國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

多條件耦合之半監督式學習於中文知識擷取之研究

Coupled Semi-Supervised Learning for Chinese Knowledge Extraction

麻立恒

Leeheng Ma

指導教授：許永真博士

Advisor: Jane Yung-jen Hsu, Ph.D.

中華民國 一百零二年 柒月

July, 2013

國立臺灣大學碩士學位論文
口試委員會審定書

多條件耦合之半監督式學習於中文知識擷取之研究

Coupled Semi-Supervised Learning for Chinese Knowledge Extraction

本論文係麻立恒君（學號 R99944037）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國一百零二年七月卅一日承下列考試委員審查通過及口試及格，特此證明

口試委員：許永真（簽名）

（指導教授）

陳信希　張嘉惠

蔡宗翰　劉昭麟

所　長：洪一平

誌謝

在這篇論文的完成過程中，受到許多人的幫助。首先我要感謝廣達提供的雲端中心和雲端中心的助教黃子桓學長的幫忙，沒有這些資源我的實驗就不可能這麼快就完成。也要感謝國網中心的公共實驗叢集和管理者 Jazz Wang，還有吳冠龍學長，在我學習 Hadoop 的路上給我很多幫助。感謝 Hiroshi Nakamura 提供的 Radix Tree open source project，讓我不需要重複打造輪子。還有 Charles Oliver Nutter，他在 12 小時內解決了 JRuby 上面的 bug 讓我的程式順利的無痛轉移。感謝我前實驗室的同學琦霖、信榮、璟騰、皓瑜、柏翔、誠軒、士賢、國書和聖翰，在我碩士期間的陪伴和互相鼓勵。還有 iAgent 的學長姊在啟嘉、Janet、喬敬、宇傑研究方面給我很多的建議。感謝 Iris 學姊幫助我釐清很多我誤解的觀念，還有 common sense group 小組的成員恰恰、季恩、昱儒還有學弟自均在這段時間的互相討論和幫忙。還要感謝我們同一期的戰友們 George、懋懋和緯倫，在這最後半年多的時間互相加油打氣、幫忙挑錯、一起承受壓力，一起畢業。

怡亭學姊，在我剛進實驗室時，對人生地不熟的我給了很多幫助。在在最後兩個多月裡，不但教了我許多寫論文的技巧還在自己的忙的焦頭爛額的情況下幫我挑出我論文中詞不達意的地方，非常感謝他花在我們這些不成材的後進上的時間和心力。

蔡宗翰老師，感謝老師在研究上給了很多建議和方向，尤其是語言處理方面，老師的經驗讓我受益良多。

許永真老師，許老師不但在我研究所最低潮的時候給我一個機會讓我加入 iAgent Lab，並且在研究的過程中毫無保留的支持我的每一個嘗試。每當我卡在某個點或是研究陷入膠著時，老師的建議總是讓我看到我沒注意的到的地方。老師不但在研究上教導我，在人生的經驗和做人做事上也讓我獲益良多。

最後要感謝我的女朋友，俞均，在研究的過程中包容我的沒有放這麼多心力在她身

上，並且支持我度過碩士生活。

還有辛苦養育我的媽媽，忍受一個讀書讀很久的兒子，沒有她就沒有我。

摘要

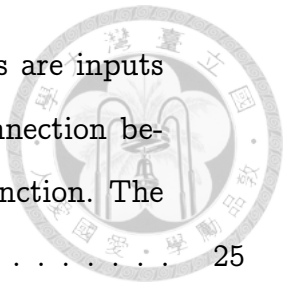
一個豐富的知識庫對於具有人工智慧的系統有很大的幫助，但是建立一個完整的知識庫卻需要花費無數的人力和時間。在「自動化的知識收集與萃取」這個領域，Never Ending Language Learning (NELL) 做了一個很好的示範，但是它在中文語言處理上的能力有限。本論文提出一個自動化中文知識萃取系統，我們發現在中文語句中，同一個類別的名詞常會和某些特定的動詞一起出現，我們利用這些動詞建立模版，來找到更多相同類別的名詞。我們結合 NELL 下的跨語言知識蒐集系統，以提高整體的正確率。最後，實驗證明我們的系統可以承載大規模的自動化中文知識蒐集。

Abstract

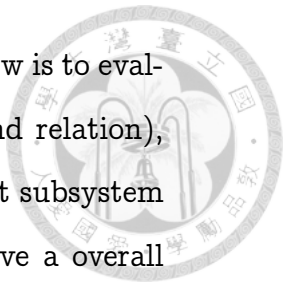
Robust intelligent applications benefit from rich knowledge bases. Building a rich and complete knowledge base is a time-comsuming and labor-intensive task. Never Ending Language Learning (NELL) is a great demonstration for large-scale automatic knowledge extraction, but unfortunately some components in NELL are not suitable to deal with Chinese. This thesis presents a Coupled Chinese Pattern Learner (CCPL), which extracts knowledge by textual patterns on relationships between nouns and verbs in Chinese sentences. We also implement Coupled Set Expander for Any Language (CSEAL) to collaborate with CCPL. The experiments show our system is capable of large-scale learning, and preserves high accuracy in automatic extraction for Chinese knowledge.

Contents

List of Figures



List of Tables

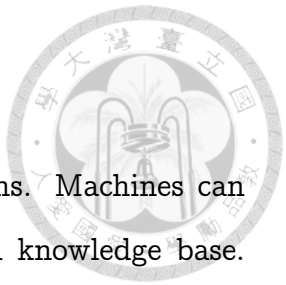


Chapter 1

Introduction

This chapter provides an overview of the thesis. At first, we explain why knowledge collection is important, and why it is a labor intensive work. Then we define knowledge collection problem and give our proposed solution to this problem.

1.1 Motivation



Knowledge base is essential for many intelligent applications. Machines can accomplish the human-intervention demanded jobs along with knowledge base. So knowledge collection is very valuable.

Traditional knowledge collection riles on expert systems, but expert systems usually collect only professional knowledge, such as medical, chemistry and physics domain-specific knowledge. The professional knowledge sometimes lack for providing common sense, and also limit the diversity of applications. People also tend to provide less information to systems and expect that responses from systems are more intelligent in all aspects, so comprehensive knowledge collection becomes more important.

One of practical problem is, knowledge collection are greatly constrained by frequency of human intervention. The frequency limits the scalability of collection. This problem motivates the development of automatic knowledge collecting methods by semi-supervised learning.

Semi-supervised learning exploits unlabeled data in stead of labeled data to continuously improve learned models. At beginning, semi-supervised learning only needs few labeled data, and improves learned model by unlabeled data to enable minimize human intervention.

The World Wide Web has been a fast growth natural language data source, even many researchers consider the web as a corpus for learning.

There were over 2.4 billion users of Internet in June 2012 [9]. When the Internet migrates to era of Web 2.0, user generated contents (UGC) becomes a large part of Internet. Chinese users increase rapidly among these users. In June 2012, Chinese users accounted for a quarter of total users. The population is still growing every day. Although contents of the Internet are rich, contents are not well-structured and contain many noises. Extracting valuable information from the Internet becomes a challenge.

1.2 Problem Definition

This section provides a formal definition for knowledge collecting problem and two basic assumptions for knowledge collection.

1.2.1 Assumption

Knowledge is defined by wiki as, "a familiarity with someone or something, which can include facts, information, descriptions, or skills acquired through experience or education." [7] In other words, knowledge is changeable based on the familiarities of different people. Although the same knowledge can have different explanations by different people, the agreements are still exist for most people.

The first assumption is that, knowledge is dynamic set. Every one have their own familiarity about knowledge. Furthermore, knowledge will grow and change over time. (e.g., Given a knowledge set K, K can be extended to a expended knowledge set K' such that $K \subseteq K'$).

The second assumption is that knowledge has an implicit order of truth confidence. A knowledge set contains many beliefs. We define $\succ$ for comparing truth confidence between two beliefs. $K \succ K'$ means K' is a knowledge set that has more people agree with compared with K. For most people K' is more credible.

According to the two assumptions, knowledge collection is extending knowledge set by beliefs that agreed by most people.

1.2.2 Knowledge Collection Problem

Knowledge collection is a process that is based on existing knowledge set to collect reliable beliefs, and keeps extending automatically. Formal definition is as following.

Given ontology O as initial knowledge base, which contains a set of predicates $O = \{P_1, P_2, ..., P_n\}$, and data source W. At the beginning, Ontology is extended from O to O' by merging beliefs which is related to predicates in O and implicit

in W. Repeat this process from O' and continuously expand knowledge base to form a never ending learning loop.

1.3 Proposed Solution

This work proposes an approach to automatically collect knowledge in Chinese. Our idea is base on some observations: First, knowledge or concept usually is noun phrase in sentences. Second, noun phrases which belong to the same type are interchangeable in sentences.

For example, a Chinese sentence "我今年暑假想要去美國。", the country name "美國" can be replaced by another country name and the sentence is still valid.

These observations lead us to a conjecture. There are some relationships exist between verb phrase and noun phrase in sentences within the same type of concept.

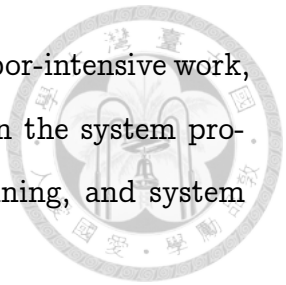
To verify our hypothesis, at first, we present a prototype based on above idea to automatically discover knowledge in Chinese. The prototype system named Coupled Chinese Pattern Learner, because it discovers knowledge by verb phrase patterns.

Second, we implement another wrapper-base knowledge extractor which is Set Expander for Any Language, which is originally developed by Wang and Cohen [23], and been applied coupling constraint by Carlson et al. [3].

Finally, we make two systems collaborate with each other together under architecture of Never-Ending Language Learner [2]. In future, it is easy to adding more Chinese knowledge extractor as subsystem component to enhance entire knowledge extraction process.

The proposed system will satisfy the following requirements:

- **Chinese knowledge extraction:** Our system is fully compatible with Chinese language. The system is based on Chinese language usage conventions. If input is a hybrid language sentence such as "我最喜歡的學校是 NTU。", then the system still has chance to extract non-Chinese knowledge.



- **Minor Human Intervening:** Knowledge extraction is labor-intensive work, because it need a large labeled data behind supported. In the system proposed, people only need to give initial ontology at beginning, and system discovers more knowledge by itself.

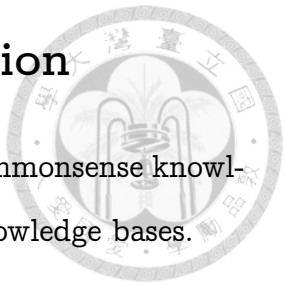
- **High Scalability:** The system employs coupling constraints proposed in NELL [3]. The better accuracy has been achieved when discovering more types of knowledge concurrently. We have proved this in Chapter 6.

Chapter 2

Related Work

This chapter presents introduction of well-known knowledge collections, followed by modern text mining techniques which extract knowledge from the web.

2.1 Commonsense Knowledge Collection



This section presents a brief introduction for well-known Commonsense knowledge collection. At first, we introduce two expert-developed knowledge bases.

WordNet

WordNet [13] was created at the Cognitive Science Laboratory of Princeton University since 1985.

WordNet is well-structured lexical database of English words, contents are carefully crafted by expert linguists. It groups words into sets of synonyms called synsets and connects synsets by relations. It has been used in varied domain applications successfully.

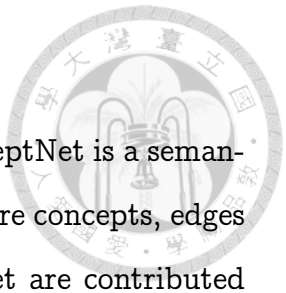
Cyc

Cyc [11] was started in 1984, the vision of Cyc is encoded million of commonsense knowledge into machine usable form. Cyc is also contributed by knowledge from experts, in order to represent these knowledge, they invented a new language "CycL" base on first order logic. CycL can support inference between concepts easily.

From experiences above, the scalability of knowledge base is limited when it is only contributed by few experts. When Internet migrates to era of Web 2.0, many researchers try to benefit from crowd-sourcing techniques.

Wikipedia

Wikipedia [7] is one of largest knowledge base of commonsense knowledge. It is easy for human searching, reading and modifying, a concept is represented by a document (web page), and wikipedia provides a taxonomy by its categories. Wikipedia sacrifices machine readability in exchange for human readability, it is purpose for projects like DBPedia [15].



ConceptNet

ConceptNet [12] was start in 1999, by MIT Media Lab, ConceptNet is a semantic network which is expressed as a directed graph whose nodes are concepts, edges are assertions between these concepts. Concepts in ConceptNet are contributed by many sources. In ConceptNet 5, sources includes DBPedia [15], ReVerb [17], English Wiktionary [8], WordNet [13] we mentioned before, and Game With A Propose (GWAP) project's word game Verbosity [22]. Even though knowledge collecting is more efficient, but the data quality is relatively low.

Freebase

Freebase [20] is a large collaborative knowledge base starting since 2007, the mainly different between Freebase and Wikipedia is way to store data, Freebase use meta data to describe concepts, try to represent concepts as topics, and use types and properties to provide more information about the concept, in the end, topics in Freebase form a graph. Because of structured data, machines can access Freebase easily. So far, most topics in Freebase are about movies, music, books and people, Freebase was acquired by Google in 2010.

2.2 Text Mining from the Web

Crowd-sourcing technique is not only solution for knowledge extraction. Researcher also tried to extract knowledge automatically. Naturally, the Internet is a plentiful corpus for researchers and it is still growing every day.

2.2.1 KnowItAll

KnowItAll [6], it is a system that extracts instances from the web. KnowItAll uses 8 generic patterns to extract instances from web using search queries, after extracting, instances which had been collected are ranked by mutual information and combined native bayesian classifier. A example for patterns in KnowItAll,

if it is known for “Taipei” is a instance of cities, then we can have pattern like “Taipei such as <city>” to extract more cities. Because of generic patterns are pre-defined, KnowItAll is language dependent.

2.2.2 Set Expander for any Language

Set Expander for any Language [23] is list extraction using wrapper induction. Modern web pages are usually generated by template, especially tables and lists in web pages. Wrapper induction attempts to capture contexts around known instances, then find more instances which around by the same contexts in web page. Because SEAL recognizes instances by contexts, contexts include sentences, punctuation marks, even markup languages, so it is for any language. The necessary condition is document has to be semi-structure. You can find more information about SEAL in Chapter 5.

2.2.3 Never Ending Language Learner

Never Ending Language Learner (NELL) [2] is a part of research project “Read The Web” under Carnegie Mellon University. It is large-scale knowledge extraction system which integrates four subsystem components to extract commonsense knowledge from the web every day. NELL accepts pre-defined ontology as input and initial knowledge base, and expand knowledge base by learned knowledge. The four subsystem components learn the knowledge by different aspect of the web, and knowledge integrator has to promote learned knowledge from four components to knowledge base. After knowledge base has been expanded, NELL starts to next learning process according to present knowledge base. We show NELL architecture in Figure 2.1.

Co-Training

The Co-Training algorithm [1] uses a bootstrap learning method to classify web pages, two classifiers are trained over different features from web pages, pre-

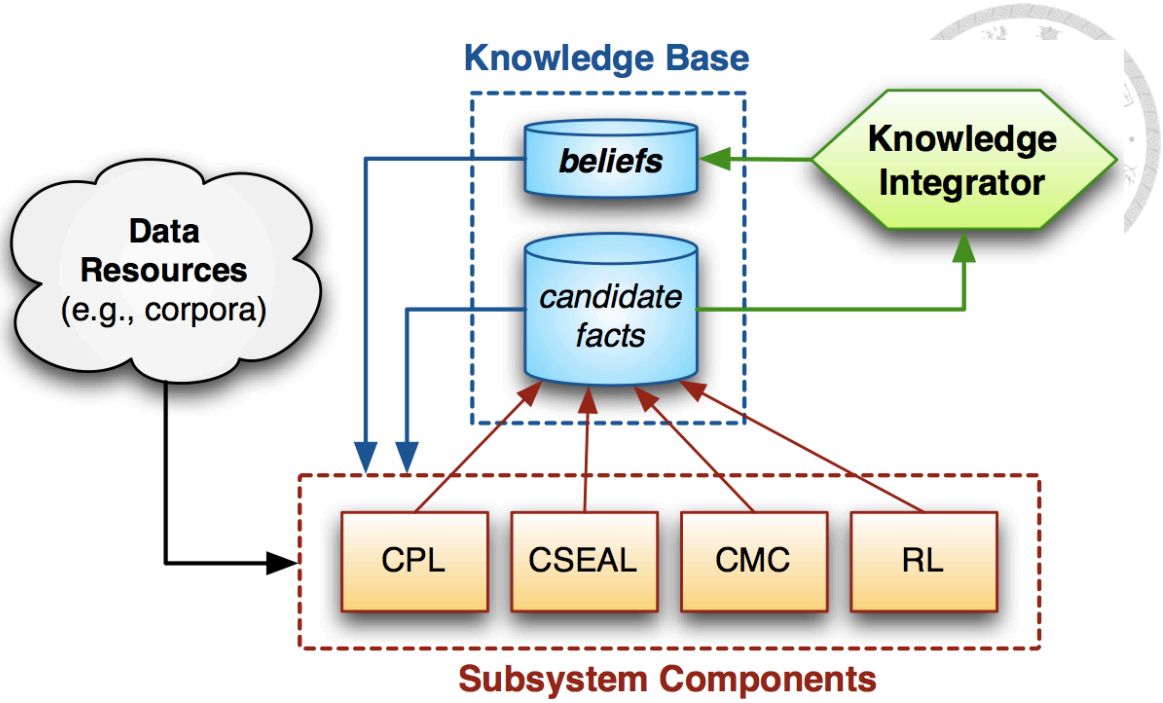


Figure 2.1: The Architecture of Never-Ending Language Learner. Subsystem components consisted of Coupled Pattern Learner(CPL), Coupled SEAL(CSEAL), Coupled Morphological Classifier(CMC) and Rule Learner(RL). Knowledge Integrator(KI) is responsible for choosing reliable candidates. The system has no end-point because it is a never-ending learning loop.

dictions which with most confident values from each classifier are used to label more documents for next iteration retraining. Co-Training can apply by any algorithms if algorithms work on different views of the same target. In NELL, web pages can be considered by document view, plain text view and morphological view. The document view includes HTML, CSS[1] and JavaScript[2] code inside the web page. The plain text view only considers free text in web page. The morphological view considers morphological features (root words, affixes, capitalization, parts of speech, intonation, stress). Coupling learning algorithms on these views can improve overall prediction.

[1]Cascading Style Sheets is a style sheet language which is used for describing the presentation semantics.

[2]Also called ECMAScript, the most widely used script language in browser Document Object Model (DOM) environment.

Never Ending

NELL use bootstrapping algorithm as core algorithm. The Internet is growing every day, large number of new concepts may arise on the web, bootstrapping learning algorithm can collect and update the knowledge base every iteration. Every day, NELL run each learning algorithms individually with exist knowledge base, knowledge integrator decides which concepts been updated to knowledge base finally, and continue to next iteration. Bootstrapping algorithms usually suffer "semantic drift"[3], NELL prove that coupling constraint is effective solution for restraining semantic drift.

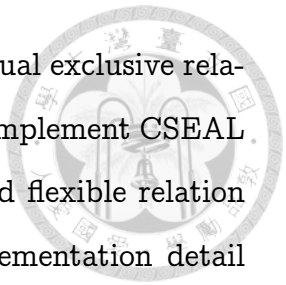
Subsystem Components

In this section, we brief introduce four subsystem components in NELL.

- **Coupled Pattern Learner:** CPL [3] is a free text extractor which uses contextual patterns to extract new instances of categories and relations. contextual patterns are "countries, such as X" and "X is teammates of Y". CPL uses co-occurrence between potential instances and contextual patterns to ensure confidence of potential instances and then find more contextual patterns by high confidence instances. Mutual exclusive relationship between predicates are used to filter out too common and general candidates, and relation instance has to pass the type-checking constraint on its domain and range. We base on idea of CPL to develop the Coupled Chinese Pattern Learner (CCPL), and adding more constraints and more language dependent handling for Chinese. We discuss more detail about CCPL in Chapter 4.

- **Coupled SEAL:** Coupled SEAL is a semi-structure extractor which query search engine for web page every iteration, and extract list items or table items by wrapper induction. CSEAL calls SEAL as sub-routine to extract

[3] About semantic drift and coupling constraint, you can find more information in Section 4.1.1 and Section 4.1.2.



new instances of categories and relations, and also use mutual exclusive relationship between predicates to ensure high precision. We implement CSEAL from scratch, and add our version coupled constraints and flexible relation extraction to CSEAL. We discuss more design and implementation detail about CSEAL in Chapter 5.

- **Coupled Morphological Classifier:** CMC is a set of binary L_2-regularized logistic regression models, each one is for a category specifically. CMC use morphological features mentioned above to classify that does a noun phrase belong to specific category. The noun phrases were provided by other components, CMC examines these noun phrases such as insurance, and promote noun phrases as candidate instances to ensure overall high accuracy.

- **Rule Learner:** Rule Learner is a first-order relational learner which learns probabilistic Horn clauses, NELL runs Rule Learner every ten iterations for inferring new relation instances from known relation instances in knowledge base.

Chapter 3

Framework

In this chapter, we introduce ChNELL, a language learner focuses on Chinese knowledge collection. ChNELL imitates the never-ending spirit from NELL, and adapts language-dependent part to the Chinese language.

The architecture of ChNELL shows that ChNELL is flexible for cooperating with other Chinese knowledge extractors, and easy to migrate sub-components of ChNELL to NELL for further collaboration. We introduce the overview of system, and define system inputs, outputs and system behavior.

3.1 Central Ideals From NELL

In this section, we introduce two central ideals in NELL, system design around these principles enables to easily integrated with NELL and other knowledge extractors follow the same rules.

3.1.1 Never Ending Learning

> *Humans learn many things, for years, and become better learners over time. Why not Machines? By Tom Mitchell.*

As the name describes, NELL was designed as a step toward never-ending learning, a system can continuously learn knowledge and improve itself. In never ending learning, NELL considers an iteration (or round) as a unit. Every iteration is a learning process based on the outputs from last iteration. This is an ideal model for knowledge extraction, because extracting all potential knowledge by performing a single stage algorithm is unrealistic. In this model, the goal of the system is to repeatedly grow knowledge base with high confidence knowledge.

3.1.2 Collaboration of Learners

Information retrieval or knowledge extraction are sophisticated fields in computer sciences. According to different principles or properties, many approaches have been proposed to deal with these problems, Collaboration of subsystem components that make uncorrelated errors can significantly improve overall precision, because the same candidate fact have been proposed by different approaches, we can have quite confident on that fact.

3.2 ChNELL

We introduce our system architecture, inputs, outputs and system behavior in this section.

3.2.1 System Behavior

Our approach is designed base on the two points above. The knowledge base is grown incrementally by knowledge which is extracted iteratively from subsystem components. The subsystem components also receive trusted knowledge from last iteration as inputs. Initial knowledge base is defined as ontology which consists of a set of predicates, predicates belong to one of categories or relations. And a handful of examples for each predicate, e.g. "台北" is one of example instance for predicate "城市".

In every iteration, inputs of subsystem components are not only trusted knowledge (or called facts), but also external resources. The external resources might be prepossessing data, results of instant query from search engine or any external corpus, it depends on subsystem component requirements.

The outputs of subsystem components are proposed candidate facts by their own. A promoting mechanism are applied on these proposed candidate facts. After promoting, the promoted candidates are facts, and ChNELL throws away the rest.

3.2.2 System Architecture

We show the system architecture in Figure 3.1.

We have two subsystem components. The first is Coupled Chinese Pattern Learner (CCPL), CCPL is a pattern based knowledge extractor which focus on relations between noun phrase and verb phrase in Chinese. The detail about CCPL will be depicted in Chapter 4. The second is Coupled Set Expander for Any Language (CSEAL), CSEAL is a wrapper based extractor which focuses on semi structures of context in documents. More detail can be found in Chapter 5.

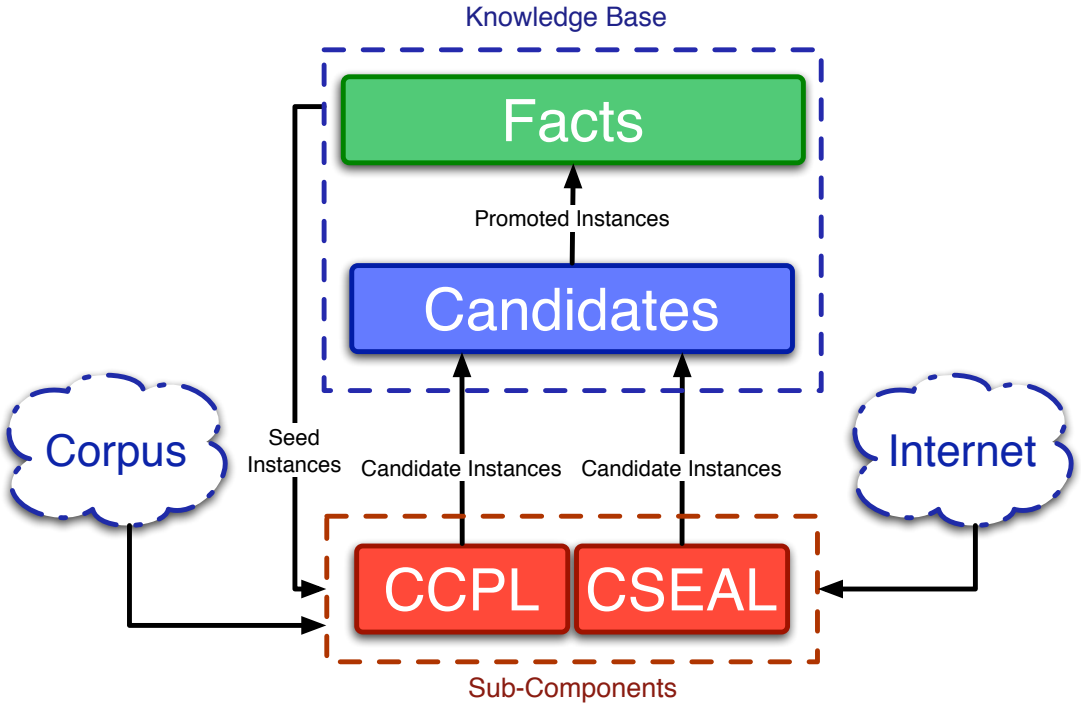


Figure 3.1: System Architecture of ChNELL. ChNELL has two subsystem components Coupled Chinese Pattern Learner(CCPL) and Coupled Set Expander for Any Language(CSEAL), introductions of CCPL and CSEAL are in chapter 4 and 5. The Arrows in figure represent the flow of data.

Chapter 4

Coupled Chinese Pattern Learner

This chapter proposes an approach which automatically extract Chinese concept knowledge from unstructured free text by textual patterns. Chinese concept knowledge includes various categories of entities extracted from unary textual patterns (e.g. “國家”, “歌手”), and relations entities extracted from binary textual patterns (e.g. “國家擁有城市”, “歌手的專輯”).

Language pattern is defined to find valid textual instances which can be used to find additional patterns. We summarize the types of coupling to a general definition. Furthermore, we proposed a additional coupling type in addition to three coupling type proposed by NELL. The coupling constraints can effectively detect extracted instances which are not quite confident, so we can improve the accuracy by excluding these instances.

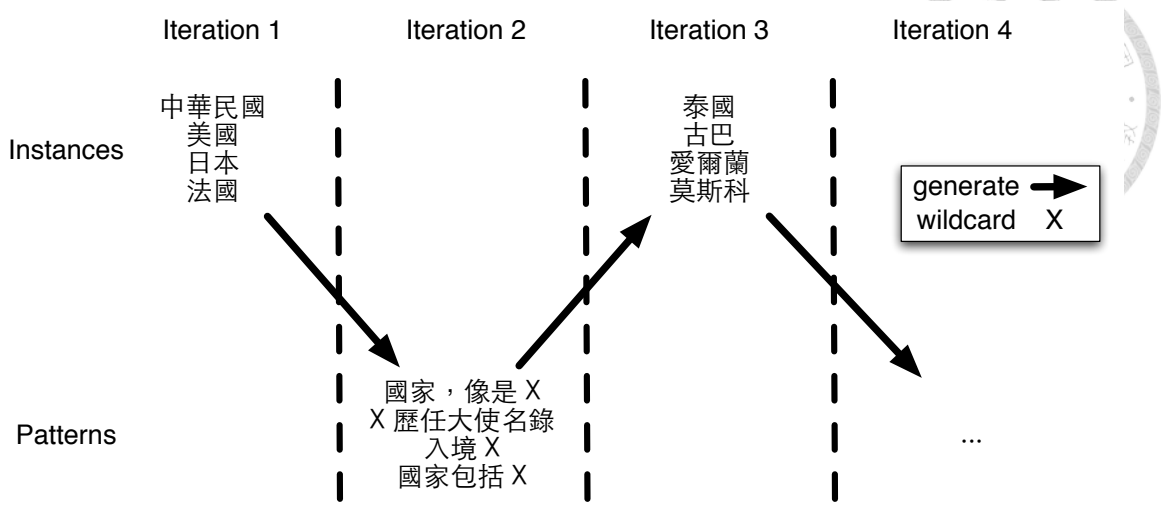


Figure 4.1: An example of bootstrapping learning by textual extraction patterns in Chinese.

4.1 Bootstrapped Learning

This approach applies textual patterns to extract the instances from free text, and then extracts new patterns by instances which are extracted more than chance. Finally, interactively continue to next iteration. The working flow is show in Figure 4.1.

Definition of textual patterns is a literal string with one or two wild-cards. Number of wild-cards is based on the pattern is a category pattern or relation pattern. Category patterns only have one wild-cards, because category patterns are used to look for concepts which are belong to a category. On the other hand, relation patterns have two wild-cards for describing a relation exists between two concepts. In textual patterns, wild-cards represent the unknown target instances.

For example, "X **的外交官**" is a pattern with literal string "**的外交官**" and wild-card X. If we apply this pattern to sentence "**駐朝鮮的外交官不理會朝鮮撤離警告**", and then "**朝鮮**" will be extracted as a country name.

This an example for extracting instances from patterns, in the other direction, we show next example for extracting patterns from instances. Now we know "**朝鮮**" is an instance of countries, and then we apply "**朝鮮**" to sentence "**朝鮮提供經濟援助**" to derive candidate pattern "X **提供經濟援助**".

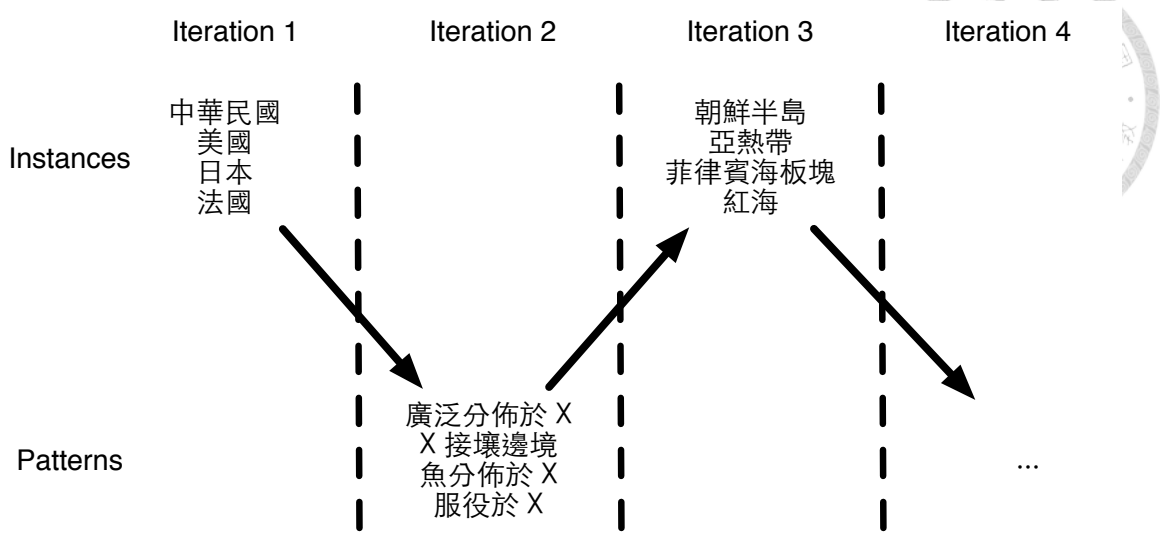


Figure 4.2: Semantic drift in bootstrapping learning.

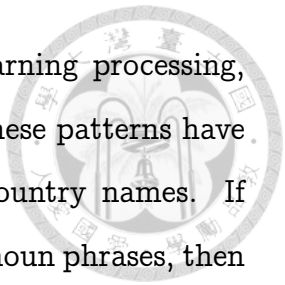
The same scheme also works for relation extraction. The relation pattern "X **的首都** Y" has two wild-cards. We apply this pattern to sentence "**拉丁人佔領了拜占庭的首都君士坦丁堡**", and then we can obtain ("**拜占庭**", "**君士坦丁堡**") pair of instance which can describe the relation "**國家擁有城市**".

If pattern extracting for categories and relations are well defined and tractable, then this bootstrapped learning for extracting Chinese knowledge is feasible. How to extract valid instances and patterns are depicted in Section 4.2.2.

The pattern-based approach is an effective method for information extraction. In articles, we notice that the noun phrase represents country name in sentence can always be replaced by other country names. We conclude that the country category shares the same prefix or suffix literal string, or more precisely, they share the same set of verbs.

4.1.1 Semantic Drift

Progression of bootstrapping have been expected to improve correctness by propagation of correct labels, but somehow when error libeling are present with few iterations propagation, exactly the opposite case happened. The error labeling accumulated in result. The problem is called "semantic drift"[4].



An example for semantic drift shows in Figure 4.2. In learning processing, textual patterns which learned in iteration 1 are ambiguous. These patterns have opportunities to co-occur with noun phrases which are not country names. If bootrapping learning proceeded according to these non-country noun phrases, then "is a country" predicate will involve unrelated. The predicate is drifted.

In an ideal case, the country name "臺灣" can be replaced to any other country name like "美國", "日本" in sentence "入境台灣有兩種通關方式", and the sentence is still valid after replacing. There is always an exception, some verbs are common used for many concepts. The location hierarchy is the most fallible case. People usually consider locations as concepts for many different types, countries, states, cities and districts. The verbs which related to locations are easily confusing. For example, It is still a feasible Chinese sentence when the country name in "產地是在法國" is replaced by city name such as "台中" or name of province of china such as "福建" or "湖北", but in fact, country and city are different categories. When these kind of errors occurred and were accepted by system, our system may seek for more patterns based on these wrong instances as seed example in next iteration, and then error in labeling accumulated.

4.1.2 Coupled Constraints

To prevent semantic drift in bootstrapping learning processing, an effective strategy proposed by NELL is adding constraints to filter out inappropriate results. The NELL proposed three types of coupling constraints. This section summarize the coupling constraints to a general formula, furthermore, introduce an additional type of coupling constraint originally. This additional coupling constraint is not only can constrain the learning problem, but also propagate result by inference.

The main idea of coupled constraint is, there is a limited relationship between outputs of different functions which share a certain degree of input domain. The definition of the general type of couplings is shown below.

Definition 1. For a set of functions, a constraint is defined as a binary relation

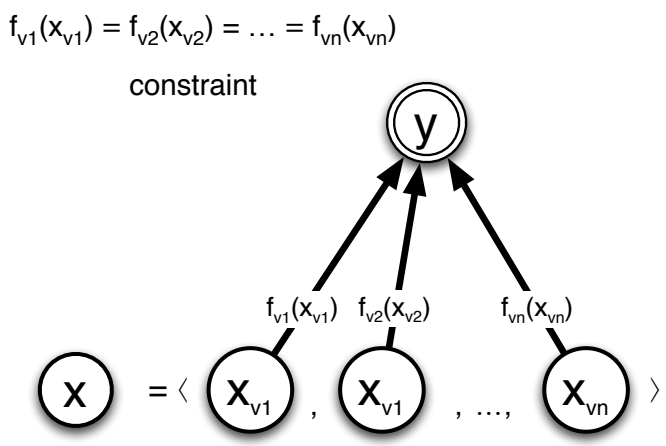


Figure 4.3: The multi-view-agreement constraint. The single stroke circles are inputs and outputs. The double stroke circle is output which is constrained. The arrows represents processing by function.

or a ternary relation among the ranges of functions.

We can enforce ranges of functions to belong to the relation for satisfying constraint. The Constraints can be divided into four categories according to overlapping of the inputs of functions which are restricted.

The introduction of coupling constraints start from the most overlapping inputs of restricted function.

1. **Multi-View-Agreement Constraint:** Functions are restricted by Multi-View-Agreement Constraint if their inputs can be partitioned into multiple views.

   For a input set X, $\forall x \in X$, x can be represented by multiple views such that $x = \langle x_{v1}, x_{v2}, x_{v3}, ... \rangle$. For each view, exist a corresponding function $f_{vi} \in F$, $f_{vi} : V_i \rightarrow Y$. The multi-view-agreement constraint is defined as a binary relation R on Y.

   For example, if binary relation R is simply equality, then each view of the same input x must be equal for multi-view-agreement constraint hold. The example is show in Figure 4.3.

2. **Output Constraint:** The output Constraint is for functions which accept

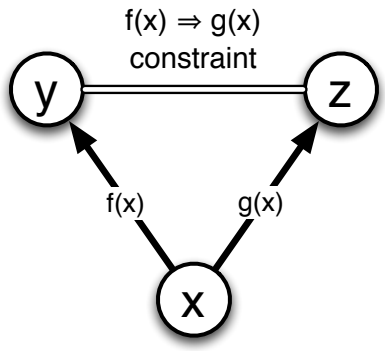


Figure 4.4: The output constraint. The single stroke circles are inputs and outputs. The double stroke line is constraint connection between outputs. The arrows represents processing by function.

the same input x. Given two functions $f : X \to Y$ and $g : X \to Z$, the output constraint is defined as a binary relation R between Y and Z.

For example which is show in Figure 4.4. If f and g are boolean functions and relation R is logical implication such that $\forall x \in X$, $f(x) \implies g(x)$, then we could enforce $g(x) = 1$ whenever $f(x) = 1$.

3. **Compositional Constraint:** The compositional constraint is for the functions which share part of inputs. Given two functions $f : X_1 \to Y$ and $g : X_1 \times X_2 \to Z$, f and g share one of input x_1. The compositional constraint is defined as a binary relation R between Y and Z.

   An example of compositional constraint is that set relation R as logical implication, function f can be used as type check pre-condition of function g by $\forall x_1 \in X_1$, $\forall x_2 \in X_2$, $g(x_1, x_2) \implies f(x_1)$. We show this in Figure 4.5.

4. **Helping Function Constraint:** The helping function constraint is for the functions their inputs are totally irrelevant, but we have external knowledge for these functions. We can add a additional pre-defined helping function based on external knowledge to connect these inputs. For a function $f : X_A \times X_B \to Y$ and helping function $g : X_A \times X_A \to Z$, the helping function constraint is defined as a ternary relation T among Y and Z.

   For example show in Figure 4.6. We assume f is a function can determine

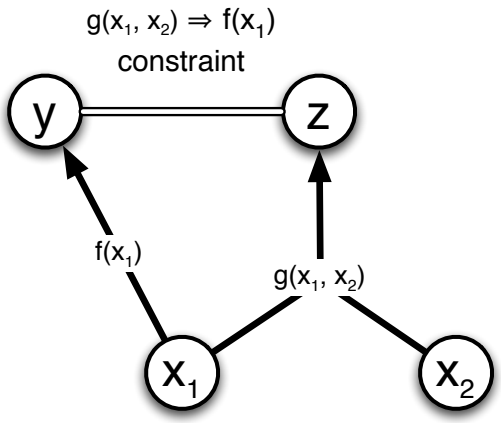


Figure 4.5: The conpositional constraint. The single stroke circles are inputs and outputs. The double stroke line is constraint connection between outputs. The arrows represents processing by function. The function g has two inputs.

whether pair of two inputs x_a, x_b are valid or not, and we have external knowledge to know, for the input pair, does not exist the same x_a can present in different functions, then g can be defined as a function which determine whether pair of inputs are not equal, this can be done by exclusive disjunction such that $z = g(x_{a1}, x_{a2}) = x_{a1} \oplus x_{a2}$. By these assumptions, we know every pair of valid input for f, they have different x_a, ternary relation $T = \{(y_1, y_2, z) : (y_1 \& y_2) \implies z\}$ makes this constraint hold.

The ternary relation T can consider as not only a constraint, it is a inference rule depends on what help function can provide. When T is satisfied, we can propagate instance to generate new instance. For example, the help function is "symmetric or not" and it is always be true for some relation such like "兩人是隊友". When we have learned a pair of new instance ("林書豪", "詹姆士哈登"), then pair ("詹姆士哈登", "林書豪") is also a valid instance of relation "兩人是隊友".

4.2 Concepts Extraction

The CCPL extracts Chinese knowledge by pattern-based approach. There is not always valid noun phrase in position of wildcat when pattern is matched, and

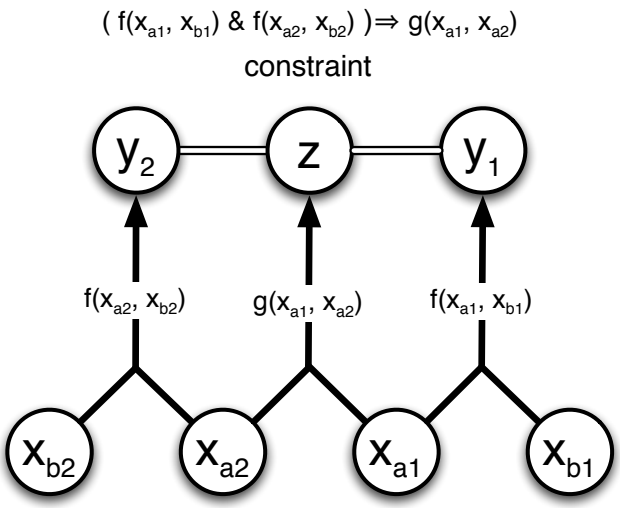


Figure 4.6: The helping function constraint. The single stroke circles are inputs and outputs. The double stroke line is constraint connection between outputs. The arrows represents processing by function. All functions in this graph have two inputs.

also not all surrounding literal strings next to matched instance are proper candidates for patterns. So question is, how to retrieve valid instances and patterns?

CCPL recognizes instance when a pattern is matched by there is at least one noun in that position of wild card, and recognizes pattern when a instance is matched by there is a verb previous to or following the instance.

To find the noun and verb, part-of-speech tags can help to identify valid instances and patterns.

4.2.1 Difficulties of Chinese Concept Learning

With helping of part-of-speech tags, Chinese still suffers from many difficulties compares with other languages.

1. **Segmentation:** English use some form of Latin alphabet, the space is naturally a word delimiter. Languages like Chinese and Japanese do not have this property, additional segmentation for preprocessing of Chinese is inevitable.

2. **Part-of-speech tagger:** In linguistic typology, languages can be simply

divided into two categories base on morpheme-per-word ratio[1], synthetic language and isolating language.

Obviously Chinese belongs to the second one, that means that inflections are not helpful to indicate grammatical relationships in Chinese. Chinese part-of-speech tagger suffers more difficulties than synthetic languages.

3. **New Named Entity:** In the field of natural language processing (NLP), unknown words handling is very crucial and difficult. Knowledge extraction are usually dealing with new words the system does not know it yet, so named entity recognition in Chinese is a big challenge.

4.2.2 Valid Instance and Pattern

We use part-of-speech tag of Chinese token to restrict instances which apparently are noun phrases, and restrict patterns which comply some grammars which we predefined by statistical results.

After segmentation and part-of-speech tagging, the sentences are consisted of a sequence of tokens. The tokens represent a single meaning and contain few Chinese words.[2] In our assumption, we do not take synonym[3] and variant Chinese character[4] into account, in other words, we consider differentiation of tokens in character level.

When an instance/pattern is matched, CCPL identify valid corresponding pattern/instance by following rules. The rules are summarized to finite state machines in figures.

- **Category Instance:** When the promoted category pattern matched, CCPL

[1] How many morphemes in a word.

[2] In extreme case, token may only contains one Chinese word.

[3] For example, "台大資訊" is a abbreviation of "國立台灣大學資訊工程學系", but we consider them as different concepts.

[4] Variant Chinese character is a special problem in Chinese language, which means Chinese characters that are homophones and synonyms in character level. For example, Chinese people usually consider "臺" and "台" are the same, no matter in meaning or in pronouncing. If we consider the difference between Traditional Chinese and Simplified Chinese, then problem become more complicated.

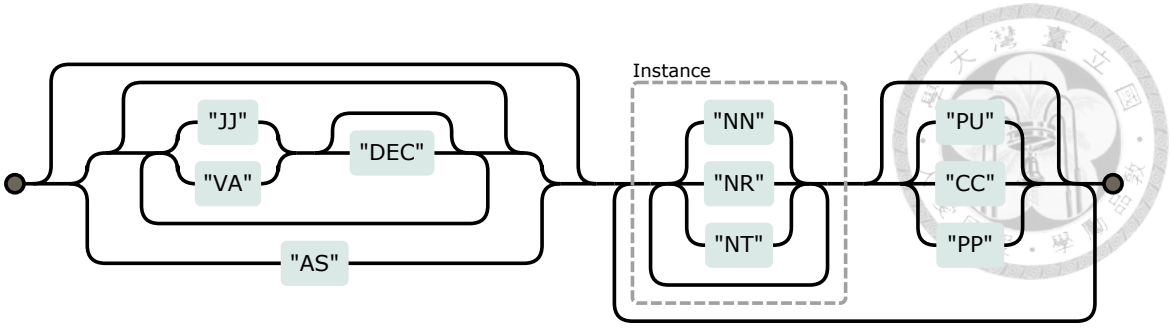


Figure 4.7: Finite-state machine of capturing of category instances. "Instance" gray box represents the noun phrase extracted.

identify a noun phrase from the location of wild-card of matched pattern. The sentence consisted of a sequence of tokens, then the exact problem has transferred to "what kind of tokens CCPL keeps to reassemble to a noun phrase?" or "when CCPL stop to retrieve more tokens?".

The noun phrase is sequences of adjective (tagged VA or JJ) [5] with a complementizer (tagged DEC) and sequences of noun (tagged NN or NR or NT) (e.g., "英勇強壯的美國隊長"), or an aspect particle (tagged AS) followed by sequences of noun (e.g., "了中國"). CCPL also permits coordinating conjunctions (tagged CC) or punctuations [6] between noun phrases.

For example, "臺灣 _NR 和 _CC 美國 _NR" and "台北 _NR 、_PU 台中 _NR 、_PU 高雄 _NR" are going to extract { 臺灣, 美國 } and { 台北, 台中, 高雄 }, all nouns in sentences.

According to the above rules, we summarize to a finite state machine show in Figure 4.7.

- **Category Pattern:** When the promoted category instance has been found, CCPL looks forward for candidate patterns from matching instance if there exist arbitrary adverbs (tagged AD) followed by at least one verb (tagged VV or VC) and a optional preposition (tagged PP[7]) (e.g., "曾就讀於 X").

[5]The Chinese taggers use the Penn Chinese Treebank [16] tag set as part-of-speech name abbreviations. The documentation of Penn Chinese Treebank. http://www.cis.upenn.edu/~chinese/posguide.3rd.ch.pdf

[6]A special punctuation called '頓號' in Chinese, it is punctuation used to set off items in a series. And no corresponding punctuation in English.

[7]Originally preposition is tagging by P in Penn Chinese Treebank, but P is a prefix of other part-

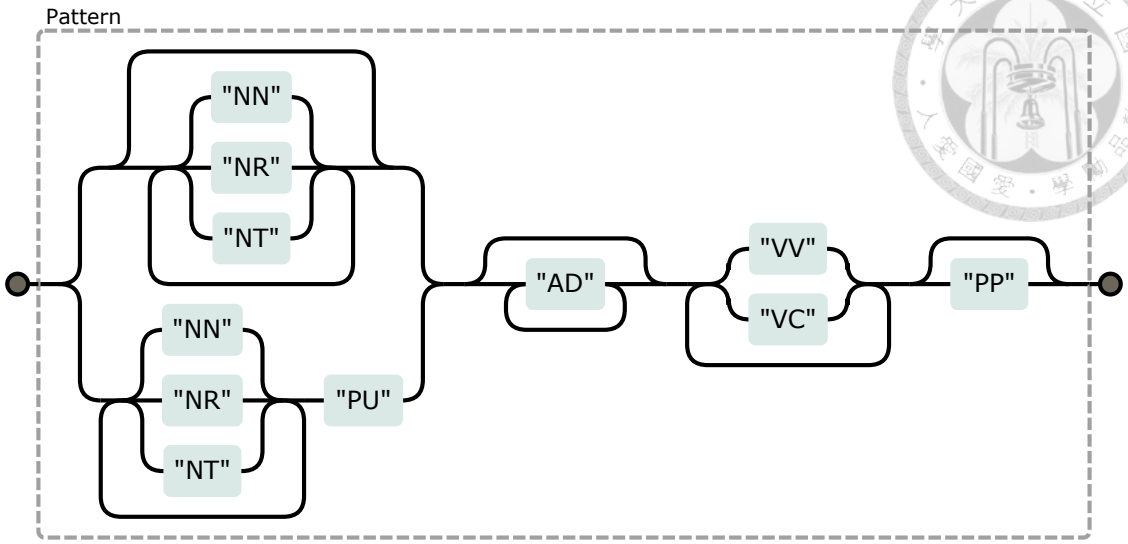


Figure 4.8: Finite-state machine of capturing of category prefix patterns. "Pattern" gray box represents the pattern extracted.

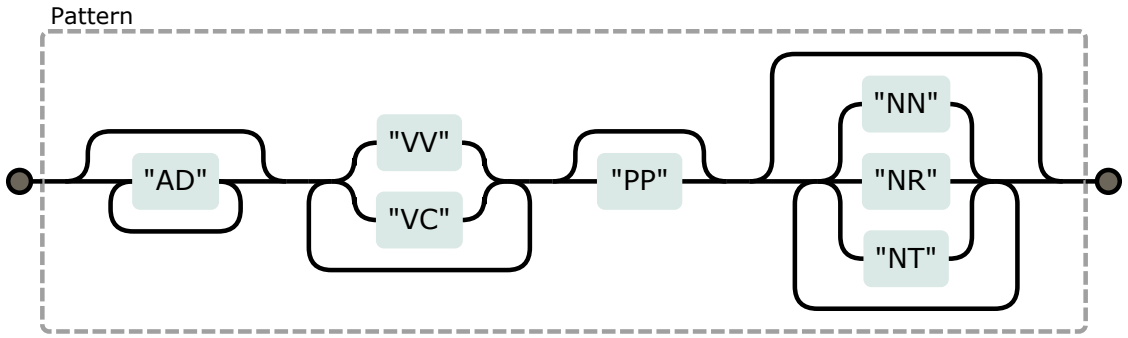


Figure 4.9: Finite-state machine of capturing of category suffix patterns. "Pattern" gray box represents the pattern extracted.

CCPL also allows an optional noun phrase at the forefront (e.g., "歌曲收錄於 X") or an optional noun phrase followed by a punctuation (e.g., "學校，位於 X").

CCPL looks backward for candidate patterns if there are arbitrary adverbs, verbs, preposition and an optional noun phrase (e.g., "X 進軍好萊塢").

The figures of finite state machine of grammar pattern are shown in Figure 4.8 and Figure 4.9.

- **Relation Instance:** When two-wildcard pattern is matched (e.g., "X 的

of-speechs. When we try to recognize whether pattern exist in sentence, this will cause ambiguous circumstance, so P had been converted to PP. The same situation we have M for measure word, and we use MW in CCPL.

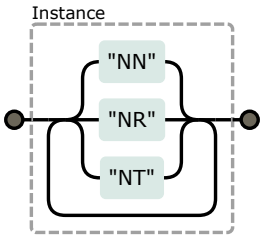


Figure 4.10: Finite-state machine of capturing of relation instances. "Instance" gray box represents the noun phrase extracted.

	At Least Words	At Most Words
Category:Instance	2	
Category:Pattern		
Relation:Instance	2	
Relation:Pattern	3	20

Table 4.1: Restrictions on length of the valid instance/patterns, empty fields represent no restrictions.

城鎮，位於該國東部，距離首都 Y"), CCPL looks for noun phrases from both forward, backward directions by simply concatenated noun tokens as long as possible. The rule used is the same as category instance capturing.

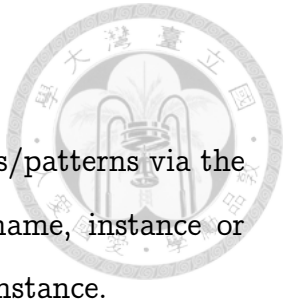
A finite-state machine for relation instances capturing is shown in Figure 4.10.

- **Relation Pattern:** When both relation instances are found in a sentence, intervening phrases between two matched relation instances are extracted as relation candidate pattern. The only restriction is that length of pattern required to be within a certain range.

After segmentation, noun phrases in Chinese usually consist of a sequence of noun tokens. The tags for single token like the proper noun tag NR or the common noun tag NN do not provide any information to the whole noun phrase for validity. The validity of noun phrase is simply restricted by the length of noun phrase

The restriction is shown in Table 4.1.

4.3 Concepts Selection



After concepts extraction, CCPL obtains candidate instances/patterns via the five-tuple format which consists of relation arity[8], predicate name, instance or pattern flag[9], promoted instance/pattern and learned pattern/instance.

4.3.1 Filtering

In filtering stage, we show four concrete examples of the output constraint, the compositional constraint and the helping function constraint.

Mutual Exclusion

CCPL applies a special case of the output constraint which show mutual exclusion property. Each pair of predicates are mutually exclusive by default, we can remove mutual exclusion property among predicates by adding mutex exception predicate. The mutual exclusive predicates can not be satisfied by the same input simultaneously, so we can filter out common candidates which co-occur with more than one predicates.

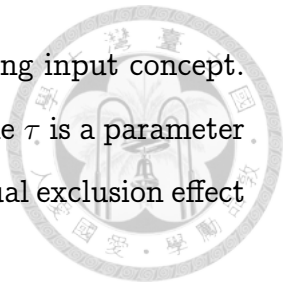
More precisely, a candidate instance is accepted if the number of times it co-occurs with a promoted pattern is at least τ times more than the number of times it co-occurs with patterns from mutually exclusive predicates.

Given m concepts $C_1, C_2, ..., C_m$, suppose that every concept $C_i, 1 \leq i \leq m$ has n corresponding promoted patterns $P_{i1}, P_{i2}, ..., P_{in}$, and we find j co-occur candidate instances $I_1, I_2, ...I_j$ based on $m \times n$ promoted patterns. The formula of accepted candidate instances of the concept C_i is:

$$candidates(C_i) = \{I_k : 1 \leq k \leq j, \forall z, z \neq i, (\sum_{x=1}^{n} co{-}oc(I_k, P_{ix})) * \tau \geq \sum_{y=1}^{n} co{-}oc(I_k, P_{zy})\}$$

[8]The arity means extracted instance/pattern belongs to unary predicate (category) or binary predicate (relation).

[9]A flag indicates the tuple is learning instance from patterns or learning pattern from instances. This flag affects the types of the rest elements in the tuple.



The $candidates$ function outputs valid instances corresponding input concept. The $co-oc$ function returns instance-pattern co-occur times. The τ is a parameter which control how strict mutual exclusive constraint is, and mutual exclusion effect will be more strict when τ increasing.

The same mutual exclusive manner apply on process which learn candidate pattern by promoted instances, the formula can be obtained by simply swapping instance and pattern.

Type Check

Type-checking is an application of compositional constraint. The type-checking constraint is also applied in filtering phase, CCPL filter out invalid relation instances by type-checking constraint. We provide auxiliary attributes for relation predicates, auxiliary attributes include the domain and the range fields indicate that the relation is connected with which types of category instances.

The more detail about format of example seeds are shown in Section 5.5.

After extracting a new relation candidate instance, CCPL checks both arguments whether belong to their corresponding concepts and already exists in our knowledge base. CCPL rejects relation instances if both arguments are unknown noun phrases.

Mathematically Binary Relation

CCPL learns the binary relation between two categories, binary relations have some traditional properties based on reality.

For example, irreflexive relation R is defined as $\forall x \in X,\ \nexists (x, x) \in R$. "兩人是隊友" is a irreflexive relation because an athlete is not a teammate of him/herself.

We also can infer relations by binary relation properties. "兩人是隊友" relation is not only a irreflexive relation, but also a symmetric relation. When CCPL learned a "兩人是隊友" relation instance such as ("林書豪", "詹姆士哈登"), there is a instance ("詹姆士哈登", "林書豪") exists in "兩人是隊友" relation for sure. The other properties

such as inverse and transitive are also helpful for inference.

For inverse property, $\forall x, y \in X$, $(x, y) \in R \implies (y, x) \in R$, a relation may has a corresponding inverse relation.

"國家擁有城市" and "城市位在國家" are inverse relationship, because they describe the same idea. The reason of using two relations to describe the same idea is that people can express the same predicate in different order. For example, sentences "台灣的首都是台北市" and "台北市，位於台灣北部" are both related to the same set of predicates "台北市是一座城市，台灣是一個國家，台北市是台灣的一部分".

When CCPL learned relation instance ("台灣", "台北") in a relation which has inverse property, we generate corresponding relation instance ("台北", "台灣") in the inverse relation. This is way to propagate instances by the help function constraint.

For a concrete example of transitive relation, transitive property is defined as $\forall x, y, z \in X$, $(x, y), (y, z) \in R \implies (x, z) \in R$. The relation "朝代之後的朝代" indicates the order of dynasties. When CCPL learned ("元朝", "明朝") and ("明朝", "清朝") already exists in knowledge base, we can infer ("元朝", "清朝") instance for relation "朝代之後的朝代".

Mapping

The last one is the mapping type constraint by the helping function. The purpose of this constraint is that based on external knowledge to restrict relation propagation. The mapping types include one-to-one, one-to-many, many-to-one and many-to-many.

- **One-to-one:** A relation with one-to-one mapping type means that for all instances of the relation, do not exist the same category instance in domain or range.

  For the relation "國家的首都", the valid relation instances such as ("台灣", "台北"), ("日本", "東京"), ("美利堅合眾國", "華盛頓特區"). If CCPL learned a relation instance ("台灣", "高雄"), then we know that one of ("台灣", "台北") and ("台

灣”, “高雄”) must be wrong, because only one national capital for countries. CCPL consider higher confidence[10] instance as correct one and reject the others.

- **One-to-many and many-to-one:** The one-to-many mapping type relation such as “國家擁有城市”. A country may contains many cities, in other words, a city does not belong to multiple countries. CCPL applies the same rejection principle when it learned relation instances their range field are deduplicating. An example such as (“美國”, “紐約”) and (“加拿大”, “紐約”). CCPL rejects lower confidence instance.

  The difference between one-to-many and many-to-one mapping type is that the restricted field is on domain.

- **Many-to-many:** Many-to-Many mapping type is the simplest, which means no restriction on both domain and range.

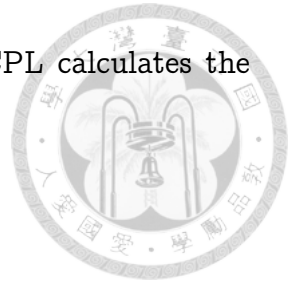
4.3.2 Ranking

After applying coupled constraints, CCPL ranks candidates by frequency of co-occurs. First CCPL ignores candidates which only co-occur with only one promoter. The candidate-promoter relationship is such as instance-pattern relationship. The candidate is a pattern when the promoter is a instance, vice versa. The candidates are ranked by different strategies based on candidates are instances or patterns.

Ranking Instance Candidates

CCPL is ranking by multiple fields sorting. The multiple fields sorting considers first field for sorting, when number of first fields are the same, considers second fields and so on.

[10] Every instances in CCPL have a confidence value between 0.5 to 1. The detail of confidence value is shown in Section 4.3.3.



The first field is diversity. For a candidate instance i, CCPL calculates the diversity as following:

$$diversity(i) = |\{p : p \in P_i\}|$$

The P_i is set of patterns which co-occurs with instance i for the predicate under consideration.

The second field is the sum of number of patterns which the instance i co-occurs with, as following:

$$frequency(i) = \sum_{p \in P_i} count(i, p)$$

The function $count(i, p)$ is the number of times the pattern p co-occurs with instance i in corpus.

We consider that the diversity of promoters is more important than the co-occurs times. The reason is that when a instance with high co-occurs times and from very few patterns, it means only few patterns agree with this instance but agreement is strong. The instance-pattern co-occur pair might be a convention usage or unfiltered template literal strings in this case.

The diversity also represents a example of multi-view-agreement constraint.

Ranking Pattern Candidates

There are three fields are used for ranking patterns in CCPL. The first and the last fields are the same formulas $diversity(i)$ and $frequency(i)$ which are in ranking candidate instances. The modification can be done by swapping variable instance i with pattern p.

The formula are listed as following.

$$diversity(p) = |\{i : i \in I_p\}|$$

$$frequency(p) = \sum_{i \in I_p} count(i, p)$$

An additional field $precision(p)$ is second considerable field for ranking patterns. We show the formula as following:

$$precision(p) = \frac{\sum_{i \in I} count(i, p)}{count(p)}$$

The I is a set of promoted instance for the predicate under consideration, and the function $count(p)$ is the number of times pattern p occurs in the corpus.

The function $precision(p)$ is proposed by NELL originally, for calculating how important this pattern p related to considerable predicate. Our priority is still the same, $diversity > precision > frequency$, the most important is the diversity, and the importance, finally the frequency.

4.3.3 Promotion

CCPL simply promotes the top n category candidates and m pattern candidates based on corpus size. We can dynamically adjust this threshold. The reasonable number for our testing data are $n = 20$ and $m = 5$. CCPL only promote candidates which do not already exist in mutual exclusive predicates in knowledge base.

There is a score for every promoted instance and pattern which indicates level of trust. We follow the formula from NELL, confidence $= 1 - 0.5^c$, where c is the number of different promoters co-occur with this candidate.

4.4 CCPL Algorithm

In this section, we show Coupled Chinese Pattern Learner algorithm in Algorithm 1.

Algorithm 1 Coupled Chinese Pattern Learner Algorithm

Input: An ontology $\hat{O}$ and preprocessing corpus $\hat{C}$.[11]
Output: Expanded ontology.
for iteration $i = 1, 2, 3, ..., \infty$ **do**
  for sentence $s \in \hat{C}$ **do**
    for predicate $p \in \hat{O}$ **do**
      Extract new candidate instances/patterns from s by promoted seeds in p ;
    end for
  end for
  Filter out candidates that violate the coupling constraints ;
  Rank candidate instances/patterns ;
  Promote the top candidate instances/patterns as new ontology $\hat{O}_{new}$;
  $\hat{O} = \hat{O} \cup \hat{O}_{new}$
end for

Chapter 5

Coupled Set Expander for Any Language

In this chapter, we implement a sub-component CSEAL of NELL to couple with CCPL. The sub-component is a wrapper-based extractor, which can accept semi-strcuture input and then extracts hidden concepts. We also apply coupling constraints to CSEAL for improving accuracy. According to the experience from NELL, we mitigate the constraint on relation extraction to increase the probability of extracting candidate instances of relation. Finally, the experiment considers Chinese wikipedia as input corpus of CCPL. CCPL cooperates with CSEAL to show that this approach with enabling coupled constraints had achieved great accuracy.

5.1 SEAL

Set Expander for Any Language (SEAL) is a text mining component based on wrapper induction. SEAL accepts seeds and semi-structure text documents as input, and to extract targets which with the similar structure around. SEAL is language-independent because it benefits on texts which are embedded in similar structure, the structure is not only about sentence level, but also all text in documents, even markup language.

A wrapper is a template for generating semi-structure document. People usually use template to handle variant data for generating documents, and in these documents, template wraps presenting data. SEAL processes the flow reversely. SEAL tries to find out hidden wrappers which are related to seeds. The wrappers which have been found are document specific, because there is no guarantee that two different documents are generated by the same template.

Based on semi-structure property of input, CSEAL is suitable for processing web pages. Nowadays, people store valuable information in databases, and render by HTML[1]. SEAL can detect these information efficiently, especially for information in a table or in a list.

For example, if we want to find the members of category "學系", and we already know several member instances "電機工程學系", "資訊網路與多媒體研究所", "醫學系" and "法律學系". The first step is to find the web pages which contain these keywords. One of simple ways for getting related web pages is querying search engine. The concatenation of known seeds, such as "電機工程學系 資訊網路與多媒體研究所 醫學系 法律學系", is used as search keywords. The first link (URL: http://www.ntu.edu.tw/academics/academics.html) of result of searching is the most relevant web page, we use it as example.

The page contains a list of all departments in National Taiwan University. We show the search result and the page in Figure 5.1 and 5.2.

[1]Hyper Text Markup Language, it is the main markup language for displaying structure content in web page.

Figure 5.1: The result from Google search engine with keywords "**電機工程學系 資訊網路與多媒體研究所 醫學系 法律學系**".

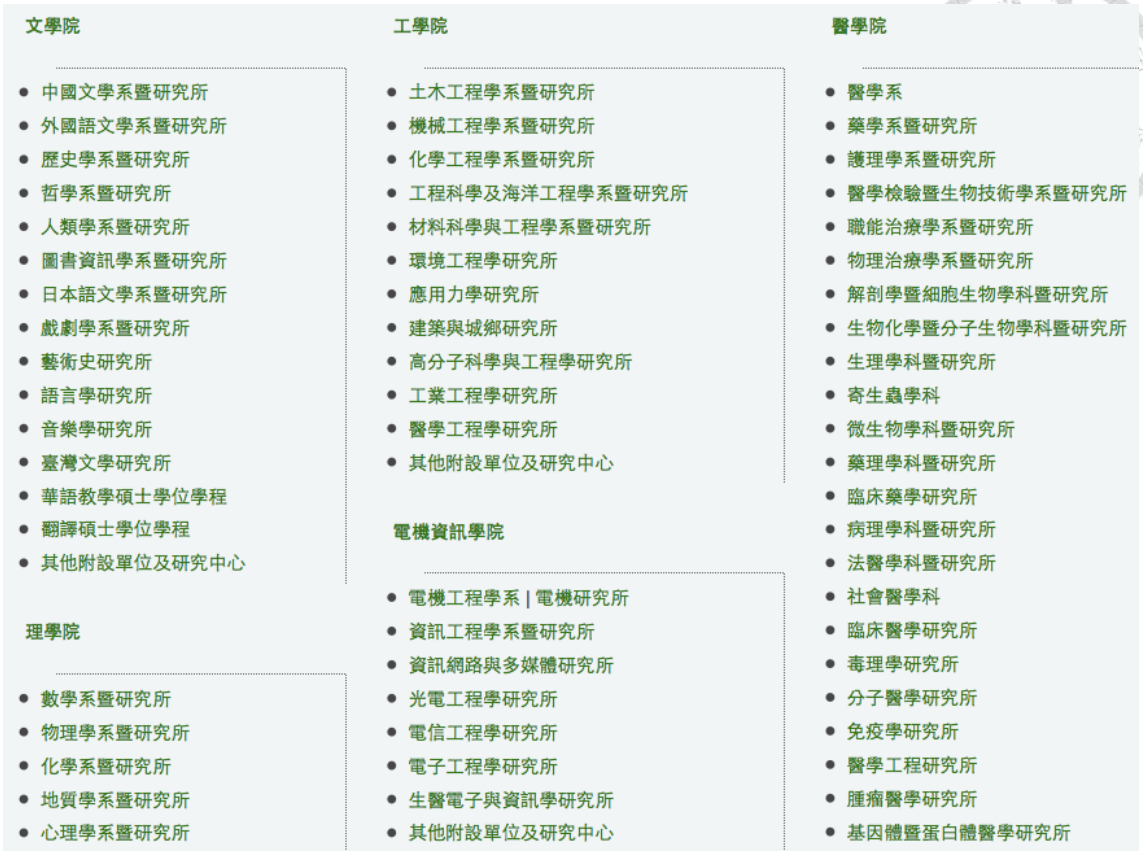


Figure 5.2: The page of list of departments in National Taiwan University.

By examining the page source, we discover that the same instance will not only appear once in page. For example, "資訊網路與多媒體研究所" appears twice in

```
<li>
<a href="http://www.inm.ntu.edu.tw" title=" 資訊網路與多媒體研究所">資訊網路與多媒體研究所</a>
</li>
```

We show all these HTML code fragments in Table 5.1.

SEAL can induce two wrappers from these fragments. The wrapper describes surrounding text by a pair of prefix string and suffix string. We concatenate the prefix, the wild-card and the suffix to generate a wrapper by regular expression, where (.+?)[2] represents the wild-card.

1. / title="(.+?)"/

[2]The regular expression suffers from inconsistent versions. The +? syntax is non-greedy version of + in our system.

ntu.edu.tw" title="資訊網路與多媒體研究所"> 資訊網路與
網路與多媒體研究所">資訊網路與多媒體研究所
<a title="電機工程學系" href="http://www.ee.ntu
ee.ntu.edu.tw">電機工程學系 | <a title=" 電機研究所"
med.ntu.edu.tw/main.php?Page=A1" title="醫學系"> 醫學系
med.ntu.edu.tw/main.php?Page=A1" title=" 醫學系">醫學系
law.ntu.edu.tw/" title="法律學系"> 法律學系
law.ntu.edu.tw/" title=" 法律學系">法律學系

Table 5.1: The HTML code fragments which contain keywords "電機工程學系 資訊網路與多媒體研究所 醫學系 法律學系". The keywords represent in red.

2. /">(.+?)/

We extract 166 candidates by each of the two wrappers, and it is not all of 332 candidates are correct, but this approach indeed has high probability to extract target instances.

5.1.1 Radix Tree

How can SEAL induce these wrappers? When target instance is matched in the source of pages, SEAL builds two radix trees[14] (also called patricia trie) for each page. One for recording reverse prefix text and one for recording suffix text, and then to find longest common pairs of prefix and suffix. The "common" means that each of seed instances is at least counted once in each of prefix/suffix to satisfy the agreement that "every seed instance in the same predicate must surrounding by the same prefix/suffix at least once".

There is a parameter l for adjusting the minimum length of prefix/suffix texts, in our experiment, we set $l = 2$ characters.

5.1.2 Ranking of SEAL

After extracting, SEAL also has to rank candidate. SEAL builds a graph which consists of nodes represented the documents, wrappers and instances. The edges between nodes represent relations such as Extracts, Be-Extracted, Contains and Be-Contained. The degree of a node represents agreement of the node, and

therefore SEAL runs random walk with restart on the graph and ranks by score on node after convergence.

5.2 CSEAL

Coupled Set Expander for Any Language is also a text mining component which is called SEAL as sub-routine, and benefited by original coupled constraints developed by NELL. In this section, we apply the four coupled constraints, two types of mutual exclusion example to SEAL and modify SEAL to enable extraction of relation flexibly.

5.2.1 Relation Extraction

SEAL only extracts category instances. We enable SEAL to handle relation instances extraction by adding an infix between two wild-cards to form a single string as input seed. The input seed is as same as category instances, but we do not know which infix is proper until we have target pages. The infix can be any string which length between 1 and 3 characters.

For example, if we want to test relation instance (“臺灣”, “台北市”), and then expression of this instance by regular expression is /臺灣.{1,3} 台北市/.[3] CSEAL processes this relation instance as usual as category instance processing. To build radix trees by the prefix and the suffix string of regular expression matching and then find more relation instances by wrappers which SEAL induced. The learned relation instance will be a concatenation literal string by two category instances and a length three arbitrary string between them. For decomposing learned relation instance string to 2-tuple relation instance, CSEAL recorded infixes when the regular expression of instance was matched, and try to decompose by these infixes.

[3]There is a arbitrary text which length is 1 to 3 Chinese characters between “臺灣” and “台北市”.

5.2.2 More Constraints

Following additional two restrictions are proposed by NELL. In our version of CSEAL, we also implement these features.

Mutual Exclusion

The mutual exclusion property between two predicates is a concrete example of output constraint. There is an another example for mutual exclusion property.

All instances which are extracted from the same page have to belong to the same predicate. If a page contains the instances of two predicates, then ignore the page. The reason is it is easy to extract instances incorrectly when two predicates in the same page.

For example, there is a page which contains a list of tourist attractions in Taiwan. CSEAL might extracts museums, night markets, national parks, temples from this page because they are in the same table, but we only want night markets category instances.

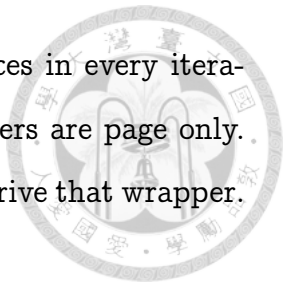
The Same Domain Restriction

CSEAL suffers from template string. A web site usually has many web page, and pages may contain the same template string. The template string will accumulate wrong instances, and then skew the result. For avoiding template string, CSEAL check URL[4] of the web pages, the web pages form the same domain are used only once.

5.2.3 Ranking Candidates

The ranking mechanism of CSEAL is the same as way in CCPL. The difference is that to consider wrappers in CSEAL as patterns in CCPL, and the remains are the same. The reason is that patterns and wrappers represent the same meaning for candidates, they indicate that how many promoter support this candidate.

[4]Uniform Resource Locator.



Another difference is CSEAL only output candidate instances in every iteration. The wrappers are not kept for future use, because wrappers are page only. CSEAL will not apply the wrapper to the pages which did not derive that wrapper.

5.2.4 Querying Search Engine

CSEAL queries Google search engine for related web pages. In our experiment we test CSEAL with two types of inputs, one is querying search engine, and another is using web pages which we crawled Chinese wikipedia[5].

There are two problems of using Chinese wikipedia as an input corpus of CSEAL. The first is that template string skewed the learning result, because the keywords in wikipedia around the same HTML tags. The errors are easy been accumulated between different web pages. The second is that running time becomes a very time consuming work. The most of time CSEAL are testing web pages which do not contain relevant keywords. After experiment, the Chinese wikipedia is not a proper input corpus, so we remove the result generated by considering Chinese wikipedia as input.

Every CSEAL experiments below are accepting web pages which query the Google search engine with related keywords.

5.3 CSEAL Algorithm

In this section, we show Coupled Set Expander for Any Language algorithm in Algorithm 2.

5.4 ChNELL Algorithm

The coupled constraints had been proven to be an effective approach to improve accuracy. In this section, we show that how CCPL was coupled with CSEAL by the

[5]The details of crawling Chinese wikipedia is shown in Section 5.5.2.

Algorithm 2 Coupled Set Expander for Any Language Algorithm

Input: An ontology $\hat{O}$ and set of web pages $\hat{W}$.
Output: Expanded ontology.
for iteration $i = 1, 2, 3, ..., \infty$ **do**
  for predicate $p \in \hat{O}$ **do**
    for relevant web page $w \in \hat{W}$ **do**
      Extract new candidate instances from w by promoted seeds in p ;
    end for
  end for
  Filter out candidates that violate coupling constraints ;
  Rank instance candidates ;
  Promote top candidate instances as new ontology $\hat{O}_{new}$;
  $\hat{O} = \hat{O} \cup \hat{O}_{new}$
end for

multi-view-agreement constraint, and experiment which use Chinese Wikipedia as CCPL inputs.

The algorithm is Chinese Never Ending Language Learner, we called this ChNELL for short.

Algorithm 3 ChNELL Algorithm

Input: An ontology $\hat{O}$ and set of extractors $\hat{E}$.(In our case, number of extractors is 2.)
Output: Expanded ontology.
for iteration $i = 1, 2, 3, ..., \infty$ **do**
  for extractor $e \in \hat{E}$ **do**
    candidates set cs = e.extract($\hat{O}$) ;
  end for
  Promote candidates as new ontology $\hat{O}_{new}$ by coupling rule ;
  $\hat{O} = \hat{O} \cup \hat{O}_{new}$
end for

We show ChNELL algorithm in Algorithm 3. The ChNELL algorithm is called CCPL and CSEAL as sub-routines, and CCPL and CSEAL do not promote instance to knowledge base on their own. ChNELL is responsible for promoting instances to knowledge base. The promoting rule is simple as following. The first case, instances which had been extracted by both components are promoted. The second, instances are only extracted by one component but confidence score > 0.9 are also promoted. The otherwise are rejected. We show these rule in following Table 5.2.

	confidence score > 0.9	confidence score ≤ 0.9
Extracted by one component	Promoted	Rejected
Extracted by both components	Promoted	Promoted

Table 5.2: ChNELL rules used for selection of confident candidates.

Attribute Name	Description
Name	Name of predicate.
Arity	Is predicate a category or a relation.
Populate	Is predicate expandable or not, it is a boolean value.[6]
Mutex Exceptions	To indicate which predicates have mutual exclusive properties.
Instances	Example seed instances with corresponding confidences.[7]
Patterns	Example seed patterns with corresponding confidences.

Table 5.3: Seed common attributes for both categories and relations.

5.5 Experimental Evaluation

In this section we introduce format of input ontology and predicates with attributes, preprocessing steps of corpus, and result after 10 iterations.

5.5.1 Ontology

The ontology consists of a series of predicates and the corresponding attributes. The attributes for both category and relation are listed in Table 5.3, and attributes only for relation predicates are listed in Table 5.4.

The predicates to learned are mainly composed of generally acknowledgment such as country, city, singer, movie and color. In this section, we run ChNELL only with 16 predicates, and show part of example seeds in Table 5.5.

5.5.2 Corpus

In this section, the input corpus is from Chinese wikipedia, which the crawler crawled web pages in August 2012 and finally obtains 889,737 web pages.

[6]We design some categories as non-expandable predicates, such as "大陸". There are only seven continents on the earth, so it is non-expandable. If ontology is without continent category, then continent instances may drift to another location categories and lower overall accuracy.

[7]The confidence is a real number between zero and one. Our handful examples is the most trusted seeds, so default value is one.

[8]Actually variables n, m can be any English alphabets.

Attribute Name	Description
Domain	Category name of domain.
Range	Category name of range.
MappingType	Mapping relation between domain and range, it can be 1-1, 1-n, n-1, n-m.[8]
Irreflexive	Is predicate a irreflexive binary relation, it is a boolean value.
Symmetric	Is predicate a symmetric binary relation, it is a boolean value.
Transitive	Is predicate a transitive binary relation, it is a boolean value.
Inverse	Category name of inverse relation.
MaintainInverse	Does predicate have corresponding inverse predicate and maintain this property.

Table 5.4: Seed attributes only for relations.

Predicates	Seed Instances
國家	中華民國, 臺灣, 台灣, 日本, 美國, 德國, 馬來西亞.
歌手	方大同, 王力宏, 韋禮安, 張惠妹, 張懸, 蕭敬騰, 周華健, 張學友, 蔡依林.
大學	台灣大學, 政治大學, 真理大學, 東吳大學, 淡江大學, 麻省理工, 卡內基美隆大學.
顏色	黑色, 白色, 紅色, 綠色, 藍色.
國家擁有城市	(中華民國, 台北), (中華民國, 臺北), (台灣, 台北), (台灣, 臺北), (臺灣, 台北), (臺灣, 臺北), (日本, 東京), (日本, 橫濱), (日本, 大阪), (美國, 紐約), (美國, 休士頓), (美國, 芝加哥), (美國, 洛杉磯).
歌手的專輯	(林俊傑, 樂行者), (林俊傑, 第二天堂), (林俊傑, 西界), (林俊傑, 100 天), (方大同, 愛愛愛), (方大同, 未來), (方大同, 橙月), (蔡依林, 看我 72 變), (蔡依林, 舞孃), (蔡依林, 王力宏), (王力宏, 公轉自轉), (王力宏, 唯一), (王力宏, 心中的日月), (王力宏, 蓋世英雄), (張惠妹, 姊妹), (張惠妹, 妹力四射), (張惠妹, 我要快樂), (張惠妹, 阿密特).

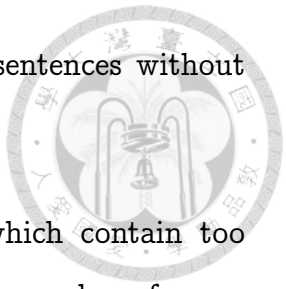
Table 5.5: Seed instances used in CCPL.

We extract page contents by deleting the HTML tags, the JavaScript code blocks and the CSS blocks. The sentences were segmented and tagged with part-of-speech by using Stanford natural language processing tool [21]. To keep the word segmentation and part-of-speech tagging accurate, we transfer punctuations in sentences from half width form to full width form and make sure every sentences are Simplified Chinese by OpenCC. [24] After segmentation and tagging, sentences are transferred back to Traditional Chinese, and then enter the the filtering step.

The criterions of filtering are listed as following.

- **Sum of tokens:** The sentences which with sum of tokens is less than 5 tokens are removed, because they are too short sentences.

- **Without verbs:** The sentences which without any verbs are removed.



Based on our original idea, CCPL learns nothing from sentences without verbs.

- **Tokens-Punctuation Marks ratio:** The sentences which contain too many punctuations are removed. We define a ratio which is number of punctuations in the sentence divided by sum of all tokens in the sentence. The sentences had been filtered out when ratio > 0.5. In preprocessing step, we simply removed tags in web pages. The left sentences may contain template strings such as "原始 _JJ 語言 _NN ：_PU Cassell_NR ；_PU 大陸 _NN ：_PU 卡塞爾 _NR ；_PU 臺灣 _NR ：_PU 卡塞爾 _NR ；_PU 香港 _NR". This filtering rule can effectively remove this kind of sentences.

- **Language detection:** Many web pages are hybrid languages. After segmenting web pages to sentences, some sentences are not Chinese and should be removed. Compact Language Detector (CLD)[10] is a language detector provided by Google. CLD can help us identify language of sentences and filter out the non-Chinese sentences.

- **Tokens-Words Ratio:** The classical Chinese[9] is also called Literary Chinese, "文言文". It is a concise written form in Chinese used by ancient china people. The classical Chinese usually use only one Chinese word to represented a meaning or a concept instead of a token. We find an interesting phenomenon for classical Chinese which have been segmented and tagged by Stanford NLP tools. When tokens-words ratio of the sentence is close to 1, the sentence has high probability is a classical Chinese sentence. When tokens-words ratio is higher, there are many tokens only contain one word in the sentence after segmentation.

  We removed the sentences their tokens-words ratio > 0.8, because there are no modern predicates in classical Chinese sentences.

[9] Classical Chinese is hard to defined well by few words, You can have more information in http://en.wikipedia.org/wiki/Classical_Chinese.

左宗棠 _NR 在 _P 深入 _JJ 調查 _NN 及 _CC 詳細 _JJ 計算 _NN 後 _LC ，_PU 估算 _VV 出 _VV 全部 _DT 軍費 _NN 開支 _NN 共 _AD 需 _VV 白銀 _NR 八百萬 _CD 兩 _CD 。_PU
在 _P 遠距 _AD 鐵路 _NN 方面 _NN 阿德雷得 _NR 是 _VC 從 _P 珀斯 _NR 去 _VV 雪梨 _NR 的 _DEC 印度 _NR 太平洋 _NR 鐵路 _NN 的 _DEG 正中站 _NN ，_PU 以及 _CC 赴 _VV 墨爾本 _NR 的 _DEC 跨域 _JJ 火車 _NN 和 _CC 通過 _P 愛麗斯泉 _NR 去 _VV 達爾文 _NR 的 _DEG 天恩號 _NR 列車 _NN 的 _DEG 終點站 _NN
在 _P 亞都 _NR 麗致 _VV 飯店 _NN 總裁 _NN 嚴長壽 _NR 的 _DEG 協助 _NN 規劃 _NN 下 _LC ，_PU 於 _P 民國 _NR 八十 _OD 年 _M （ _PU 1991_NT 年 _NN ）_PU 增設 _VV 餐飲科 _NN 。_PU

Table 5.6: Randomly selected examples from Corpus generated from Chinese wikipedia.

To deal with template string which appear in many web pages, template string such as slogan in every web pages of the same company, we de-duplicated sentences finally. After all of preprocessing steps, we obtain 7,591,910 sentences, and it is roughly 1.4 GB text file.

5.5.3 Configuration

In beginning of iteration, if there are no enough instances(15) and patterns(5) promoted by last iteration, CCPL randomly selects instances and patterns from knowledge base until enough. After running, CCPL selects the top 30 candidate instances as trusted instances, and 5 candidate patterns as trusted pattern for ChNELL.

CSEAL does not consider promoted instances from last iteration are more important than instances in knowledge base. For every predicates in every iteration, CSEAL samples 5 times from knowledge base. For every sampling, CSEAL randomly selects 5 instances for category and 3 instances for relation to concatenate to an query string. The order of instances are random permutation.

Both of CCPL and CSEAL have a restriction on the length of candidates. To ensure accepted candidate instances/patterns which are in a reasonable range. We show this constraints in Table 5.7 (inclusive), empty means no restrictions.

The mutual exclusion parameter τ is 10.0. All configurations for CCPL and

	At Least	At Most
Category Instance in CCPL	2	
Category Pattern in CCPL		
Relation Instance in CCPL	2	
Relation Pattern in CCPL	3	20
Category Instance in CSEAL	2	50
Relation Instance in CSEAL	2	50

Table 5.7: Restriction on length of instances and patterns in sub-components.

lowerBound4InstancesSeeds	15
lowerBound4PatternsSeeds	5
mutualExclusionConstrain*	10
upperBound4CategoryInstances	30
upperBound4CategoryPatterns	5
upperBound4RelationInstances	30
upperBound4RelationPatterns	5
atLeastCategoryInstanceLength	2
atLeastRelationInstanceLength	2
atLeastRelationPatternLength	3
atMostRelationPatternLength	20

Table 5.8: Attributes and values of Configuration in CCPL. The mutualExclusionConstrain with a star * is τ we mentioned in section 4.3.1, it is used for controlling how soft the filtering is, we reference NELL to set to 10, but without a formal study to find a more proper value.

CSEAL are shown in Table 5.8 and Table 5.9.

5.5.4 Result

The results are promoted by ChNELL after 10 iterations. The input of CCPL is preprocessing corpus from Chinese wikipedia and the input of CSEAL is web pages from instant queries of Google search engine.

We labeled these promoted facts by ourself because size of the result is small. Every promoted instances which are ambiguous had been checked by the Internet. The definitions of countries, cities and movies are clear. The singers have been defined to someone who had released albums, and actors have been defined to someone who had participated any drama.

Table 5.10 are listed estimated precisions and number of promoted facts for

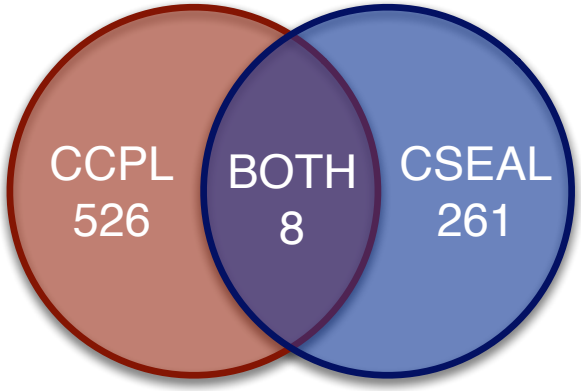


Figure 5.3: Number of promoted facts for subsystem components CCPL and CSEAL, without category "專輯".

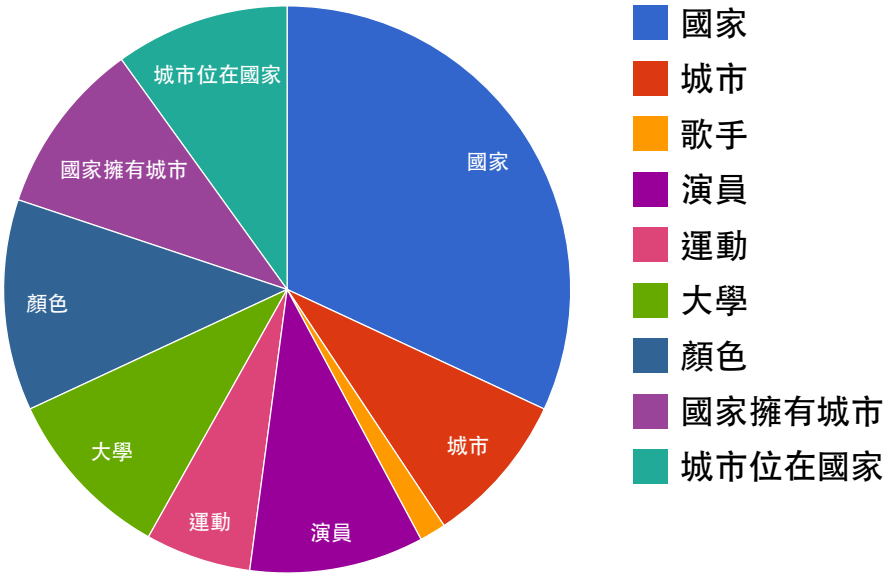


Figure 5.4: Distribution of predicates of correct promoted facts in CCPL.

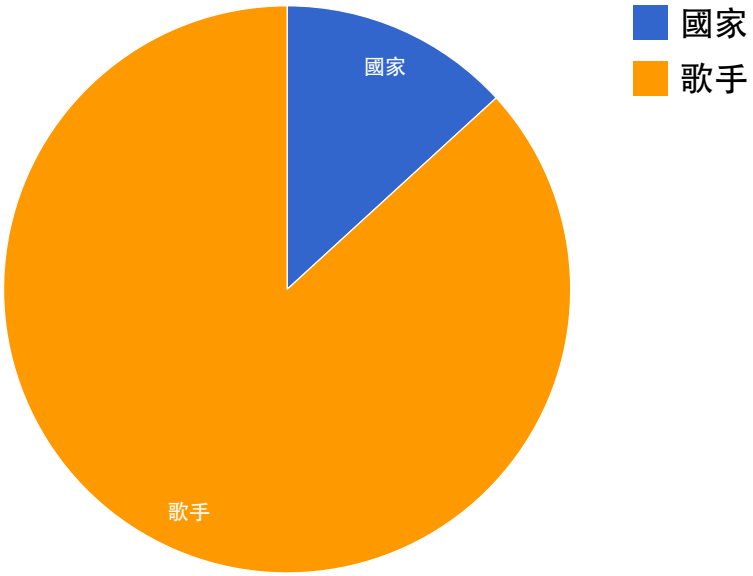


Figure 5.5: Distribution of predicates of correct promoted facts in CSEAL.

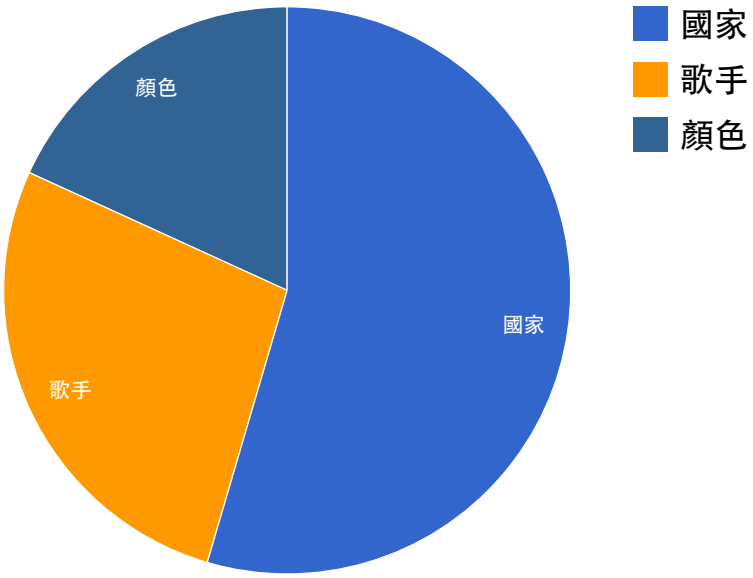


Figure 5.6: Distribution of predicates of correct promoted facts in both components.

lowerBound4CategoryInstances	5
lowerBound4RelationInstances	3
lowerBound4PreInfixOfWrapper	2
mutualExclusionConstrain*	10
upperBound4CategoryInstances	30
upperBound4RelationInstances	30
atLeastCategoryInstanceLength	2
atMostCategoryInstanceLength	50
atLeastRelationInstanceLength	2
atMostRelationInstanceLength	50
samplingTimes	5

Table 5.9: Attributes and values of Configuration in CSEAL.

each predicate after 10 iterations.

The average row summarize promoted instances which cross for all predicates. The predicate "專輯" is drifting to general unrelated phrase since second iteration. We have 202 promoted instances of album but none of them are correct. For the failure in albums, we add one more row for average without "專輯" predicate.

Figure 5.3 gives number of promoted facts for each subsystem components without "專輯" predicate.

Figure 5.4, 5.5 and 5.6 show distribution of predicates of correct promoted facts in each subsystem components.

5.5.5 Discussion

Estimates of Precision

The predicates which overall accuracy belows 60% are "專輯", "運動", "大學", "國家擁有城市" and "城市位在國家". The two corresponding inverse relations can be considered as the same relation because they are inverse relations for each other. We turn inverse propagation on by default. The instances in the corresponding inverse relation shall be the same, because they propagate to each other.

- **"專輯" (Album):** All potential instances which CCPL captured are a series of noun tokens, but some predicate instances usually do not consisted of

Predicate Name		Precision(%)	Number of Promoted(#)
國家	Country	63.6	228
城市	City	96.6	30
歌手	Singer	98.2	229
專輯	Album	0	202
演員	Actor	86.8	38
電影	Movie	-	0
運動	Sport	35.0	57
大學	University	57.8	57
高中	High school	-	0
顏色	Color	93.3	45
國家擁有城市	CountryContainsCity	57.8	57
城市位在國家	CityLocatedInCountry	57.8	57
歌手的專輯	Singer'sAlbum	-	0
專輯屬於歌手	AlbumOfSinger	-	0
電影中的演員	MovieStarActor	-	0
演員去演電影	ActorStarredInMovie	-	0
Average		59.3	1000
Average without "專輯"		74.3	798

Table 5.10: Precisions and number of promoted facts for each predicates.

noun tokens. It might be a sentence, or a clause. For example, Chinese Albums such as "陶喆: 再見你好嗎", "林俊傑: 因你 而在", "盧廣仲: 有吉他的流行音樂". There are not all noun tokens in these named entities after segmentation.

- "運動" (Sport): The sport predicate are mixed by some hobbies, such as sport events and sport team name. The reason is that in our 16 predicates, there are no proper mutual exclusive predicates to prevent drifting.

- "大學" (University): The university predicate extend to some types of school, such as elementary schools, junior high schools and name of degrees. This problem can also be fixed by adding more mutual exclusive predicates.

- "國家擁有城市" (CountryContainsCity): We have approximately 60% accuracy for relations. The errors which have been made for instances learned in relations are generated by wrong segmentation in CCPL. The assumption for a series of noun tokens can be combined to a noun phrase is not appropriate for some case. ("日本", "沖繩縣鐵路車站") is an example for combining

too much tokens.

Number of Promotions and Distribution

The promotions from both components are rare in our results, because the non-confidence promotions will not be kept until next iteration. Maybe it needs some lucky for promotion which is promoted by both components in the same iteration.

We find out that CCPL has more promotions than CSEAL. Based on Figure 5.4 and Figure 5.5, we know the promotions from CSEAL focus on few predicates such as "國家" and "歌手", and the promotions from CCPL have more diversity by comparison. Our conjecture is that critical causes are amount number of web pages for CSEAL and sampling times of seeds. In CCPL, every sentence contains patterns can be a potential promoter, but in CSEAL, number of documents which can induce wrappers are potential promoters. The number of potential promoters of CSEAL is much less than CCPL.

After 10 iterations, the average downloaded web pages from search engine is 3000 for each iteration, so there is 187.5 web pages for each predicate in average. It is not easy for extracting instances in such few documents.

We can fix this up by two solutions, the first is increasing amount of downloaded web pages by sampling keywords more times, and the second is lowering restriction of wrapper induction by querying Google in fewer instances.

Chapter 6

Scalability

In Chapter 5, we show result of coupling learning of CCPL and CSEAL, and discuss the causes of error. NELL have pointed out that more constraints make the results more accurate, there is a easy way to increase coupled constraints by learning more independent predicates concurrently. In this chapter, we try to test scalability of our work, by introducing a new input source - The ClueWeb09 Dataset as corpus for CCPL, and parallelize our work by famous distributed computing framework - Apache Hadoop.

6.1 Introduction to Clueweb

> The ClueWeb09 dataset was created to support research on information retrieval and related human language technologies. It consists of about 1 billion web pages in ten languages that were collected in January and February 2009. The dataset is used by several tracks of the TREC conference. by ClueWeb09 official web site. [19]

There is 177,489,357 Chinese pages in ClueWeb 09 dataset, most of these pages are Simplified Chinese, but few of them are Traditional Chinese, Korean or Thai. Even in Simplified Chinese web pages, there are encoded by character sets like GB2312, GB18030, GBK. In Traditional Chinese web pages, there are encoded by Big-5 or utf-8, so, at first, we use International Components for Unicode (ICU4C [18]) to detect file encoding automatically, then convert all pages to utf-8 encoding.

We detect language type of pages and filter out non-Chinese pages, and make sure all pages are Simplified Chinese by OpenCC before segmentation and POS tagging, rest steps are the same as we mentioned in Section 5.5.2, segmentation and POS tagging are time consuming job, it took about 80 days running with 25 personal computers, after all steps, we transfer sentences back to Traditional Chinese.

After segmentation, POS tagging and filtering, we got about 1 TB (terabyte) sentences, we encountered a problem in deduplicating step, there is no computer can contain all 1 TB data in memory to deduplicate then avoiding the same sentence, we divide deduplicating job to different computer according to first 5 characters of sentence as dispatching key, then deduplicate locally.

Finally, we got a Traditional Chinese corpus, which contains 1,794,760,146 sentences, it is about 408 GB (gigabyte). [1]

[1] It is an interesting coincidence, we extract 1.7 billion sentences from 0.17 billion web pages.

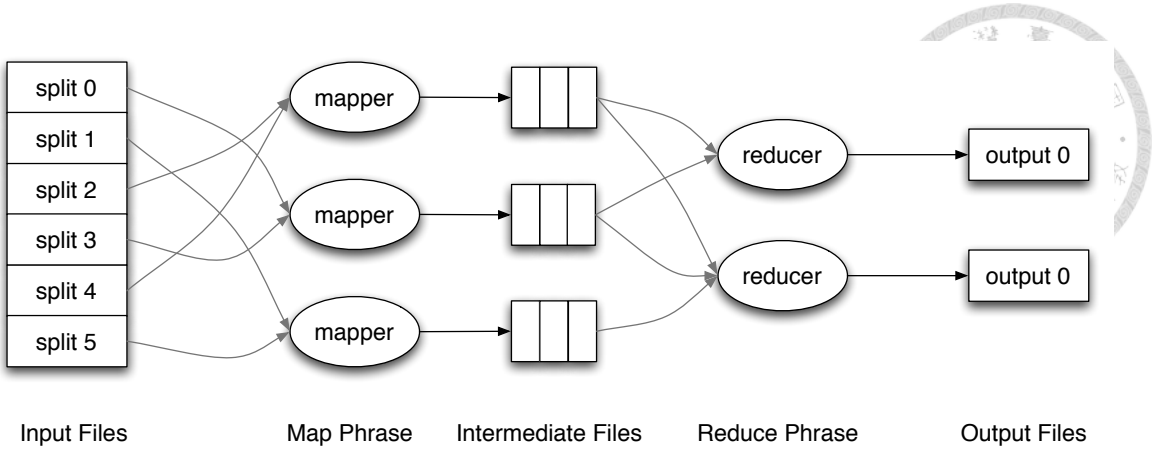


Figure 6.1: MapReduce work flow with 2 reducers.

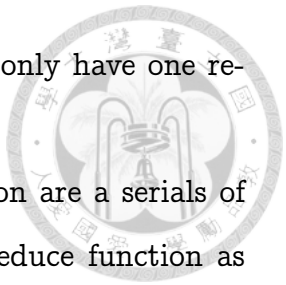
6.2 Parallelization

The most time consuming step in CCPL is concepts extraction, because time of running concepts extraction is linear to input corpus, and it is easy to parallelication. Base on experiment, running CCPL with about 200 predicates will approximately produce 10 GB result, so after extractions, the bottleneck is concepts selection, we will brief introduce MapReduce model in next subsection and transfer CCPL to MapReduce model, let CCPL fully parallelization even on concepts selection step.

6.2.1 MapReduce

MapReduce is a programming model for large data sets processing proposed by [5]. The model is inspired by two famous functions "map" and "reduce" in functional programming. In pure functional programming, the functions are "referential transparent", or called "without side effect", it means that function result depends only on the values of its parameters. With this property, function can easy be parallelly computing, "map" is a good example, mapper is instance of running map function, and reducer is instance of running reduce function, after mappers processing input data, then we aggregates all outputs from mappers to as input of reducers. We show this flow in Figure 6.1.

The number of reducers depends on natural of the problem, some problems



have strong correlation on result of mappers, then they might only have one reducer.

The inputs and outputs of map function and reduce function are a serials of (key, value) pair, we list the function signature of map and reduce function as following.

- $map(k_1, v_1) \rightarrow list(k_2, v_2)$
- $reduce(k_2, list(v_2)) \rightarrow list(v_2)$

It is important for determining how to pass output of mappers to reducers. By default, MapReduce framework will guarantee that the pairs of the same k_2 will pass to the same reducer for processing. We will exploit this to decompose concepts selection to 2 phrases, then fully parallelize CCPL.

6.2.2 Multi-Level MapReduce

If typical $map \rightarrow reduce$ are 2 levels procedure, then we divide CCPL to 3 levels procedure, just like $map \rightarrow map \rightarrow reduce$.

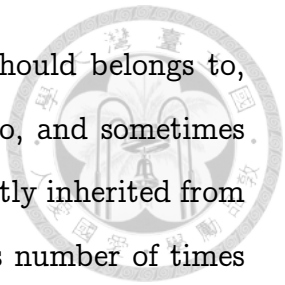
First level mappers do the original concepts extraction, they read line by line from input files and output 5-tuple, the output data looks like

(“_ 過境簽證”， “Category”， “Instance”， “國家”， “美國”)

The first field is the extracted candidate, and is also the “key” in MapReduce, second and third fields indicate it belongs to category or relation, extracted from instance or pattern, fourth field is predicate name and fifth field is promoter.

Second level mappers is responsible for filtering. These 5-tuples which have the same candidate will be processed in the same mapper, so CCPL can check mutual exclusion property and helping function property are valid or not, and type-checking for relation instances. There is no limit on number of second level mappers, because we consider candidate as dispatching key. The output of second level mappers are 7-tuple, the output data looks like

(“國家”， “Instance”， “_ 再出資援助銀行業”， “Category”， “國家”， “英國”，14)



The first field is the category name which this candidate should belongs to, category name in fifth field is the category promoter belongs to, and sometimes they might be different. Second, third and fourth fields are directly inherited from fields 1-3 in 5-tuple, sixth filed is promoter and seventh field is number of times this promoter co-occur with the candidate.

The "key" between level 2 mappers and reducers is composite of first and second fields in 7-tuple, which mean all instance/pattern candidates in the same predicate will be passed to the same reducer. In reducers, CCPL only need to rank candidates. The setting of key make number of reducers can be as more as $2 \times |predicates|$, because instances/patterns of predicates can sort individually.

6.3 Experimental Evaluation

We introduce extended ontology and corpus in this section, and evaluate result from ChNELL after 5 iterations.

6.3.1 Ontology, Corpus and Configuration

We extend our ontology to contains 206 predicates, there are 145 categories and 61 relations. Some predicates are non-propagable, for example, "大陸" and "大洋". And there are some predicates which we regard as worthless to learned, for example, "日期", "星期", "月份" and "年份", but they exist for providing negative examples, to enhance mutual exclusive constraints.

The categories mainly consist of different types of location and different types of celebrity. Some categories are chosen for testing perceptivity of ChNELL. The relations is easy for selecting. We choose relations from combinations of any two categories in the 145 categories if the combination is obviously meaningful. The full lists of categories and relations are showed in Appendix A.

The corpus consists of preprocessing sentences from clueweb we mentioned in 6.1. And all configurations and parameters are the same as experiments in Section

Figure 6.2: The evaluation website interface.

5.5.3.

6.3.2 Result

We evaluate results from ChNELL after 5 iterations. ChNELL generates 5777 unique instances, and we builds a website for evaluating extracted instances by human. The website is shown in Figure 6.2.

Table 6.1 and 6.2 are listed estimated precisions and number of promoted facts for categories and relations individually from ChNELL after 5 iterations.

The categories accuracy table only lists categories which extracted over 50 instances in first 5 iterations, and the full table which lists all categories is shown in Appendix B.

Table 6.3 shows accuracy of extracted instances in different view. We estimates accuracy by predicate types such as accuracy of categories and relations, or estimates accuracy by different subsystem components, and finally we estimates overall accuracy in extracted instances.

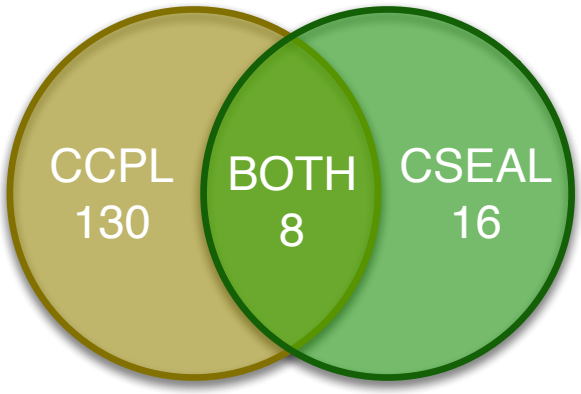


Figure 6.3: Distribution of predicates for subsystem components CCPL and CSEAL.

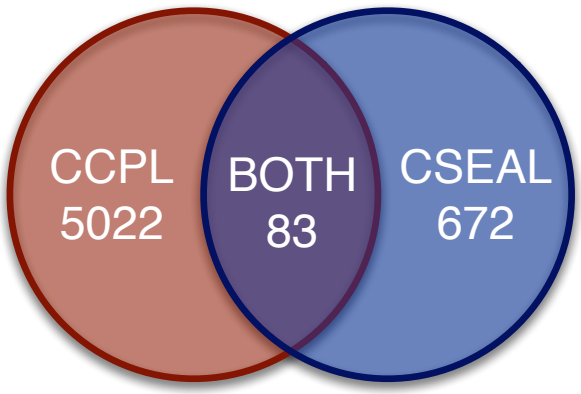


Figure 6.4: Number of promoted facts for subsystem components CCPL and CSEAL.

Figure 6.3 shows distribution of predicates which extracted instance belong to. Figure 6.4 gives number of promoted facts for each subsystem components.

6.3.3 Discussion

As the Table 6.3 represents, the average accuracy of categories (79.8%) is much better than relations (54.3%). One of reasons might be that the relation patterns may not be significant enough for relation extraction. For example, "A is teammates of B" is a sentence which interleaving strings have strong semantic meaning to represent a relation. If in Chinese, the sentence might be "A 和 B 是隊友" or "A 是 B 的隊友". There is no interleaving strings exist for representing such relation, so based on Chinese grammar, interleaving strings are not enough for relation extraction.

As results from Section 5.5.4, the CCPL makes more mistakes than CSEAL because CSEAL employs stronger constraints or fetches too few web pages.

The diversity of predicates which extracted instances also supports this guess. CSEAL learns instances from too few types of predicates, and quantity of instances is less than CCPL. In next step, first thing is to release the constraints in CSEAL such as using fewer keywords to query, sampling more times from knowledge base and fetching more pages.

The overall accuracy is improved by comparing with results from Section 5.5.4 because extraction benefited by concurrently learning. Few predicates still drift to irrelevant knowledge after only 5 iterations such as "醫院", "鳥" and "魚". We plan to correct classification of predicates by scattering present predicates to smaller or combining predicates. Adding more mutual exclusive predicates to knowledge base and adding more counter examples in non-propagate predicates might be helpful.

Predicate Name		Precision(%)	Number of Promoted(#)
中國皇帝	Chinese monarch	91.5	71
中藥	Chinese medicine	95.8	72
公園	Park	47.3	76
國家	Country	95.6	159
天氣現象	Weather	90.9	55
寺廟	Temple	75.3	81
導演	Director	67.3	52
島嶼	Island	41.1	68
情緒	Emotion	95.7	94
捷運站	MRT station	96.4	111
政府機構	Government organization	80.6	93
政治職位	Political position	87.3	71
方位	Direction	67.1	67
服裝	Clothing	89.6	87
歌手	Singer	100	123
單位	Unit	77.3	106
疾病	Disease	93.2	74
程式語言	Programming language	85.2	61
節慶	Festival	73.1	67
罪行	Crime	86.3	117
花草	Flower	96.6	59
行政區	Administrative division	96.9	131
街道	Street	93.1	88
詩人	Bard	98.7	77
調味品	Condiment	88.7	62
貨幣	Currency	49.5	105
資訊工程領域	CS area	55.4	74
身體部位	Body part	92.5	54
醫院	Hospital	37.5	96
銀行	Bank	75	68
顏色	Color	95.7	95
鳥	Bird	66.6	54
Average		79.9	4226

Table 6.1: Precisions and number of promoted facts for categories which are extracted over 50 instances in first five iterations.

Predicate Name		Precision(%)	Number of Promoted(#)
國家使用的語言	CountryUsedLanguage	17.3	138
國家使用的貨幣	CountryUsedCurrency	10.8	157
國家擁有博物館	CountryContainsMuseum	13.2	53
國家有的機場	CountryContainsAirPort	0	1
國家有的縣市	CountryContainsCity	65.3	52
國家統治的元首	CountryRuledByPresident	48.8	129
大陸上的國家	ContinentContainsCountry	46.6	135
導演導的電影	DirectorOfMovie	91.7	109
政黨的政治家	PoliticalPartyParticipatedPolitician	87.5	8
朝代之後的朝代	DynastyAfterDynasty	91.8	282
歌手出品的專輯	SingerPerformsAlbum	0	1
疾病治療的藥物	DiseaseCuredByMedicine	33.3	153
節慶的日期	FestivalOccursDate	86.9	23
縣市有的夜市	CityContainsNightMarket	50.0	6
縣市有的街道	CityContainsStreet	44.7	76
臺灣偶像劇的演員	TaiwanDramaActedByActor	0	4
詩人寫的詩詞	PoetWritesPoetry	96.6	60
運動的運動位置	SportContainsPosition	13.2	53
電影參與的演員	MovieActedByActor	80	30
首席執行官的公司	CEOOfCompany	87.6	81
Average		54.3	1551

Table 6.2: Precisions and number of promoted facts for relations in first five iterations.

	#Extracted Instances	#Correct Instances	Accuracy
Category	4,226	3,375	79.8
Relation	1,551	843	54.3
CCPL	5,022	3,499	69.6
CSEAL	672	642	95.5
Both	83	77	92.7
Total	5,777	4,218	73

Table 6.3: Accuracy table for different view of results. The first view is to evaluate results from different predicate types(category and relation), and the second view is to evaluate results from different subsystem components(CCPL, CSEAL and Both. Finally, we give a overall accuracy for extracted instances from ChNELL.

Chapter 7

Conclusion and Future Work

In this chapter, we summarize that we have done for now and the contributions of this thesis, and we also propose future work to improve our knowledge base.

7.1 Conclusion

This system proposed a prototype for automatically extracting commonsense knowledge in Chinese from the Internet. We design Coupled Chinese Pattern Learner, which extracts Chinese commonsense knowledge by patterns which frequently occur in Chinese sentence. The patterns are generated by discovering noun phrases usually appear together with verb nouns.

After that, we implement Coupled Set Expander for Any Language. CCPL and CSEAL collaborate with each other to to improve the diversity of knowledge and the overall precision of ChNELL.

Our experiments show that ChNELL is a scalable system. ChNELL has the scalability for concurrently learning 206 predicates, the predicates include 145 categories and 61 relations.

7.2 Future Work

The system can be improved by following directions, the directions are also interesting research topics in academic.

1. Based on the results in Chapter 6, large portion of mistakes which had been made by ChNELL is capturing the wrong range of a noun phrase as named entity. Introducing sophisticated named entity recognition (NRE) in Chinese can effectively improve this problem.

2. This idea is inspired by Coupled Morphological Classifier (CMC). Although part of traditional morphological features such as root words and affixes do not fit Chinese and are language-dependent, Chinese still has its own morphological feature like "部首" (radical), "偏旁" and "六書". It is exciting research topic for building morphological classifier based on these traditional Chinese morphological features.

3. In CCPL, we emphasize the relationship between noun phrases and verb

phrases, because the noun phrases belong to the same category usually share common verb phrases. We are curious about other relationships are implicit in Chinese, even more, in any languages.

Bibliography

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory (COLT 1998)*. ACM, 1998.

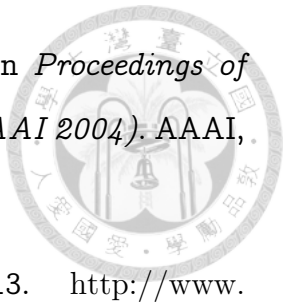
[2] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*. AAAI, 2010.

[3] A. Carlson, J. Betteridge, R. C. Wang, E. R. H. Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*. WSDM, 2010.

[4] J. R. Curran, T. Murphy, and B. Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACL 2007)*, 2007.

[5] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6 (OSDI 2004)*. USENIX Association, 2004.

[6] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Methods for domain-independent information



extraction from the web: An experimental comparison. In *Proceedings of the 19th national conference on Artifical intelligence (AAAI 2004)*. AAAI, 2004.

[7] W. Foundation. Wikipedia, the free encyclopedia, 2013. http://www.wikipedia.org/.

[8] W. Foundation. Wiktionary, the free dictionary, 2013. http://www.wiktionary.org.

[9] M. M. Group. Internet world stats, 2012. http://www.internetworldstats.com/stats.htm.

[10] G. Inc. Compact language detector library (cld), 2013. http://code.google.com/p/chromium-compact-language-detector/.

[11] D. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38:33–38, 1995.

[12] H. Liu and P. Singh. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22:211–226, 2004.

[13] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[14] H. Nakamura. Radix tree naive implementation of radix tree for ruby, 2013. https://github.com/nahi/radixtree.

[15] F. U. of Berlin, the University of Leipzig, and O. Software. Dbpedia, 2013. http://dbpedia.org/.

[16] D. of Computer and U. o. P. Information Science. Chinese language processing at penn, 2013. http://www.cis.upenn.edu/~chinese/.

[17] U. of Washington's Turing Center. Reverb: Open information extraction software, 2013. http://reverb.cs.washington.edu/.

[18] T. P. M. C. (PMC). Icu - international components for unicode, 2013. http://site.icu-project.org/.

[19] T. L. Project. The clueweb09 dataset, 2013. http://lemurproject.org/clueweb09/.

[20] M. Technologies. Freebase, 2013. http://www.freebase.com/.

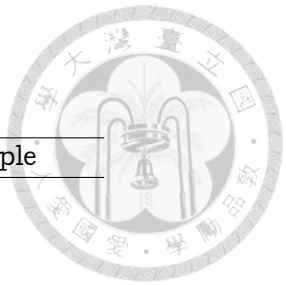
[21] S. University. The stanford natural language processing group, 2013. http://www-nlp.stanford.edu/.

[22] L. von Ahn, M. Kedia, and M. Blum. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI 2006)*. ACM, 2006.

[23] R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In *Proceedings of IEEE International Conference on Data Mining (ICDM 2007)*. ICDM, 2007.

[24] 郭家寶 (BYVoid). Open chinese convert (opencc) 開放中文轉換, 2013. http://code.google.com/p/opencc/.

Appendix A: Ontology

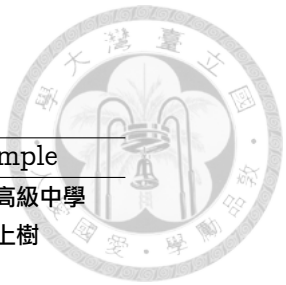


Chinese Name	English Name	Propagable	Example
大陸	continent	X	亞洲
大洋	ocean	X	太平洋
島嶼	island	O	長灘島
湖泊	lake	O	日月潭
山峰	mountain	O	喜瑪拉雅山
海灘	beach	O	邁阿密海灘
海域	sea	O	台灣海峽
海灣	bay	O	墨西哥灣
河流	river	O	黃河
沙漠	desert	O	戈壁沙漠
方位	direction	O	東方
博物館	museum	O	羅浮宮
商場	shopping mall	O	家樂福
摩天大樓	skyscraper	O	哈里發塔
街道	street	O	羅斯福路
農場	farm	O	武陵農場
火車站	train station	O	紐約中央車站
動物園	zoo	O	木柵動物園
海港	port	O	基隆港
餐廳	restaurant	O	鼎泰豐
公園	park	O	陽明山國家公園
機場	airport	O	松山機場
橋樑	bridge	O	永福橋
水庫	reservoir	O	石門水庫
水族館	aquarium	O	香港海洋公園
銀行	bank	O	臺灣銀行
隧道	tunnel	O	七堵隧道
鐵路	railroad	O	京滬鐵路
公路	highway	O	福建公路
醫院	hospital	O	臺大醫院
旅館	hotel	O	圓山大飯店
遊樂園	themepark	O	六福村
早餐店	breakfaststore	O	美而美
咖啡廳	coffee shop	O	星巴克
飲料店	beverage store	O	五十嵐
書店	book store	O	誠品書店
百貨公司	department store	O	新光三越
電影院	movie theater	O	信義威秀
速食店	fastfood store	O	麥當勞
寺廟	temple	O	鎮瀾宮

Table 1: Categories(1 - 40) in the ontology.(continued on next page)

Chinese Name	English Name	Propagable	Example
教堂	church	O	聖索菲亞大教堂
市場	market	O	永安市場
古蹟	historic monuments	O	安平古堡
音樂廳	music hall	O	國家音樂廳
棒球場	baseball field	O	天母棒球場
捷運站	MRT station	O	公館站
圖書館	library	O	台北市立圖書館
夜市	night market	O	羅東夜市
加油站	gas station	O	全國加油站
州或省	state orprovince	O	浙江
國家	country	O	中華民國
縣市	county or city	O	台北市
行政區	administrative division	O	大安區
工作職位	job position	O	廚師
演員	actor	O	周星馳
歌手	singer	O	張學友
運動員	athlete	O	彭政閔
首席執行官	ceo	O	郭台銘
廚師	chef	O	阿基師
導演	director	O	李安
罪犯	criminal	O	陳進興
中國皇帝	chinese monarch	O	秦始皇
政治家	politician	O	馬英九
教授	professor	O	許永真
科學家	scientist	O	亞里斯多德
作家	writer	O	金庸
詩人	bard	O	李白
元首	president	O	歐巴馬
哺乳動物	mammal	O	猴子
鳥	bird	O	畫眉鳥
昆蟲	insect	O	蟑螂
魚	fish	O	鮪魚
蔬菜	vegetable	O	菠菜
花草	flower	O	玫瑰花
家具	furniture	O	書桌
文具	office item	O	鉛筆
浴室物品	bathroom item	O	牙刷
廚房物品	kitchen item	O	菜刀
學校	school	O	普通高中
大學	university	O	臺灣大學

Table 2: Categories(41 - 80) in the ontology.(continued on next page)



Chinese Name	English Name	Propagable	Example
高中	senior highschool	O	建國高級中學
中式菜名	chinese food	O	螞蟻上樹
小吃	snack	O	雞排
飲料	beverage	O	珍珠奶茶
調味品	condiment	O	沙茶醬
水果	fruit	O	蘋果
身體部位	body part	O	腹部
肌肉	muscle	O	三角肌
顏色	color	O	紅色
幾何數形狀	geo metricshape	O	梯形
貨幣	currency	O	歐元
情緒	emotion	O	寂寞
疾病	disease	O	心臟病
精神病	mental disorder	O	強迫症
民族	ethnic group	O	阿美族
語言	language	O	閩南語
行星	planet	O	火星
宗教	religion	O	基督教
武器	weapon	O	步槍
天氣現象	weather phenomenon	O	藍天
網站	website	O	百度
罪行	crime	O	偽證罪
科系	department	O	資工系
節慶	festival	O	除夕
朝代	dynasty	O	明朝
中藥	chinese medicine	O	人參
藥物	drug	O	阿斯匹林
電玩	video game	O	刺客教條
電視節目	television show	O	康熙來了
運輸工具	transportation	O	火車
政治職位	political office	O	市長
政黨	political party	O	中國國民黨
報紙	newspaper	O	蘋果日報
服裝	clothing	O	毛衣
戲劇類型	drama type	O	京劇
臺灣偶像劇	taiwan drama	O	我可能不會愛你
美國影集	american drama	O	冰與火之歌
大陸影集	inland drama	O	步步驚心
韓國偶像劇	korea drama	O	我叫金三順
電影	movie	O	神鬼認證

Table 3: Categories(81 - 120) in the ontology.(continued on next page)

Chinese Name	English Name	Propagable	Example
音樂專輯	music album	O	阿密特
音樂流派	music genre	O	爵士
樂器	music instrument	O	鋼琴
詩詞	poetry	O	長恨歌
運動	sport	O	籃球
體育賽事	sports event	O	奧運
球類運動隊伍	sport team	O	休士頓火箭
球類運動位置	sport position	O	捕手
運動場地	stadium	O	台大體育館
資訊工程領域	computer sciencearea	O	人工智慧
程式語言	programming language	O	ruby
機構	institution	O	世界貿易組織
公司	company	O	宏達電
汽車製造商	automobile maker	O	凱迪拉克
航空公司	airline company	O	國泰航空
政府機構	government organization	O	考試院
時間量詞	time	O	今晚
日期	date	O	7 月 31 日
星期	week	O	週日
月份	month	O	十月
年份	year	O	1985 年
世紀	century	O	二十一世紀
戰爭	war	O	中日甲午戰爭
稱謂	appellation	O	父母
物理單位	unit	O	密度

Table 4: Categories(121 - 145) in the ontology.(continued on next page)

Chinese Name	English Name	Domain	Range	Inverse Relation							Example
大陸上的國家	ContinentContainsCountry	大陸	國家	國家位在大陸	1-n	O	O	X	X	O	(北美洲, 加拿大)
國家擁有博物館	CountryContainsMuseum	國家	博物館	博物館位在國家	1-n	O	O	X	X	O	(英國, 大英博物館)
國家有的摩天大樓	CountryContainsSkyscraper	國家	摩天大樓	摩天大樓位在國家	1-n	O	O	X	X	O	(臺灣, 臺北 101)
縣市有的街道	CityContainsRoad	縣市	街道	街道位在縣市	n-m	O	O	X	X	O	(臺北市, 復興南路)
縣市有的動物園	CityContainsZoo	縣市	動物園	動物園位在縣市	1-n	O	O	X	X	O	(新竹縣, 六福村野生動物園)
國家有的機場	CountryContainsAirport	國家	機場	機場位在國家	1-n	O	O	X	X	O	(法國, 夏爾戴高樂機場)
縣市有的捷運站	CityContainsMRTStation	縣市	捷運站	捷運站位在縣市	1-n	O	O	X	X	O	(臺北市, 六張犁站)
縣市有的夜市	CityContainsNightmarket	縣市	夜市	夜市位在縣市	1-n	O	O	X	X	O	(高雄, 花園夜市)
國家有的縣市	CountryContainsCity	國家	縣市	縣市位在國家	n-m	O	O	X	X	O	(美國, 紐約)
首席執行官的公司	CEOOfCompany	首席執行官	公司	公司和他的首席執行官	n-m	O	O	X	X	O	(林百里, 廣達集團)
導演導的電影	DirectorOfMovie	導演	電影	電影被導演執導	1-n	O	O	X	X	O	(陳可辛, 投名狀)
罪犯犯下的罪行	CriminalCommitCrime	罪犯	罪行	罪行被罪犯犯下	n-m	O	O	X	X	O	(陳進興, 擄人勒贖殺人罪)
國家使用的貨幣	ConuntryUseCurrency	國家	貨幣	貨幣在國家流通	n-m	O	O	X	X	O	(泰國, 泰銖)
電影參與的演員	MovieActedByActor	電影	演員	演員所演的電影	n-m	O	O	X	X	O	(賭神, 周潤發)
歌手出品的專輯	SingerPerformAlbum	音樂專輯	歌手	音樂專輯屬於歌手	1-n	O	O	X	X	O	(王力宏, 唯一)
詩人寫的詩詞	PoetWritesPoetry	詩人	詩詞	詩詞的創作者	1-n	O	O	X	X	O	(李白, 將進酒)
運動裡的運動員	SportMajoredAthlete	運動	運動員	運動員做的運動	n-m	O	O	X	X	O	(曾雅妮, 高爾夫球)
國家使用的語言	CountryUsesLanguage	國家	語言	語言被國家使用	n-m	O	O	X	X	O	(臺灣, 國語)
政黨的政治家	PoliticalPartyParticipatedPolitician	政黨	政治家	政治家參與政黨	1-n	O	O	X	X	O	(民主進步黨, 蔡英文)
中國皇帝統治的朝代	MonarchRulesDynasty	中國皇帝	朝代	朝代時期的中國皇帝	n-1	O	O	X	X	O	(唐朝, 唐太宗)
科系相關的教授	DepartmentRelatedProfessor	科系	教授	教授在科系執教	n-m	O	O	X	X	O	(資訊工程學系, 許永真)
為大學工作的教授	UniversityHiresProfessor	大學	教授	教授在大學執教	n-m	O	O	X	X	O	(台灣大學, 陳信希)
國家統治的元首	CountryRuledByPresident	國家	元首	元首統治的國家	n-m	O	O	X	X	O	(中國, 溫家寶)
疾病治療的藥物	DiseaseCuredByMedicine	疾病	藥物	藥物可以治癒疾病	n-m	O	O	X	X	O	(咳嗽, 川貝枇杷膏)
臺灣偶像劇的演員	TaiwanDramaActedByActor	台灣偶像劇	演員	演員出演台灣偶像劇	n-m	O	O	X	X	O	(痞子英雄, 陳意涵)
大陸影集的演員	ChinaSeriesActedByActor	大陸影集	演員	演員出演大陸影集	n-m	O	O	X	X	O	(步步驚心, 吳奇隆)
節慶的日期	FestivalOccurDate	節慶	日期	日期有的節日	1-1	O	O	X	X	O	(教師節, 9 月 28 日)
運動隊伍做的運動	SportTeamPlaysSport	運動隊伍	運動	主要運動對於運動隊伍	1-1	O	O	X	X	O	(紐約洋基, 棒球)
運動的運動位置	SportContainsPosition	運動	運動位置	運動位置在運動中	n-m	O	O	X	X	O	(籃球, 中鋒)
運動員是隊友	IsTeammateOf	運動員	運動員	運動員是隊友	n-m	O	O	O	X	X	(林易增, 陳致遠)
朝代之後的朝代	DynastyAfterDynasty	朝代	朝代	朝代之前的朝代	n-m	O	O	X	O	O	(唐朝, 宋朝)

Table 5: Relations(31) in the ontology. The properties order in a row are mapping type, propagable, irreflexive, symmetric, transitive and propagation of inverse.

Appendix B: Result from ChNELL

Predicate	Name	Precision(%)	Number of Promoted(#)
大陸	Continent	-	0
大洋	Ocean	-	0
島嶼	Island	41.1	68
湖泊	Lake	100	4
山峰	Mountain	83.3	12
海灘	Beach	-	0
海域	Sea	61.1	36
海灣	Bay	40	20
河流	River	85.1	27
沙漠	Desert	21.4	14
方位	Direction	67.1	67
博物館	Museum	100	2
商場	Shopping mall	100	1
摩天大樓	Skyscraper	100	1
街道	Street	93.1	88
農場	Farm	-	0
火車站	Trainstation	-	0
動物園	Zoo	-	0
海港	Port	100	7
餐廳	Restaurant	-	0
公園	Park	47.3	76
機場	Airport	75	8
橋樑	Bridge	-	0
水庫	Reservoir	-	0
水族館	Aquarium	-	0
銀行	Bank	75	68
隧道	Tunnel	100	1
鐵路	Railroad	80	15
公路	Highway	-	0
醫院	Hospital	37.5	96
旅館	Hotel	66.6	6
遊樂園	Theme park	100	4
早餐店	Breakfast store	-	0
咖啡廳	Coffee shop	-	0
飲料店	Beverage store	-	0
書店	Book store	-	0
百貨公司	Department store	100	4
電影院	Movie theater	-	0
速食店	Fast food store	100	8
寺廟	Temple	75.3	81

Table 6: Precisions and number of promoted facts for categories(1 - 40) first five iterations.

Predicate Name		Precision(%)	Number of Promoted(#)
教堂	Church	-	0
市場	Market	-	0
古蹟	Historic monument	80	30
音樂廳	Music hall	-	0
棒球場	Baseball field	80	5
捷運站	MRT station	96.4	111
圖書館	Library	-	0
夜市	Night market	100	5
加油站	Gas station	-	0
州或省	State or province	100	2
國家	Country	95.6	159
縣市	County or city	90	10
行政區	Administrative division	96.9	131
工作職位	Job position	75	20
演員	Actor	76.9	13
歌手	Singer	100	123
運動員	Athlete	28.5	7
首席執行官	Ceo	50	22
廚師	Chef	-	0
導演	Director	67.3	52
罪犯	Criminal	100	19
中國皇帝	Chinese monarch	91.5	71
政治家	Politician	66.6	3
教授	Professor	-	0
科學家	Scientist	80	5
作家	Writer	60	5
詩人	Bard	98.7	77
元首	President	88	25
哺乳動物	Mammal	55.5	9
鳥	Bird	66.6	54
昆蟲	Insect	100	1
魚	Fish	40	20
蔬菜	Vegetable	60.8	46
花草	Flower	96.6	59
家具	Furniture	74.1	31
文具	Office item	69.2	13
浴室物品	Bathroom item	76.9	13
廚房物品	Kitchen item	80.5	36
學校	School	50	30
大學	University	-	0

Table 7: Precisions and number of promoted facts for categories(41 - 80) first five iterations.

Predicate Name		Precision(%)	Number of Promoted(#)
高中	Senior high school	-	0
中式菜名	Chinese food	-	0
小吃	Snack	71.4	21
飲料	Beverage	80.6	31
調味品	Condiment	88.7	62
水果	Fruit	100	11
身體部位	Body part	92.5	54
肌肉	Muscle	80.4	46
顏色	Color	95.7	95
幾何數形狀	Geometric shape	100	13
貨幣	Currency	49.5	105
情緒	Emotion	95.7	94
疾病	Disease	93.2	74
精神病	Mental disorder	60	5
民族	Ethnicgroup	88.3	43
語言	Language	88	50
行星	Planet	81.2	48
宗教	Religion	-	0
武器	Weapon	82.7	29
天氣現象	Weather phenomenon	90.9	55
網站	Website	75	4
罪行	Crime	86.3	117
科系	Department	100	30
節慶	Festival	73.1	67
朝代	Dynasty	100	39
中藥	Chinese medicine	95.8	72
藥物	Drug	88.1	42
電玩	Video game	69.7	33
電視節目	Television show	50	2
運輸工具	Transportation	97.8	46
政治職位	Political position	87.3	71
政黨	Political party	77.7	9
報紙	News paper	81.8	11
服裝	Clothing	89.6	87
戲劇類型	Drama type	94.1	17
臺灣偶像劇	Taiwan drama	100	2
美國影集	American drama	100	3
大陸影集	Inland drama	33.3	3
韓國偶像劇	Korea drama	-	0
電影	Movie	42.8	49

Table 8: Precisions and number of promoted facts for categories(81 - 120) first five iterations.

Predicate Name		Precision(%)	Number of Promoted(#)
音樂專輯	Music album	-	0
音樂流派	Music genre	66.6	3
樂器	Music instrument	95.9	49
詩詞	Poetry	58.3	48
運動	Sport	90	40
體育賽事	Sports event	88.1	42
球類運動隊伍	Sport team	50	2
球類運動位置	Sport position	73.6	19
運動場地	Stadium	-	0
資訊工程領域	Computer science area	55.4	74
程式語言	Programming language	85.2	61
機構	Institution	62.8	35
公司	Company	100	6
汽車製造商	Automobile maker	64.2	28
航空公司	Airline company	100	2
政府機構	Government organization	80.6	93
時間量詞	Time	75	8
日期	Date	100	8
星期	Week	44.4	9
月份	Month	42.5	40
年份	Year	93.7	32
世紀	Century	-	0
戰爭	War	94.1	17
稱謂	Appellation	83.7	43
物理單位	Unit	77.3	106
Average		79.9	4226

Table 9: Precisions and number of promoted facts for categories(121 - 145) first five iterations.

Appendix C: Chinese-English Mapping Table

Categories			
國家	country	城市	city
大陸	continent	歌手	singer
專輯	album	運動	sport
大學	university	學系	departments of university
Relations			
國家擁有城市	CountryContainsCity	城市位在國家	CityIsLocatedAtCountry
歌手的專輯	SingerProducesAlbum	朝代之後的朝代	DynastyFollowedByDynasty
兩人是隊友	IsTeammatesOf	國家的首都	CountryAndCorrespondingNationalCapital
Locations			
臺灣	Taiwan	台北	Taipei
台北市	Taipei city	台中	Taichung
高雄	Kaohsiung	美利堅合眾國	United States of America
美國	U.S.A	華盛頓特區	Washington, D.C.
紐約	New York	加拿大	Canada
朝鮮	Korea	拜占庭	Byzantium
君士坦丁堡	Constantinople	日本	Japan
東京	Tokyo	福建	Fujian
湖北	Hubei	沖繩縣鐵路車站	Okinawa prefecture railway station
Dynasties			
元朝	Yuan dynasty	明朝	Ming dynasty
清朝	Qing dynasty		
Players			
林書豪	Jeremy Lin	詹姆士哈登	James Harden
Departments of university			
醫學系	Department of Medicine	法律學系	Department of Law
		電機工程學系	Department of Electrical Engineering
		資訊網路與多媒體研究所	Graduate Institute of Networking and Multimedia
Part of sentences			
		的外交官	diplomatist of
		產地是在法國	Made in France
		朝鮮提供經濟援助	Korea provide financial assistance
		英勇強壯的美國隊長	strong and heroic Captain America
		台灣的首都是台北市	the capital of Taiwan is Taipei
		台北市，位於台灣北部	Taipei is located at northern Taiwan
		入境台灣有兩種通關方式	there are two ways to entry to Taiwan
		駐朝鮮的外交官不理會朝鮮撤離警告	diplomat stationed in Korea ignore evacuation warnings
		拉丁人佔領了拜占庭的首都君士坦丁堡	Latins occupied the Byzantine capital of Constantinople

Table 10: Chinese-English mapping table for Chinese texts in thesis.