourse markers in Chinese texts. We

[20] We exclude 740,146 non-Chinese tokens in Google unigrams.

identify 319 single word markers (such as *另外/in addition to*), 85 inter-sentential connectives (such as *首先/the first …其次/the second*), and 404 intra-sentential connectives (such as *除非/unless …否则/otherwise*). When we identify the markers, inter-sentential connectives and intra-sentential connectives have higher priority than that of single markers. This means that if a word is used in inter-sentential connective or intra-sentential connective, it cannot be we a single marker. For example, *反而/instead* is a single marker, but it also appears in intra-sentential marker *不但/not only …反而/instead*. If we identify *反而/instead* to be a part in *不但/not only …反而/instead*, it will not be a single word marker. Sometimes, a intra-sentential marker can also be a inter-sentential marker. For example, *一方面/on the one hand …另一方面/on the other hand* can be both intra- and inter-sentential connectives. There are 50 unique connectives of this type. We show them in Table 11.

一方面,另一方面	因为,而	或者,或者	由于,所以	还是,还
不仅,同时	固然,但是	接着,最后	由于,而	还是,还是
不仅,而	固然,然而	接着,然后	目前,未来	随后,最后
不仅,而且	尽管,可是	既,更	虽然,不过	首先,之后
不但,更	尽管,然而	既然,所以	虽然,但	首先,其次
不但,还	尽管,而	既然,而	虽然,但是	首先,接着
因为,因此	当初,后来	早期,然后	虽然,可是	首先,最后
因为,就	当初,如今	最初,后来	虽然,然而	首先,然后
因为,所以	当初,目前	最早,后来	虽然,而	首先,目前
因为,於是	或,或	由于,因此	虽说,但	首先,而且

Table 11. 50 unique connectives that can be both intra- and inter-sentential connectives

The identified markers are stored in a predefined format, and Java APIs are provided to access those information. We hope that the resulting datasets can boost NLP researches.

6.5 A Verification of ClueWeb09 Dataset

Because ClueWeb09 is a small part of the whole web, we want to know is this size enough for our purpose. We want to produce useful datasets, and we want to assure that the datasets can derive reliable probability for NLP tasks. We use an NLP tasks and compare the use of two different resources (our dataset and Google Web N-gram corpus) to see if the results are the same. In addition to verifying the usefulness of the datasets, we also introduce the use of concept representation. We use the proposed representation to represent sentences. We adopt the detections of word ordering error in Chinese sentences for this purpose (Yu & Chen, 2012b).

The word ordering error in Chinese is the cases where words are placed in the wrong places in Chinese sentences. This may results in wrong words and grammatical errors. For example, *<我/I (am) 很/very 有兴趣/interest in 对/for 关于/related 服装的工作/Work on clothing>* is an error sentence, and the correct sentence should be *<对/for 关于/related 服装的工作/Work on clothing 我/I (am) 很有/very 兴趣/interesting>*.

The adopted concept representation uses syntactical knowledge and knowledge from two datasets[21]: *NTU PN-Gram corpus* and Google Web N-gram corpus. We show the process in Figure 8.

[21] We also use another features, but we did not show it here.

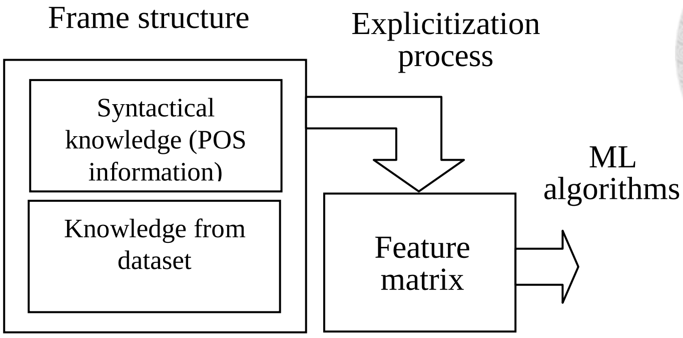

Figure 8. Illustration of concept representation in WOE detections.

The knowledge we used in WOEs detection is simple, and the static part of the knowledge is POS tagging information of a sentence and probability derived from *n*-gram datasets. It is clear that this representation is easy to understand and easy to extent.

If we replace the knowledge source from the larger Google Web N-gram corpus to the smaller *NTU PN-Gram corpus* and then compare system's results, we can answer the question that is the size of ClueWeb09 enough for deriving reliable probability. The answer is yes, and we describe experiments below.

We conduct experiments on two kinds of datasets (HSK-HSK and NAT-HSK) from HSK corpus. The results is shown in Table 12.

Features	HSK-HSK		NAT-HSK	
	accuracy (%)	stdev.	accuracy (%)	stdev.
C_{gW}	50.84	2.26	63.17	1.54
C_{gS}	52.64	2.01	65.01	2.23
$(C_{gW,} C_{gS})$	52.59	2.21	64.77	2.29
C_{bW}	50.90	2.94	63.90	1.81
C_{bS}	51.95	2.67	65.09	1.69
$(C_{bW,} C_{bS})$	56.99	1.98	65.93	2.01

Table 12. Compare results using *NTU PN-Gram corpus* and Google Web N-gram corpus.

In Table 12, feature C_{gW} denotes the system adopts Google Web N-gram corpus and does not use segmentation system. Feature C_{gS} denotes the system adopts Google Web N-gram corpus and does use segmentation system. Feature ($C_{gW,}$ C_{gS})[22] denotes we combine two feature vectors to produce a new vector. The subscript character b in C_{bW} denotes the same setting except we adopt *NTU PN-Gram corpus* as reference corpus. We can find that although the size of *PN-Gram corpus* are much smaller than that of Google Web N-gram corpus, this do not make difference (see C_{gW}. and C_{bW}). This conclusion still holds when we use segmentation system (see C_{gS}. and C_{bS}). This give us confidence to use ClueWeb09 and the resulting datasets.

[22] For detail setting, please refer to our paper (Yu & Chen, 2012b).

Chapter 7. Conclusion and Future Work

In this dissertation, we focus on concept representation in building intelligent systems. For this purpose, we define concept a continuation, which is a kind of temporary state in concept computation process of human. We put continuation in the context of the evolutionary language game. Based on this setting, we discuss some theoretical aspects of the definition. We also consider the concept theories, which relate to the world behind mathematical models. We derive some conclusions on three kinds of stability: input stability, test stability, and dogma stability.

We propose a concept representation scheme, which contains static frame structure and dynamic explicitization process. This concept representation scheme features transparency and flexibility as its core advantages. We want the concept representation can be adopted in many different tasks.

To demonstrate the application of our concept definition, we apply our concept definition in two problems: commonsense knowledge classification and word sense disambiguation. We use commonsense knowledge classification to demonstrate how to use our definition in traditional machine learning process. We further use our definition to derive new concepts to handle WSD problem. We investigate concept appropriateness and concept fitness in the relation between concept and its context, which is similar to continuation and its context. We use these two concepts to formulate new algorithms to learn models for WSD.

In addition to the concept definition, we conduct experiments and assert that the texts contain commonsense knowledge, therefore, texts is a good source for mining proofs to test system's understanding level. We also conduct experiments and assert that ClueWeb09 is a good knowledge source although it contains small part of the whole web.

We preprocess ClueWeb09 and produce three datasets for researchers. The first dataset is

POS-tagged and phrase-chunked English datasets. The second dataset is segmented, POS-tagged, and discourse marker identified Chinese dataset. The third dataset is *NTU PN-Gram corpus*, a Chinese *n*-gram dataset with POS information (*n* in [1, 5]). We have design a web system for general users to access this *NTU PN-Gram corpus*. The system is designed to boost the speed of query this big *n*-gram dataset.

In the future, we want to deeply investigate our concept definition in many aspects such as developing internal architecture of a continuation and grounding these structures in philosophical viewpoints. We want to use our concept representation scheme in more problems, and finally, we hope this concept definition can boost researchers to build a real intelligent system in the future.

REFERENCE

Agarwal, S., & Niyogi, P. (2005). Stability and generalization of bipartite ranking algorithms. In *Proceedings of the 18th Annual Conference on Learning Theory* (pp. 32–47). Berlin, Heidelberg: Springer-Verlag.

Agirre, E., & Edmonds, P. (Eds.). (2006). *Word sense disambiguation: Algorithms and Applications*. Springer.

Agirre, E., & Martinez, D. (2004). Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 25–32.

Agirre, E., & Stevenson, M. (2006). Knowledge sources for WSD. In *Word sense disambiguation: Algorithms and Applications. Agirre, Eneko and Edmonds, Philip (Eds.)*. Springer.

Ando, R. K. (2006). Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 77–84. Association for Computational Linguistics.

Barabanov, N. E., & Prokhorov, D. V. (2002). Stability analysis of discrete-time recurrent neural networks. *IEEE Transactions on Neural Networks*, *13*(2), 292–303.

Barker, C. (2004). Continuations in natural language. In *Proceedings of the Fourth ACM SIGPLAN Continuations Workshop*.

Bengio, Y. (2008). Neural net language models. In *Scholarpedia, 3(1):3881*.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research, 2,* 499–526.

Brants, T., & Franz, A. (2006). Web 1T 5-gram Version 1. Linguistic Data Consortium, Philadelphia.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–479.

Cankaya, H. C., & Moldovan, D. (2009). Method for extracting commonsense knowledge. In *Proceedings of the Fifth International Conference on Knowledge Capture* (pp. 57–64). ACM.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010).*

Carpuat, M., & Wu, D. (2005). Word sense disambiguation vs. statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 387–394. Stroudsburg, PA, USA: Association for Computational Linguistics.

Carpuat, M., & Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* , 61–72.

Chan, Y. S., & Ng, H. T. (2007). Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07),* 33–40.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*(3), 27:1–27:27.

Chklovski, T. (2003). Learner: A system for acquiring commonsense knowledge by analogy. In *Proceedings of the 2nd International Conference on Knowledge Capture* (pp. 4–12). ACM.

Chklovski, T., & Gil, Y. (2005). An analysis of knowledge collected from volunteer contributors. In *Proceedings of the 20th International Conference on Artificial Intelligence*, 564–570. AAAI Press.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Clark, P., & Harrison, P. (2009). Large-scale extraction and use of knowledge from text. In *Proceedings of the Fifth International Conference on Knowledge Capture* (pp. 153–160). ACM.

Cohen, W. W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *Journal of Artificial Intelligence Research, 10*(1), 243–270.

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the IEEE/ACL 2006 Workshop on Spoken Language Technology*. Stanford University.

Dhillon, P. S., Foster, D., & Ungar, L. H. (2011). Minimum description length penalization for group and multi-task sparse learning. *Journal of Machine Learning Research*, *12,* 525–564.

Dhillon, P. S., & Ungar, L. H. (2009). Transfer learning, feature selection and word sense disambiguation. In *Proceedings of the ACL-IJCNLP 2009 Conference*, 257–260. Stroudsburg, PA, USA: Association for Computational Linguistics.

Edmonds, P., & Cotton, S. (2001). Senseval-2: Overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems* (pp. 1–5). Association for Computational Linguistics.

Erk, K., & McCarthy, D. (2009). Graded word sense assignment. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 440–449. Association for Computational Linguistics.

Erk, K., McCarthy, D., & Gaylord, N. (2009). Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 10–18. Association for Computational Linguistics.

Erk, K., & Padó, S. (2008). A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 897–906. Association for Computational Linguistics.

Escudero, G., Màrquez, L., & Rigau, G. (2000). Naive Bayes and exemplar-based approaches to word sense disambiguation. *Proceedings of the 14th European Conference on Artificial Intelligence*, 421–425.

Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, *51*(12), 68–74.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Felleisen, M. (1988). The theory and practice of first-class prompts. In *Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (pp. 180–190).

Florian, R., & Yarowsky, D. (2002). Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 25–32.

Friedman, M. (1974). Explanation and scientific understanding. *Journal of Philosophy*, *71*(1), 5–19.

Geng, X., Liu, T.-Y., Qin, T., Arnold, A., Li, H., & Shum, H.-Y. (2008). Query dependent ranking using K-nearest neighbor. In *Proceedings of the 31st Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, 115–122.

Girju, R., Badulescu, A., & Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics, 32*(1), 83–135.

Gold, E. M. (1967). Language identification in the limit. *Information and Control, 10*(5), 447–474.

Gonzalo, J., & Verdejo, F. (2006). Automatic acquisition of lexical information and examples. In *Word sense disambiguation: Algorithms and Applications. Agirre, Eneko and Edmonds, Philip (Eds.)*. Springer.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

Hjørland, B. (2009). Concept theory. *Journal of the American Society for Information Science and Technology, 60*(8), 1519–1536.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, 133–142.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*, 1–37.

Jurafsky, D., & Martin, J. H. (2009a). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall.

Jurafsky, D., & Martin, J. H. (2009b). The representation of meaning. In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

Kilgarriff, A. (2006). Word senses. In *Word sense disambiguation: Algorithms and Applications. Agirre, Eneko and Edmonds, Philip (Eds.)*. Springer.

Kintsch, W. (2001). Predication. *Cognitive Science*, *25*(2), 173–202.

Komarova, N. L., Niyogi, P., & Nowak, M. A. (2002). Computational and evolutionary aspects of language. *Nature, 417*(6889), 611–617.

Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, 41–48. Association for Computational Linguistics.

Lee, Y. K., Ng, H. T., & Chia, T. K. (2004). Supervised word sense disambiguation with Support Vector Machines and multiple knowledge sources. In R. Mihalcea & P. Edmonds (Eds.), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text* (pp. 137–140). Association for Computational Linguistics.

Lenat, D. B., & Guha, R. V. (1989). *Building Large Know-ledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc.

Liu, F., Yang, M., & Lin, D. (2010). Chinese Web 5-gram version 1. Linguistic Data Consortium.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–331.

Margolis, E., & Laurence, S. (2011). Concepts. In *The Stanford Encyclopedia of Philosophy*.

Markert, K., & Nissim, M. (2007). SemEval-2007 Task 08: Metonymy resolution at SemEval-2007. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (Vol. 36–41).

Martinez, D., de Lacalle, O. L., & Agirre, E. (2008). On the use of automatically acquired examples for all-nouns word sense disambiguation. *Journal of Artificial Intelligence Research*, *33*(1), 79–107.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Mihalcea, R., Chklovski, T., & Kilgarriff, A. (2004). The Senseval-3 English lexical sample task. In R. Mihalcea & P. Edmonds (Eds.), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text* (pp. 25–28).

Mihalcea, R. F. (2002). Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC), Las Palmas*.

Mihalcea, R., & Moldovan, D. I. (1999). An automatic method for generating sense tagged corpora. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative*

Applications of Artificial Intelligence, 461–466.

Minsky, M. (1986). *The Society of Mind*. New York: USA: Simon & Schuster, Inc.

Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 236–244.

Mueller, E. T. (2010). *Commonsense Reasoning*. Elsevier Science.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR), 41*(2), 10:1–10:69.

Navigli, R., & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(4), 678–692.

Navigli, R., Litkowski, K. C., & Hargraves, O. (2007). SemEval-2007 Task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 30–35).

Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 40–47.

Nowak, M. A., Plotkin, J. B., & Krakauer, D. C. (1999). The evolutionary language game. *Journal of Theoretical Biology, 200*(2), 147 – 162.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., & Dang, H. T. (2001). English tasks: all-words and verb lexical sample. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems* (pp. 21–24).

Pantel, P., & Lin, D. (2002). Discovering word senses from text. In *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, 613–619. ACM.

Plate, T. A. (1995). Holographic Reduced Representations. *IEEE Transactions on Neural Networks, 6*(3), 623–641.

Plate, T. A. (2003). *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. Stanford, CSLI Publications.

Plotkin, J. B., & Nowak, M. A. (2000). Language evolution and information theory. *Journal of Theoretical Biology, 205*, 147–159.

Priss, U. (2004). Linguistic applications of Formal Concept Analysis. In *Proceedings of the*

First International Conference on Formal Concept Analysis. Springer.

Priss, U. (2006). Formal Concept Analysis in information science. *Annual Review of Information Science and Technology*, *40*(1), 521–543.

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach, 2/E*. Pearson Education.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, 142–151.

Schubert, L. (2009). From generic sentences to scripts. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, Workshop: Logic and the Simulation of Interaction and Reasoning (LSIR).*

Schubert, L., & Tong, M. (2003). Extracting and evaluating general world knowledge from the Brown corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9* (pp. 7–13).

Schuemie, M. J., Kors, J. A., & Mons, B. (2005). Word sense disambiguation in the biomedical domain: An overview. *Journal of Computational Biology, 12*(5), 554–565.

Schwartz, H. A., & Gomez, F. (2009). Acquiring applicable common sense knowledge from the Web. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics* (pp. 1–9).

Searle, J. (1980). Minds, brains and programs. *Brains and Programs. Behavioral and Brain Sciences, 3*(3), 417–457.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27,* 379–423.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., & Zhu, W. L. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems* (pp. 1223–1237). Springer-Verlag.

Snyder, B., & Palmer, M. (2004). The English all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text* (pp. 41–43).

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Boston: Addison-Wesley Longman Publishing Co., Inc.

Stevenson, M., & Guo, Y. (2010). Disambiguation in the biomedical domain: The role of ambiguity type. *Journal of Biomedical Informatics*, *43*(6), 972–981.

Stevenson, M., Guo, Y., & Gaizauskas, R. (2008). Acquiring sense tagged examples using relevance feedback. In *Proceedings of the 22nd International Conference on Computational Linguistics*, 809–816.

Stokoe, C., Oakes, M. P., & Tait, J. (2003). Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th Annual International Conference on Research and Development in Informaion Retrieval*, 159–166.

Thater, S., Fürstenau, H., & Pinkal, M. (2010). Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 948–957.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 173–180.

Towell, G., & Voorhees, E. M. (1998). Disambiguating highly ambiguous words. *Computational Linguistics*, *24*(1), 125–145.

Trapa, P. E., & Nowak, M. A. (2000). Nash equilibria for an Evolutionary Language Game. *Journal of Mathematical Biology*, *41*(2), 172–188.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., & Manning, C. (2005). A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*(1), 141–188.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Yu, C.-H., & Chen, H.-H. (2010). Commonsense knowledge mining from the Web. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1480–1485.

Yu, C.-H., & Chen, H.-H. (2012a). Chinese web scale linguistic datasets and toolkit. In

Proceedings of the 24th International Conference on Computational Linguistics, 501–508.

Yu, C.-H., & Chen, H.-H. (2012b). Detecting word ordering errors in Chinese sentences for learning Chinese as a foreign language. In *Proceedings of the 24th International Conference on Computational Linguistics*, 3003–3018.

Yu, C.-H., Tang, Y., & Chen, H.-H. (2012). Development of a web-scale Chinese word n-gram corpus with parts of speech information. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12),* 320–324.

Zhong, Z., & Ng, H. T. (2012). Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* 273–282.

APPENDICES

APPENDIX I. The Definition of *definition*

We propose that “a *definition* connects object to other objects and use criteria to rate the goodness of this connection” in text. We denote this viewpoint as **Connection Rating**. Here, we argue that this explanation of term *definition* can subsume three viewpoints in Friedman's (1974) article.

Friedman's article gives three viewpoints of scientific explanations: (quotes below are from Friedman's article)

1. **D-N model**: “According to the D-N model, a description of one phenomenon can explain a description of a second phenomenon only if the first description entails the second.”
2. **Familiarity**: “scientific explanations give us understanding of the world by relating (or reducing) unfamiliar phenomena to familiar ones.”
3. **Intellectual Fashion**: “the phenomenon doing the explaining must have a special epistemological status ... this status varies from scientist to scientist and from historical period to historical period. At any given time certain phenomena are regarded as somehow self-explanatory or natural.”

Because “explanation” is also a concept, we can interpret these three views by using four families of concept theories. For example, Intellectual Fashion obviously uses concept theories from rationalism and historicism.

We can see that if the connection is restricted to having entailment property, Connection Rating subsumes D-N model viewpoint. If the connection is restricted to having reduction property and the explanatory objects must be familiar ones, Connection Rating subsumes

Familiarity viewpoint. If the explanatory objects have special epistemological status, and the explanatory objects and rating function are time-variant and scientist-depend, Connection Rating subsumes Intellectual Fashion viewpoint.

Actually, philosophers have proposed many criteria for the rating function. For Karl Popper (1902 – 1994), the criterion is falsifiability when we judge the goodness of a science theory. In logical positivism, the criterion is verifiability of explanatory objects. For Thomas Kuhn (1922 – 1996), the author of *The Structure of Scientific Revolutions*, the criteria are changing for different scientist societies and different historical periods. For scientists whose believe Galilean style, the criterion gives mathematical models higher priority to the reality. These criteria can be subsumed by the Connection Rating viewpoint, which gives a unified viewpoint of scientific explanation.

APPENDIX II. The Filtering of Noise Texts

To have a cleaner dataset for knowledge extraction, it is necessary to filter out noises as much as we can, especially when the ClueWeb09 dataset are composed by web pages. The filtering procedures are as follows.

1. Convert a HTML page to a sequence of Java String.
2. For each string, we filter out it if it did not contain knowledge we want.

We use Jericho HTML Parser[23] to convert HTML pages to pure text in RFC3676 format. In the converting procedure, we write the converted strings in Java's serialization format. The serialization format can preserve the order of texts as they appear in the web pages. All extracted elements in HTML page are converted to a sequence of Java strings. For example, the content in <P> tag will be transform to single string, as well as the content of table's cell (<TD> tag) will be in a single string. HTML tags, script codes, and other HTML elements which are for formatting purpose are removed, and the result of transformed HTML page is a sequence of text string. A single string may be a word, a phrase, a sentence, or a paragraph which depends on the author of a web page. Not all strings are helpful for knowledge extraction, so we design a simple and fast approach to filter out useless strings.

We filter out many types of strings that obviously did not help for knowledge extraction. These filtered strings included script codes (due to the mal-formed HTML page), words for site's functions link, number, and named entity such as proper name, date, and time. Some of filtered strings are shown as follows.

[23] http://jericho.htmlparser.net/docs/index.html

* Demographics (64)	(list item)
* Economics (64)	(list item)
Henry L. Williams	(name)
Hire Me to Write!	(slogan)
as of 2005.	(word + date)
^ Page 238.	(word + number)
30 Jan 2006.	(date)
About OLX | Terms Of Use | Contact Us	(site's function link)
Login | Sign Up!	(site's function link)
//<![CDATA[Sys.Application.add_init(function()…	(script codes)
Tags: al green, alex gopher, annie, black keys, …	(tags)

When processing the huge ClueWeb09 dataset, the speed is the most important consideration. We investigate two possible approaches, rule-based and linear SVM classifier (Fan, Chang, Hsieh, Wang, & Lin, 2008), for deciding a string of being selected or not. Rule-based approach uses a list of if-else decisions which is simple and fast, but is hard to find out the best filtering rules. Linear SVM classifier, on the other hand, is more theoretical valid, and it is fast because it uses linear decision function $w^t x$, where *w* is the learned weights and *x* is feature vector. The hard problem here is to construct a test set for evaluating both approaches. We adopt a blended approach.

We first use a simple rule-based approach as bootstrap step to construct training set. We process 4 ClueWeb09 data files first, and use heuristic rules to decide invalid strings, which are the strings we want to filter. Each data file contains about 33000 HTML pages, and resulting 1893,512 valid strings and 10,860,623 invalid strings. We then adopt Liblinear[24] to train a classifier with L2-regularized L2-loss support vector classification.

The features we used include number of sentences, number of period, string length, number of sentence markers (.!?), the ratio of maximum word length to string length, and the

[24] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

ratios of different character categories, such as digit, space, A-to-Z characters, and special chars used in scripts (@+-&%*/~#;,.!?|$^\"\:=`). For some scalar features in range [0,1], we add four flags to indicate the scalar quantization result. The number of total features is 92, and all features are normalized to [-1, 1]. These features can be extracted in linear time O(L) where L is the string length. The best inside test performance of linear SVM is about 97.50%.

The third step is applying the learned model to training set, and we manually induce simple rules from these errors. The final rules we used for filtering are as follows.

1. The minimum string length must larger than 30 characters. This will filter out most list item, named entities and table cells.
2. The number of space must larger than 1.
3. If $\frac{\text{the number of vertical bar characters } |}{\text{number of word} + 1} > 0.1$, we skip the string.
4. If $r = \frac{\text{the number of characters in set}\, \{ <>\{\}[]() \} \text{ and special chars used in scripts}}{\text{string length}} > 0.1$, we skip the string.
5. If $r > 0.05$ and $\frac{\text{the number of characters in set}\, \{ <>\{\}[]()\ @+-\&\%*/\sim\#;,|\$\hat{}\backslash"\backslash:=\grave{}\}}{\text{string length}}$ >0.05, we skip the string. This rule takes the sentence delimiters (.?!) into consideration.
6. Otherwise, this string is a valid string we want.

By using the induced rules in third step, we found that the result set is more feasible for knowledge extraction.

APPENDIX III. English POS Tag Distribution

We show the English POS tag distribution in the following table.

PSO Tag	Count	PSO Tag	Count
NN	28,620,799,002	-RRB-	937,675,464
IN	17,750,641,457	-LRB-	900,374,892
NNP	16,881,018,047	FW	860,773,391
DT	14,758,146,518	WDT	744,615,654
JJ	12,012,182,379	WRB	670,162,968
NNS	10,635,509,446	'	656,528,827
.	9,376,537,123	`	631,464,670
,	8,190,673,858	POS	599,654,658
VB	6,438,881,097	NNPS	544,771,661
RB	6,157,087,150	WP	544,603,373
CC	5,890,474,901	JJR	529,219,620
PRP	5,748,676,031	RP	468,632,400
VBZ	4,281,854,212	JJS	379,247,918
TO	4,028,295,683	EX	237,663,308
VBP	3,851,865,455	RBR	213,693,574
VBN	3,745,399,209	SYM	175,364,641
CD	3,617,712,870	RBS	106,185,138
VBG	3,022,735,634	$	98,689,184
VBD	2,998,857,011	PDT	92,562,716
PRP$	2,409,436,097	UH	77,926,846
MD	2,021,238,205	LS	49,512,193
:	1,981,648,298	#	24,026,175
~~	~~~~~~~~~~~~~~~~	WP$	15,558,197

Table 13. English POS tag distribution.